Spring 3-2018

A Holistic Methodology for Profiling Ransomware Through Endpoint Detection

Stefani K. Hobratsch
Dakota State University

A HOLISTIC METHODOLOGY FOR PROFILING RANSOMWARE THROUGH ENDPOINT DETECTION

A dissertation submitted to Dakota State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Cyber Operations

March 2018

by

Stefani K. Hobratsch

Dissertation Committee:

Dr. Joshua J. Pauli
Dr. Wayne E. Pauli
Dr. Joshua A. Stroschein

DSU
DAKOTA STATE

DISSERTATION APPROVAL FORM

This dissertation is approved as a credible and independent investigation by a candidate for the Doctor of Philosophy in Cyber Operations degree and is acceptable for meeting the dissertation requirements for this degree. Acceptance of this dissertation does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department or university.

Student Name: Stefani K. Hobratsch

Dissertation Title: A HOLISTIC METHODOLOGY FOR PROFILING RANSOMWARE THROUGH ENDPOINT DETECTION

Dissertation Chair: ______________________ Date: 3/20/18
Dr. Joshua J. Pauli

Committee Member: Wayne Pauli Date: 3/20/18
Dr. Wayne E. Pauli

Committee Member ______________________ Date: 3/20/18
Dr. Joshua A. Stroschein

ACKNOWLEDGMENT

This dissertation was made possible by the support of numerous family members, friends, and colleagues. I am forever grateful to the people and experiences that have enabled me to complete this accomplishment.

To my husband, thank you for your love and support. You are the best husband in the world!

To my parents, thank you for instilling in me the importance of faith, family, and a strong work ethic. Thank you for encouraging me to study computer science and urging me to pursue advanced degrees. I’ve come a long way since learning to type on our Apple IIGS!

To my dissertation committee, thank you for guiding me through this research, spurring me forward with ideas, and driving me to develop relevant results. Dr. Josh Pauli, I was always impressed by your lightning fast feedback! Thank you for your support, enthusiasm, and guidance both academically and professionally. Dr. Wayne Pauli, you have done an excellent job directing this degree program. Thank you for keeping me on schedule and helping me navigate the requirements involved in getting to the finish line. Dr. Josh Stroschein, your malware expertise was essential in helping me formulate this project. Thank you for always sharing new ideas and tools with me to keep my technical skills sharp. I could not have asked for a better committee!

To my many professors over the years at various academic institutions, thank you for sharing your knowledge with me and inspiring me to be curious.

To Dakota State University and all those involved in creating this degree program, thank you for being forward thinking and identifying a need for highly-trained cyber security professionals. This degree program fills a critical gap in academia by providing advanced technical classes and relevant applied research. I am proud to be a graduate of this degree program from Dakota State University.

Most of all, thank you to my Lord and Savior Jesus Christ for the many blessings in my life.

ABSTRACT

Computer security incident response is a critical capability in light of the growing threat of malware infecting endpoint systems today. Ransomware is one type of malware that is causing increasing harm to organizations. Ransomware infects an endpoint system by encrypting files until a ransom is paid. Ransomware can have a negative impact on an organization's daily functions if critical business files are encrypted and are not backed up properly.

Many tools exist that claim to detect and respond to malware. Organizations and small businesses are often short-staffed and lack the technical expertise to properly configure security tools. One such endpoint detection tool is Sysmon, which logs critical events to the Windows event log. Sysmon is free to download on the Internet. The details contained in Sysmon events can be extremely helpful during an incident response. The author of Sysmon states that the Sysmon configuration needs be iteratively assessed to determine which Sysmon events are most effective. Unfortunately, an organization may not have the time, knowledge, or infrastructure to properly configure and analyze Sysmon events. If configured incorrectly, the organization may have a false sense of security or lack the logs necessary to respond quickly and accurately during a malware incident.

This research seeks to answer the question "What methodology can an organization follow to determine which Sysmon events should be analyzed to identify ransomware in a Windows environment?" The answer to this question helps organizations make informed decisions regarding how to configure Sysmon and analyze Sysmon logs. This study uses design science research methods to create three artifacts: a method, an instantiation, and a tool. The artifacts are used to analyze Sysmon logs against a ransomware dataset consisting of publicly available samples from three ransomware families that were major threats in 2017 according to Symantec. The artifacts are built using software that is free to download on the Internet. Step-by-step instructions, source code, and configuration files are provided so that other researchers can replicate and expand on the results. The end goal provides concrete results that organizations can apply directly to their environment to begin leveraging the benefits of Sysmon and understand the analytics needed to identify suspicious activity during an incident response.

DECLARATION

I hereby certify that this dissertation constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the dissertation describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Stefani K. Hobratsch

Stefani K. Hobratsch

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Background of the Problem

This research focuses on computer security incident response and provides rigorous results that help organizations make informed decisions about the tools and techniques used to identify suspicious activity on computer systems during an incident response. The Verizon "2017 Data Breach Investigations Report" defines an incident as "a security event that compromises the integrity, confidentiality or availability of an information asset". Computer incident response, which is the act of responding to an incident, is an increasingly critical skill for all Information Technology (IT) professionals. In 2004, the National Institute of Science and Technology (NIST) stated that a computer security incident response capability is "necessary for rapidly detecting incidents, minimizing loss and destruction, mitigating the weaknesses that were exploited, and restoring IT services." (Grance, Kent & Kim, 2004). A recent CSO Online article predicted that the growing impact of cybercrime has made every IT position a cybersecurity position (Morgan, 2018).

Logging important data from computers and networks is essential to facilitate incident response efforts. In 2006, NIST recommended that "organizations be proactive in collecting useful data" to aid incident response effectiveness (Kent, Chevalier, Grance & Dang, 2006). Specific recommendations included auditing operating system behavior and implementing centralized logging so that data is available during forensic events. One month later, NIST published recommended guidelines for how to manage computer security logs, including how to generate, transmit, store, analyze, and dispose of computer security log data (Kent & Souppaya, 2006). In the years since this NIST guidance was published, computer security incidents have increased dramatically. An information security survey conducted by PwC concluded that the number of reported security incidents averaged 117,339 attacks per day in 2014 and the growth rate of detected incidents has increased 66% each year since 2009 (PwC, 2015). In this rapidly growing environment of security incidents and cybercrime, organizations must be even more prepared to respond to computer security incidents.

Statement of the Problem

Organizations must be prepared to respond to computer security incidents. Many tools and techniques exist to provide insight into what events took place on a system during an incident response. The Sysinternals Suite provides a bundle of tools to gain insight into Microsoft Windows systems (Russinovich & Garnier, 2016). Sysmon is a tool in the Sysinternals Suite that logs specific system activity to the Windows event log to help record and identify malicious or anomalous activity (Russinovich & Garnier, 2017). The Symon author Mark Russinovich has given numerous talks showing how he used Sysmon to identify malware running on a Windows system (Russinovich, 2014, 2015, 2016, 2017). Sysmon provides logging only; the tool does not analyze events. Sysmon documentation states that Sysmon should be used with a Security Information & Event Management (SIEM) system to provide event correlation for analysis (Russinovich & Garnier, 2017).

The Sysmon tool is useful for logging Windows system activity, but there is little guidance concerning which Sysmon events are most likely to indicate suspicious activity. Russinovich has stated that creating a good configuration file is an iterative process consisting of a cycle that involves editing the configuration, deploying it, and assessing it (Russinovich, 2017a). Unfortunately, many organizations do not have the knowledge or the infrastructure to perform detailed testing of security tools such as Sysmon. In addition, when faced with so many Sysmon configuration options, many organizations do not have the time or the expertise to configure the tool effectively or analyze the results properly.

Objectives of the Research

This research strives to answer the question “What methodology can an organization follow to determine which Sysmon events should be analyzed to identify ransomware in a Windows environment?” The objective of this research is to develop a practical and relevant method that organizations can utilize to analyze Sysmon events in their Windows environments to aid incident response procedures. The purpose of this research is to prove the effectiveness and value of Sysmon event logging by creating a comprehensive implementation of the method and demonstrating how to analyze Sysmon events. The implementation will be used to study which Sysmon events are triggered most frequently by a publicly available ransomware dataset to provide guidance regarding which Sysmon events are most useful in detecting ransomware.

Summary

This research study explores how to analyze Sysmon logs to identify ransomware and improve incident response procedures. This paper presents the resulting research in a manner that can be applied and implemented by other organizations and researchers. Chapter 1 provides an introduction to the problem statement and describes the importance and relevance of the research question. Chapter 2 explores prior work regarding related software, analytic approaches, and datasets. A gap in existing literature is identified regarding analyzing Sysmon events with ransomware. The gap will be addressed by this research. Chapter 3 explains the design science research methodology that is followed to answer the research question. Three artifacts developed by the research are introduced and requirements for the artifacts are defined. Chapter 4 contains the results of the conducting the iterative research. Implementation details are provided regarding the developed artifacts and results are evaluated to demonstrate the effectiveness of profiling ransomware using Sysmon events. Chapter 5 summarizes the unique contributions provided by this research, including limitations, recommendations, and future work. References and Appendices are provided to augment the results and provide further details to organizations seeking to replicate the results obtained by this research.

CHAPTER 2

LITERATURE REVIEW

Endpoint Detection and Response (EDR) is a category of computer security tools that detect and respond to suspicious activity observed on hosts, as opposed to activity observed on network segments (Chuvakin, 2013). EDR solutions are popular because endpoint devices are often the target of exploitation (Reis, 2016). More than 30 EDR vendors existed in 2015 and EDR marketplace was expected to double in 2016 (Higgins, 2015). Although EDR tools are robust, they cannot stop every attack. Users might have a false sense of security if they do not fully understand an EDR tool's capabilities and limitations. In addition, while EDR tools provide detailed incident response information, the technical details may only be understood by the most experienced security professionals (Chuvakin, 2016). Smaller companies and less experienced security teams may not be able to leverage the capabilities and data insight that EDR tools provide. Sysinternals Sysmon is a tool that performs endpoint detection only, not response. This literature review explores previous research and open source contributions regarding the endpoint detection tool Sysmon. Specifically, existing literature is examined to determine how Sysmon has been used to identify ransomware during an incident response.

Sysmon Capabilities

Sysmon is an endpoint detection tool that logs specific Microsoft Windows system events to the Windows event log (Russinovich & Garnier, 2017). Sysmon was written by Mark Russinovich and Thomas Garnier (Perez, 2014). Sysmon was originally released in 2014, and the tool is frequently updated. Sysmon version 6.20 was released on November 22, 2017 (Russinovich & Garnier, 2017). Sysmon provides documentation online and from the command line tool. An additional resource developed by Michael Haag provides numerous references to online sources with Sysmon details, including configuration, deployment, and analysis examples (Haag, 2017a). Sysmon only operates on Microsoft Windows systems. Microsoft Windows is the most popular desktop operating system worldwide, consisting of 82.68% of the market share as of December 2017 (StatCounter, 2017). Therefore, studying

Microsoft Windows desktop endpoints is extremely relevant and useful to a large majority of organizations.

Each Sysmon event records various types of system activity information in descriptive fields to help an incident responder understand what happened during an event. Sysmon version 6.20 is capable of logging 21 event types, which are enumerated in Appendix A. Sysmon logging can be filtered by providing an Extensible Markup Language (XML) configuration file to specify which events to include and which events to filter out. The full Sysmon schema can be accessed from the command line to understand the type of details that are recorded by Sysmon (Russinovich & Garnier, 2017). SwiftOnSecurity (2017) provides a Sysmon configuration file that provides a good starting point for deploying Sysmon. The configuration is very well documented and provides great insight into various Sysmon filtering options. The SwiftOnSecurity configuration file is licensed with a Creative Commons Attribution 4.0 license, which allows others to "privatize, fork, edit, teach, publish, or deploy for commercial use - with attribution in the text" (SwiftOnSecurity, 2017).

This research builds upon the SwiftOnSecurity Sysmon configuration file because the configuration is thorough and robust and the license allows such usage. Creating a Sysmon configuration file from scratch would be duplicating efforts that the community has already completed. In addition, the SwiftOnSecurity configuration file is still an actively developed project, so updates provided by SwiftOnSecurity can be leveraged by this study. The SwiftOnSecurity configuration file specifies which Sysmon events to log, but there is no existing mechanism to verify that the desired Sysmon events are being logged as expected. A review of existing literature did not identify a tool available to trigger all the Sysmon events and confirm that the configuration file is working as expected.

Sysmon During Incident Response

Several existing efforts have leveraged Sysmon during incident response. In 2016, Mark Russinovich presented "Tracking Hackers on Your Network with Sysinternals Sysmon" at the RSA Conference, a highly-respected computer security conference (Russinovich, 2016). In the presentation, Russinovich discussed how he used Sysmon to detect malware running as a scheduled task, as well as malware escaping from restricted environments. In 2017, Russinovich presented "How to Go from Responding to Hunting with Sysinternals Sysmon" and demonstrated how to use Sysmon to hunt a phishing email and identify credential stealing (Russinovich, 2017a). Others, such as Jerome Quentin, have said that Sysmon is a good tool for threat hunting and incident response (Quentin, 2017). Lenny Zeltser of SANS, a popular cybersecurity training organization, pointed out that Sysmon is helpful for “system administrators, incident responders and forensic investigators” because it records interesting events for log collection (Zeltser, 2014). These efforts demonstrate that Sysmon is an appropriate tool to use during incident response, which gives credence to this study.

Analyzing Sysmon Results

Sysmon provides logging only; the tool does not analyze events or respond to alerts. Sysmon documentation explicitly states that “Sysmon does not provide analysis of the events it generates, nor does it attempt to protect or hide itself from attackers” (Russinovich & Garnier, 2017). Sysmon documentation recommends using a Security Information & Event Management (SIEM) system to provide event correlation (Russinovich & Garnier, 2017).

One popular SIEM system is Splunk. There are discussions online about using Splunk to analyze Sysmon logs (ncis0x007, 2016; Haag, 2017b; Hall, 2016; Hayes, 2016; Splunk, 2017; Crypsisgroup, 2016), but Splunk is an expensive corporate tool and may not be a feasible solution for individuals or small businesses due to high costs (Dreyfuss, 2015). Another popular SIEM is LogRhythm (LogRhythm, 2018). LogRhythm released a webcast to demonstrate how to use LogRhythm to collect and investigate Sysmon logs (Reynolds & Smith, 2017). LogRhythm may also be priced out of the reach of many organizations, with the LogRhythm appliance solution starting at $35,000 (Stephenson, 2017).

Unlike Splunk and LogRhythm, which are commercial data analysis tools, the Elastic Stack is an open source collection of tools to perform data analysis. The Elastic Stack, formerly referred to as the ELK stack, consists primarily of the open source software products Elasticsearch, Logstash, and Kibana (Langlois, 2016). Elasticsearch provides an open source, high-speed, near real-time, distributed search engine. (Banon, 2010). Elasticsearch is currently the most popular search engine (solid IT gmbh, 2018). Logstash performs data collection and parsing (Elastic, n.d.). Kibana is a visual front-end for Elasticsearch (Langlois, 2016).

The Elastic Stack is a promising tool to use during the analysis portion of this study. In 2017, Elastic announced that the Elastic Stack had been downloaded over 100 million times (Elastic, 2017). Elastic products are rising in popularity to monitor and analyze security logs (adm, 2015). There is a precedence for using Elastic to examine security event logs. Mozilla uses Elastic to analyze their security logs (Bryner, 2015). The CyberSponse Security Operations Platform has integrated Elastic into their platform (CyberSponse, 2015). Informal tutorials are available online that discuss using Sysmon logs and Elastic products during incident response (Bandos, 2016; Churchill, 2015; Lewis, 2015), but no rigorous academic studies exist to explore using the two tools together.

In addition, many of the online discussions do not provide access to all of the configuration files used to create the Elastic environment. This makes it difficult to replicate and implement similar results. Some of the studies that do provide source code only provide basic implementations. For example, two web articles have published Logstash pipelines to support the parsing of Sysmon events (Andy, 2017; Delgado, 2017). However, the pipelines are simplistic. They provide a good starting point for organizations getting started with Sysmon, but they do not implement data enrichment and customized parsing to suit the needs of detailed incident response analytics. There is a need for robust, open source guidance to help organizations implement their own Elastic environment to analyze Sysmon logs.

Ransomware

Ransomware is affecting users and businesses worldwide. Ransomware is a type of malware that restricts access to an infected system until the ransom is paid (US-CERT, 2016). Crypto-ransomware uses an algorithm to encrypt files on an infected machine to prevent access to the files until payment is received (F-Secure, 2018). Encrypted files are usually renamed with a new file extension (Dubey, 2016). Since the attacker’s goal is to be paid a ransom, the user is typically notified via a text document or webpage that files have been encrypted. The notification document contains instructions on how to pay the ransom to recover and decrypt the files (Trend Micro, n.d.). Although there are different types of ransomware besides crypto-ransomware (Steinberg, 2018), this research will use the simplified term ransomware to refer to crypto-ransomware.

The threat of ransomware has increased over the past few years. The Federal Bureau of Investigation (FBI) stated that law enforcement observed an increase in ransomware attacks against organizations during 2015 (FBI, 2016). Check Point reported that ransomware attacks consisted of 10.5% of all worldwide malware attacks in the second half of 2016 (Check Point, 2017a, 2017b). Studies show that ransomware is a growing threat to organizations (Schulze, 2017; O'Brien, 2017). Sometimes paying the ransom is an organization's only option, due to lack of backups and ransomware detection tools (Palmer, 2017).

Due to the increasing threat and impact of ransomware, this research explores the usefulness of analyzing Sysmon events to identify ransomware. Splunk posted an article in 2016 on how to use Splunk to find ransomware in Sysmon logs by searching for a large number of files being created in a short time (Hayes, 2016). Hayes only demonstrated the analytic using one ransomware sample, and the name of the ransomware sample was not mentioned. This article provides good ideas for Sysmon analytics, but it does not provide a repeatable environment that organizations can implement on their own. As previously mentioned, Splunk is not a feasible solution for all organizations due to cost. There is a need to expand the analytics presented by Hayes into an open source solution, with detailed guidance and advanced customization to profile multiple ransomware samples.

LogRhythm posted an online article on analyzing WannaCry ransomware activity in Sysmon logs using the LogRhythm software (Costis, 2016). The article only focused on two Sysmon events, instead of all possible Sysmon events. In addition, details about the WannaCry sample used were not disclosed. As previously mentioned, LogRhythm may be too expensive for some organizations. Once again, existing efforts have discussed using Sysmon to investigate ransomware, but an open source, repeatable solution is not available. There is a need to expand existing analytics into a robust solution that leverages all Sysmon event logs and that an organization can implement fully to perform their own analytics.

Elastic posted a blog entry on how to detect WannaCry ransomware using Sysmon with an Elastic Stack (McDiarmid, 2017). McDiarmid looked specifically for activity related to WannaCry downloading, executing, spreading, and terminating. McDiarmid investigated only one WannaCry sample executable, and he focused specifically on searching for signatures that WannaCry is known to exhibit. There is a need to expand Sysmon analysis over multiple ransomware families to compare how different samples behave. In addition, a generic approach is needed to profile ransomware behavior, instead of looking for specific, known behaviors. The results need to be presented in a repeatable manner that organizations can leverage in their own environments.

The articles discussed above confirm that the community is interested in using Sysmon events during incident response, indicating that this topic is relevant and timely. Existing work has been performed with Elastic and Sysmon, but gaps have been identified to promote further understanding of how Sysmon logs can be used to investigate ransomware.

Summary

This research explored previous research conducted using the endpoint detection tool Sysmon. Sysmon documentation states that Sysmon does not provide analysis of events, and that any analysis must be done in another tool. Some researchers have used Splunk and LogRhythm to analyze Symon events, but commercial tools are expensive and not viable for many organizations. Elastic Stack is an open source solution that provides storage and searching of distributed data. A gap was identified in existing literature regarding the analysis of Sysmon events to identify ransomware in a generic, repeatable manner. Solutions to fill this gap need to provide configuration files and ransomware samples so that results can be repeated and enhanced by the incident response community. The remainder of this paper describes a research study that was conducted to fill the identified research gap and answer the question “What methodology can an organization follow to determine which Sysmon events should be analyzed to identify ransomware in a Windows environment?” Details regarding the methods followed to construct this research study are found in Chapter 3. Results and design documents created by the study are found in Chapter 4. A summary of contributions provided by this research to address the identified research question, including limitations, recommendations, and future work, is found in Chapter 5.

CHAPTER 3

RESEARCH METHODOLOGY

The literature review identified a gap in the availability of detailed guidance for organizations desiring to analyze Sysmon events to identify ransomware. This gap led to the development of the research question “What methodology can an organization follow to determine which Sysmon events should be analyzed to identify ransomware in a Windows environment?” Selecting an appropriate research methodology provides guidance and techniques to answer the research question (Kumar, 2014). Due to the nature of the research question, a design science research methodology was selected to provide a framework for this study. This chapter describes why a design science methodology was selected to address the research question. Advantages and shortcomings of using design science methods to conduct this study are discussed. The research model and artifact requirements that directed the execution of the study are described in detail. Validation that the research conducted by this study conformed to design science principles is also presented. Chapter 4 contains the results that were obtained by performing the study. Chapter 5 contains a detailed summary of contributions produced as an outcome of this research.

Research Methodology Justification

Three research methodologies were originally considered for this study: quantitative, qualitative, and design science.

Quantitative research methods are typically used to measure relationships between two variables to explain an observation (USC, 2018). Quantitative research commonly explores observable facts and typically results in numerical outcomes and objective outcomes. Quantitative research was not appropriate for this study because the research goal was not to measure the performance of Sysmon, but rather to develop useful tools and techniques to produce a deeper understanding of how to leverage Sysmon events to identify ransomware. Thus, quantitative methods were not appropriate to develop tools to analyze Sysmon events. However, once tools and techniques are fully developed, quantitative methods would be very appropriate in future research to compare and study the efficiency of the resulting tools and techniques to identify the solution that has the best performance or maximizes resource utilization.

Qualitative research methods are commonly used to study behaviors and subjective research questions that require inductive reasoning (Creswell & Creswell, 2018). Qualitative studies typically employ questionnaires and interviews to collect relevant data to answer a qualitative research question. Qualitative research was not appropriate for this study because the research goal was not to understand behavior related to Sysmon or study why users do or do not prefer using Sysmon. The goals of this study were to develop tangible procedures that could be used by organizations desiring to use Sysmon to detect ransomware. Thus, qualitative methods were not appropriate to address this research problem. However, once a solution is developed by design science, qualitative methods would be quite appropriate in future research to study human-computer interaction of the resulting tools and to understand what an organization finds most beneficial when using the outcomes of this study.

A design science research methodology was selected to conduct this study. Design science research seeks to create useful artifacts that solve an unknown problem or improve an existing solution (Hevner et al., 2004). Sysmon and Elastic products are existing, freely available tools. However, the literature review identified a gap in guidance on analyzing Sysmon events to identify ransomware using Elastic. This study will improve upon existing ideas that have been implemented in other tools, as well as add new capabilities to use Sysmon logging to investigate and profile ransomware.

Design science produces new knowledge relevant to the community (Vaishnavi & Kuechler, 2004). This study produced new knowledge in the form of a comprehensive environment to analyze Sysmon events that will be useful to the computer security incident response community. The environment can be implemented by organizations seeking to address the identified research question "What methodology can an organization follow to determine which Sysmon events should be analyzed to identify ransomware in a Windows environment?"

The primary driver behind selecting design science research methods was to iteratively develop a solution that organizations could apply directly and immediately to leverage Sysmon in Microsoft Windows environments. Hevner, et al. (2004) state that the ultimate evaluation for research is the question “What are the new and interesting contributions?” By following design science methods, this study ensured utility of the research contributions because development was performed in a cyclical manner enabling rapid improvements and adjustments. Design science methods led directly to the development of artifacts that can be leveraged by organizations to identify ransomware using Sysmon log analysis.

Research Model

This section describes the research steps that were planned and executed using a design science research methodology to answer the research question “What methodology can an organization follow to determine which Sysmon events should be analyzed to identify ransomware in a Windows environment?” Design science methods allowed for iterative research, which enabled the development of research artifacts by implementing incremental improvements throughout the research study to properly address the research question.

The following steps were conducted to implement this research. The results obtained after conducting the research steps below are found in Chapter 4.

1. Compile a corpus of ransomware to use during testing.
2. Develop a method to analyze Sysmon events.
3. Develop a virtual environment to apply the method.
4. Develop a tool to trigger all Sysmon events and verify that the virtual environment logs Sysmon events as expected.
5. Execute ransomware samples in the virtual environment and analyze results.
6. Develop dashboards, alerts, and queries to aid in automating Sysmon analysis.
7. Repeated Steps 2-6 iteratively as needed to profile ransomware and answer the research question “What methodology can an organization follow to determine which Sysmon events should be analyzed to identify ransomware in a Windows environment?”
8. Make results, configuration files, and code available for usage by the community to implement and enhance through future work.

Artifacts of Design Science

Artifacts are the primary contribution created by design science research (March & Storey, 2008). A design science research artifact is a useful product developed as part of the research process to help answer the research question. This study used a design science research methodology to create innovative and useful artifacts that can be leveraged by the cyber security community during an incident response. The purpose of the artifacts developed by this research was to help organizations utilize Sysmon for event logging and ransomware detection. The artifacts can be used to make educated decisions about which Sysmon events to enable and to guide the analysis of Sysmon events to determine what occurred on an infected system during an incident response.

This research identified three artifacts that were needed to explore the research question: a method, an instantiation, and a tool. This section documents the requirements of the three artifacts developed by this study. Accurate requirements help ensure that the resulting artifacts are useful and applicable to the research domain. Details regarding the resulting development of the artifacts are found in Chapter 4.

A Method. A method is needed to help organizations leverage Sysmon during incident response in Windows environments. In design science, a method is "a set of steps (an algorithm or guideline) used to perform a task" (March & Smith, 1995). The method created by this research will document the key steps needed to deploy and analyze Sysmon logging to investigate ransomware. The Sysmon tool is freely available, but there is little guidance on how to fine-tune the configuration files and analyze the results. The method will serve as a guideline to organizations who wish to use Sysmon but do not know where to start. The only requirement is that the method must provide guidance to answer the research question, "What methodology can an organization follow to determine which Sysmon events should be analyzed to identify ransomware in a Windows environment?" The method will be generic so that it can be applied to various types of ransomware.

An Instantiation. The most critical artifact needed for this research is an instantiation that implements the method designed above. In design science, an instantiation is "the realization of an artifact in its environment" (March & Smith, 1995). Instantiations can operationalize a method and can help articulate a method further (March & Smith, 1995). In this research, the method guided the development of the instantiation, and the instantiation helped formalize the steps in the method. The instantiation will be used to analyze which Sysmon events are triggered most frequently by ransomware. The instantiation will prove that the method can be implemented and used by organizations seeking to leverage Sysmon to identify ransomware. The instantiation is a tangible example of how to apply the method to answer the research question. The requirements for the instantiation artifact are introduced in Table 1. Hardware specifications are included as a requirement to demonstrate the relevance of the solution to an average user due to the use of low cost and commodity hardware and software. The completed instantiation can be implemented by anyone desiring to setup an infrastructure and begin exploring Sysmon data.

Table 1. Requirements for Instantiation Artifact

Number	Requirement
1.	Must be implemented in a virtual environment consisting of at least 2 systems: a) Microsoft Windows 7 Service Pack 1 endpoint system to generate Sysmon events b) Correlation Server to collect Sysmon events
2.	Must operate on the following hardware: **System:** Dell Inspiron 7559 **CPU:** Intel Core i7-6700HQ @ 2.60GHz **RAM:** 16 GB **Hard Drive:** 1TB Solid State Drive
3.	Must implement open source and/or freely available downloadable software whenever possible.
4.	Must document software versions implemented in final solution so that others may repeat the results.

A Tool. A third artifact is needed to verify Sysmon logging. A tool will be created to trigger every Sysmon event. This tool is technically considered a design science instantiation, but this research refers to the artifact as a tool to distinguish it from the instantiation artifact described above. This tool will be used to ensure that the test environment is properly configured by verifying the logging of all Sysmon events. No capability of this nature was identified while reviewing existing literature. The requirements used to develop the trigger tool artifact are introduced in Table 2.

Table 2. Requirements for Trigger Tool Artifact

Number	Requirement
1.	Must run on Microsoft Windows 7 Service Pack 1 operating system.
2.	Must be run with administrative privileges.
3.	Must be written in an interpreted language so users can easily confirm that the tool contains no malicious behavior.
4.	Must trigger all 21 of the Sysmon events. Does not need to trigger the Sysmon error event (event ID 255).
5.	Must execute fully without requiring a reboot.
6.	Must limit external dependencies as much as possible. If dependencies on other software or libraries cannot be avoided, those dependencies must be fully documented.

Advantages and Shortcomings of Design Science Research

Every research methodology has advantages and shortcomings. Design science is advantageous in this research scenario because it provides utility to organizations desiring to use Sysmon effectively to identify ransomware. Utility is the main advantage of design science research. The output of this research is three artifacts designed to help organizations and system administrators leverage Sysmon during an incident response. The resulting artifacts are useful to organizations, even though an organization may not fully understand every aspect of Sysmon. Another benefit to design science is the iterative approach that allows the researcher to evaluate if progress is being made toward the research goal. If the research is not leading to an artifact that is useful to the problem statement, the research can be adjusted and modified to ensure that research results are useful. Continuous iterations that are constantly critiqued against the research question enable the development of useful artifacts that are immediately relevant to the incident response community.

In this research scenario, design science has a few shortcomings when compared to quantitative and qualitative methodologies. Design science results in the creation of artifacts, but not in an understanding of how or why Sysmon or ransomware performs in a certain manner. This research study provides guidance to organizations regarding Sysmon analysis, but it does not provide in-depth understanding of how Sysmon works or how ransomware triggers specific Sysmon events. Another shortcoming is that design science results can be perishable because the cybersecurity domain changes so quickly. Resulting artifacts could become obsolete before they are effectively implemented in an organization (Hevner et al., 2004). Throughout the duration of this research, both the Sysmon and the Elastic Stack software versions were regularly updated. Fortunately, backwards compatibility was maintained, and the updated software did not break any functionality during this research. However, newer versions of Sysmon and Elastic have already been released prior to the publication of this research. Another drawback to design science is that artifacts are customized to a specific environment (Hevner et al., 2004). This research is customized to Sysmon version 6.20 running on Windows 7 using a specific ransomware dataset. Results may not be applicable to other environments, including other operating systems, software versions, or malware samples. These shortcomings are reduced during this research by producing a repeatable method that can be applied to other tools and datasets. Overall, the advantages of design science far exceeded the shortcomings when applied to this research question.

Validation of Design Science Research

To validate that the resulting research followed design science principles, the research goals and outcomes are dissected below into the five steps of the Vaishnavi and Kuechler (2004) design science research process.

Awareness of Problem. This research identified that endpoint detection logging is critical for effective incident response. The Sysmon tool was identified as a tool used to log critical Microsoft Windows events, but there were no existing academic studies to measure the effectiveness of Sysmon at recording malicious activity. Because there was no rigorous guidance on how to configure Sysmon, users could not ensure that they were monitoring their Windows systems in the most effective manner to log potential ransomware activity. This left users without confidence in their Sysmon implementations and incident response readiness.

Suggestion. This research suggested and explored a solution to the identified research question through the creation of three design science artifacts. The research investigates the effectiveness of Sysmon at logging malicious activity and provides empirical evidence regarding which Sysmon events are triggered most frequently by ransomware.

Development. This research developed three useful design science artifacts: a method, an instantiation, and a tool. The method documented a process to follow to identify ransomware using Sysmon logging. The instantiation implemented the method to allow tangible research on the effectiveness of Sysmon logging against the selected ransomware dataset. The trigger tool verified Sysmon logging in the instantiation. All of the artifacts could be further modified to test other endpoint detection tools or other malware samples.

Evaluation. The artifacts were evaluated to determine the effectiveness of the Sysmon configuration. The method was tested by implementing an instantiation to verify that the method was sound. Results were analyzed to determine which Sysmon events were triggered most frequently by the ransomware dataset. Regardless of the outcome of the tests, the artifacts deliver utility to the incident response community by providing insight into how Sysmon performs against the selected ransomware dataset. By providing tangible results, this research contributed to the body of knowledge by enabling organizations evaluate the usefulness of Sysmon logging in Microsoft Windows environments.

Conclusion. This research provided conclusions regarding the results of the research, including conclusions regarding the effectiveness of using Sysmon to identify ransomware. The usefulness of the developed artifacts was articulated and provided the community with a greater understanding of Sysmon logging. Source code and configuration files were made available for the community to download and reuse in their own environments to replicate this research. Future work is documented so that other researchers may continue to expand the field.

If there is any remaining doubt that this research followed sound design science principles, the seven design science guidelines offered by Hevner et al. (2004) are evaluated in the following sections.

Design as an Artifact. The research resulted in the design of a method, an instantiation, and a tool. The creation of the artifacts was the primary focus of the research to answer the identified research question “What methodology can an organization follow to determine which Sysmon events should be analyzed to identify ransomware in a Windows environment?” All of the artifacts worked together to allow an organization to evaluate the effectiveness of Sysmon at logging ransomware activity.

Problem Relevance. The research addressed a relevant and important business problem: the lack of guidance regarding how to use Sysmon to analyze malicious events, leaving organizations on their own to configure, deploy, and assess the effectiveness of Sysmon event logging, an especially difficult task for small organizations without dedicated computer security personnel. The use of a current ransomware dataset added to the relevance of this research.

Design Evaluation. The three developed artifacts helped evaluate the research results. The method artifact was used to define how to leverage Sysmon events during incident response. The instantiation artifact was used to evaluate the effectiveness of the method at identifying ransomware. The trigger tool artifact confirmed that Sysmon logging was behaving as expected in the instantiation. The Sysmon events were analyzed with observational methods and experimental methods. Histograms were generated to make conclusions regarding which Sysmon events are triggered most frequently by the selected ransomware dataset.

Research Contributions. This research provided valuable contributions to the incident response community and organizations desiring to use Sysmon logging to identify ransomware. The developed artifacts were well-documented for others to understand, use, and enhance for their own needs. The method provided guidance on the steps to take during Sysmon log analysis. The instantiation provided a realistic environment to implement. The tool allowed verification and validation of a Sysmon implementation. Source code and configuration files were made available for others to reuse and modify for their own environment. More details regarding the contributions of this research can be found in Chapter 5.

Research Rigor. A rigorous experimental evaluation of the instantiation was presented to profile ransomware. Data analysis of Sysmon logging was performed using statistical analysis through Elastic Stack dashboards and queries.

Design as a Search Process. The research process sought to find the best way to leverage Sysmon events during incident response to identify ransomware. The artifacts were developed using an iterative approach. Satisficing was used to identify a satisfactory solution to the research question without needing to answer every question about how and why Sysmon performed a certain way (March & Storey, 2008).

Communication of Research. The results were presented as a formal dissertation. The resulting artifacts were made available to the community, including code and configuration files that can be found in the Appendices. The content of this research was appropriately technical so that other researchers may replicate the study. The results were presented in an actionable manner so that organizations without deep technical knowledge can apply the results within their own environment and get started using Sysmon to identify ransomware.

CHAPTER 4

RESULTS AND ANALYSIS

The goal of this research was to provide academically sound results to help organizations leverage Sysmon logging during incident response. The specific research question studied was "What methodology can an organization follow to determine which Sysmon events should be analyzed to identify ransomware in a Windows environment?" A review of existing literature indicated a lack of rigorous research studies involving the analysis of Sysmon events to identify ransomware. A design science research study was conducted to answer the research question and to develop unique and useful artifacts as a result of the research. This chapter discusses the details of the results obtained from carrying out the design science research study proposed in Chapter 3. An overview of research contributions, limitations, and future work are found in Chapter 5.

Ransomware Dataset

As discussed in Chapter 2, studies show that ransomware is a growing threat to organizations (Schulze, 2017; O'Brien, 2017). To achieve relevant and repeatable research results, a ransomware dataset was needed in this research to demonstrate the effectiveness of the developed artifacts at profiling ransomware in Sysmon logs. The use of modern, relevant, freely-accessible ransomware samples was the most important consideration in selecting the dataset.

Sample size was considered when compiling the ransomware dataset. This study was based on design science methods, not quantitative or qualitative methods. Therefore, the dataset served as a proof of concept to exhibit the usefulness of the research results. This research did not rely on statistical methods to measure relationships or derive numerical truths. The sample size of the ransomware dataset used in this design science research study was not as critical as it would have been in a statistical study. The purpose of the dataset was to prove the effectiveness of the developed artifacts. This research study deemed the content of the dataset to be more important than the size of the dataset. A dataset consisting of a few high-ranking ransomware threats would provide more valuable results in this study than a dataset consisting of a thousand out-of-date or unfamiliar ransomware samples.

For the purpose of this study, three samples were considered sufficient to represent a ransomware family. One sample was deemed insufficient to represent an entire ransomware family. Two samples would not provide a means to determine which results are more accurate if discrepancies are found. Three samples were deemed to provide a reasonable comparison to prove the concepts developed by this design science research study. Malware samples change frequently to avoid detection, which can make it challenging to obtain current malware samples. Therefore, requiring more than three samples could prove difficult to find enough samples of a given ransomware family.

Design science research excels at providing relevant results. The first step in selecting the ransomware dataset was to identify ransomware families to study. To ensure that the results of this study were relevant to organizations today, relevant ransomware samples were required. In July 2017, Symantec published a “Ransomware 2017” report which identified six major ransomware threats seen in 2017 (O’Brien, 2017). The six ransomware threats identified by Symantec were Cerber, Jaff, Sage, GlobeImposter, Locky, and Mamba. The samples selected for the dataset used in this research were limited to those six ransomware families due to their significance in 2017. All six of these ransomware families are considered crypto-ransomware (Abrams, 2017; Ducklin, 2016; Hasherezade, 2016; MalwareBytes, 2017; MalwareBytes, n.d.; Trend Micro, 2017). Details regarding crypto-ransomware behavior is found in Chapter 2.

The second step was to identify specific samples within the six major ransomware threats identified by Symantec in 2017. Only publicly available ransomware samples were used in this study to ensure that others could repeat, verify, and expand the results of this research. VirusTotal provided three months of student access to VirusTotal Intelligence during the early stages of this research (VirusTotal, 2018a), but that service was not used to collect samples for this study because the service is not freely available to everyone. Several online malware repositories exist that provide free access to malware samples for researchers (Zeltser, 2018). Hybrid Analysis is a web-based malware analysis service that allows users to upload samples for free and obtain an analysis report (Hybrid Analysis, 2018). If the user marked the uploaded sample as public, then other Hybrid Analysis users could download the sample from the Hybrid Analysis website. The Hybrid Analysis website provides a robust searching capability to identify malware samples based on specific criteria such as file type and hashtag. All of the malware samples used in this research were downloaded from the Hybrid Analysis website due to the search features offered by the website and due to the fact that samples could be downloaded for free from the website (Hybrid Analysis, 2018). Using only one source for the ransomware samples allowed other researchers to obtain the samples more easily without having to register for multiple accounts on multiples repositories.

The third step was to select current ransomware samples to ensure that the samples used in this research were representative of ransomware that an organization might likely encounter today. In order to ensure the ransomware samples were current, only samples dated between July 2017 and December 2017 on the Hybrid Analysis website were considered for the dataset. In addition, only samples that were clearly and only tagged as one of the Symantec-identified ransomware families was considered to ensure that the labeling of the samples was accurate. Some samples in Hybrid Analysis were tagged with multiple ransomware families, which created ambiguity as to how to properly label the ransomware. Such samples were discarded and not used in this study.

The fourth and final step in selecting the ransomware dataset was to confirm that the samples executed properly on a Microsoft Windows 7 Service Pack 1 system without any additional software. Therefore, only Portable Executable (PE) files were considered because PE files are the native executable format used by Windows (Microsoft, n.d.). This requirement ensured that all samples in the dataset would be compatible with the artifacts developed by this study. Proper execution was defined as files getting encrypted, file extensions being renamed, and notification provided to the user regarding how to pay the ransom to decrypt the files. Some samples downloaded from Hybrid Analysis did not execute properly on the test endpoint system. Such samples were discarded and not used in this research because they did not simulate a ransomware infection.

The Hybrid Analysis website was searched for ransomware samples based on the restrictions identified in the steps above. No Sage ransomware samples were available on the Hybrid Analysis website that met the criteria. Only one Mamba ransomware sample and one Jaff ransomware sample was available on the Hybrid Analysis website that met the criteria. Although sample size is not critical in this study, one sample was not sufficient to represent an entire ransomware family. Therefore, Sage, Mamba, and Jaff ransomware families were not considered for this study. Only GlobeImposter, Cerber, and Locky ransomware families were selected for the dataset compiled by this research study due to availability of acceptable samples on the Hybrid Analysis website. Three samples each of GlobeImposter, Cerber, and Locky ransomware were selected from the Hybrid Analysis webpage that met the criteria above. For the purposes of this design science research study, where the samples are used to prove the conceptual ideas developed in this study, not to prove statistical truths, three ransomware samples were deemed sufficient to represent a ransomware family.

The use of recent and prevalent ransomware samples made this research applicable and relevant to organizations. In summary, a dataset consisting of three samples each from three prevalent ransomware families was compiled for this study. GlobeImposter, Cerber, and Locky ransomware families were selected because they were among the list of six major ransomware threats identified in the Symantec 2017 ransomware report (O'Brien, 2017) and public samples were available to download from the Hybrid Analysis website (Hybrid Analysis, 2018). Different ransomware families were used so that this research could investigate if all ransomware families triggered the same Sysmon events or if different ransomware families interacted with Sysmon in a different way.

Overall, nine samples were selected to form the ransomware dataset. Details regarding the dataset samples are provided to assist other researchers in understanding the behavior of the samples without having the execute the samples in their own environment. In addition, the samples are uniquely identified so that other can download the samples and replicate these results in their own environment if desired. A cryptographic hash of each sample is provided to uniquely identify each sample. A cryptographic hash is computed by a cryptographic hash function, which takes an arbitrary number of bytes as input (in this case, the bytes that comprise a ransomware sample) and output a unique fixed length hash value (Gilbert & Handschuh, 2003). In this study, the Secure Hash Algorithm called "SHA-256" was used to generate 256-bit hashes of the ransomware samples (Dang, 2015). Cryptographic hashes are commonly used to uniquely identify files in digital forensics (Roussev, 2009). SHA-256 hashes are available when viewing samples on the Hybrid Analysis website.

Three GlobeImposter ransomware samples were selected for the dataset. The SHA-256 hash of each sample, the reference citation, and the label this research used to refer to each sample are introduced in Table 3. All three samples were executed in the research environment to confirm that the ransomware was functional. The GlobeImposter samples used file extensions “`.crypt`”, “`.STN`”, and “`.coded`” to encrypt files. The samples created the notification files “`how_to_back_files.html`” and “`0_HELP_DECRYPT_FILE.html`” to instruct the user on how to pay the ransom and decrypt the files.

Table 3. GlobeImposter Ransomware Samples

Label	**SHA-256 Hash**
G1	7d49a2a9d788fc8dbaa6331c8b740f689e20600ff7e8d3692b1a9c6d37a37bd6 (Hybrid Analysis, 2017g)
G2	edf67ba035e52cd903017a24271544caba57dace039be51b1e867fdfd5252744 (Hybrid Analysis, 2017b)
G3	b2282de3df95c6a9d0151ad61d2ab4e99400ca3104ce9003a0b13290260a7a55 (Hybrid Analysis, 2017c)

Three Cerber ransomware samples were selected for the dataset. The SHA-256 hash of the sample, the reference citation, and the label this research used to refer to each sample are introduced in Table 4. The Cerber samples used file extensions “`.cerber3`” and “`.8899`” to encrypt files. The samples created notification files “`@__README__@.txt`”, “`@__README__@.html`”, “`_R_E_A_D__T_H_I_S_######.txt`”, and “`_R_E_A_D__T_H_I_S_######.txt`” to instruct the user on how to pay the ransom and restore their files, where `######` in the file name was a random string.

Table 4. Cerber Ransomware Samples

Label	SHA-256 Hash
C1	6563059c4e556e2bc1589b9711a328c4499baed3b0a14b533a467ce65ee37af6 (Hybrid Analysis, 2017i)
C2	403577074344d4832649881daf8885fed4d9afc3e7a4b02247ceb9b51d858794 (Hybrid Analysis, 2017h)
C3	e67834d1e8b38ec5864cfa101b140aeaba8f1900a6e269e6a94c90fcbfe56678 (Hybrid Analysis, 2017a)

Three Locky samples were selected for the dataset. The SHA-256 hash of the sample, the reference citation, and the label this research used to refer to each sample are introduced in Table 5. All three Locky samples used the file extension "`.asasin`" to encrypt files. All the samples created notification files "`asasin.html`" and "`asasin.bmp`" to instruct the user on how to pay the ransom and restore their files.

Table 5. Locky Ransomware Samples

Label	SHA-256 Hash
L1	c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f (Hybrid-Analysis, 2017l)
L2	294f55a28930c8afed9b95d2af108a6916eeb2c79967e91f4dde48026bab15ce (Hybrid-Analysis, 2017e)
L3	4c054127056fb400acbab7825aa2754942121e6c49b0f82ae20e65422abdee4f (Hybrid-Analysis, 2017d)

An interesting feature of ransomware is that it needs to inform the user that the system is infected so that the user will know to pay the ransom. Other types of malware may want to run silently so that users do not know the malware is present. This is not the case with ransomware. While executing the malware, the key effects of the ransomware samples were observed and documented. The observed effects were discussed in the paragraphs above. The file extensions used by each ransomware sample and the notification files created by each sample are introduced in Table 6. This information is useful because it helps incident responders identify infected systems in their environment and identify which ransomware family caused the incident. Screenshots of the notification files created by the samples to demand ransom payment are found in Appendix R. It is important for incident responders to be familiar with how ransomware behaves and what it looks like so that they can act promptly when an incident occurs. The rest of this study used the compiled ransomware dataset to demonstrate the usefulness of the developed artifacts at effectively identifying and analyzing ransomware.

Table 6. Observed Effects of Ransomware Samples

Label	File Extension	Notification Files Created
G1	`.crypt`	`how_to_back_files.html`
G2	`.STN`	`0_HELP_DECRYPT_FILE.html`
G3	`.coded`	`how_to_back_files.html`
C1	`.cerber3`	`@__README__@.txt` `@__README__@.html`
C2	`.cerber3`	`@__README__@.txt` `@__README__@.html`

C3	`.8899`	`_R_E_A_D__T_H_I_S_######.txt`
		`_R_E_A_D__T_H_I_S_######.hta`
		(where ###### is random)

L1	`.asasin`	`asasin.html` `asasin.bmp`
L2	`.asasin`	`asasin.html` `asasin.bmp`
L3	`.asasin`	`asasin.html` `asasin.bmp`

Artifact 1: A Method to Analyze Sysmon Event Logging

This research iteratively created a method that organizations can follow to analyze Sysmon events in Windows environments to identify ransomware. The literature review identified a gap in guidance regarding how to analyze Sysmon events. This method was created to be a guideline for organizations needing to analyze Sysmon events. The method was developed iteratively based on the research study designed in Chapter 3. The method was constructed specifically for this research to identify and document the steps needed to analyze Sysmon logs. The resulting method is introduced in Table 7 and summarized in the paragraphs below.

The method is an overview of steps an organization can take to analyze Sysmon events. The method was used to guide the development of the instantiation and trigger tool artifacts discussed later. The instantiation and trigger tool artifacts also helped improve the method and gave credence to the worthiness of the method. Although the method is straight-forward, it took time to develop and iteratively improve the steps of the method to ensure the results were valuable to organizations and the incident response community. The method was developed according to the design and requirements set forth in Chapter 4.

Table 7. Detailed Method for Utilizing Sysmon Event Logging

Step	Description
1.	Update Sysmon configuration to collect events of interest on the endpoint
2.	Collect Sysmon events into a centralized Security Information & Event Management (SIEM) server. Concrete example: instantiation artifact.
3.	Verify Sysmon logging on the endpoint. Concrete example: trigger tool artifact.
4.	Execute ransomware samples on the endpoint, to simulate an incident.
5.	Analyze the Sysmon events to identify suspicious activity: a. Alert automatically if known suspicious activity is seen. 1. Look for malicious VirusTotal hashes. 2. Look for files created with file extensions commonly used by ransomware. 3. Look for processes that changed file creation times. 4. Look for large number of files created in a short amount of time. b. Review dashboards for suspicious activity: 1. Inspect malicious VirusTotal hashes. 2. Look for network connections to suspicious countries. 3. Review word cloud to see the most common Sysmon event triggered during a given timeframe. 4. Identify the start time and end time of any suspicious activity. c. Query other Sysmon events that occurred around the same timeframe of any suspicious events identified to investigate. d. Create a histogram of Sysmon events triggered by the ransomware samples based on the timeframe of start and end times of the ransomware process.
6.	Automate the identification of the suspicious activity identified in Step 5 by creating new queries, dashboards, and alerts.
7.	Repeat this cyclical method to improve the Sysmon configuration and to implement additional analytics in order to identify suspicious activity more quickly and effectively to answer the research question.

The first step in the method was to install and configure Sysmon on the endpoint system. Next, a correlation server was setup to receive and consolidate the Sysmon logs for analysis. The implementation was verified to confirm that Sysmon was logging the desired events. Ransomware samples were then executed in the test environment to simulate an incident. After each sample was executed, Sysmon events were analyzed to identify suspicious activity. The analytic techniques were iteratively developed as the entire research study was conducted. Suspicious activity was identified through alerting, dashboards, and queries based on observed behavior and Sysmon events triggered by the ransomware dataset. Automated alerting drew attention to events suspected to indicate the presence of ransomware without requiring manual intervention. Data was enriched with metadata from VirusTotal to identify potentially malicious processes that were executed (Hayes, 2016). Since ransomware encrypts user files, the number of files created was analyzed to identify a large number of files created in a short amount of time. Ransomware typically rewrites the file extension, so file extensions of files created were analyzed to look for known ransomware extensions (MalwareBytes, n.d.). Malware has been known to change the timestamp of a file to help the new file blend in with other files on the system (Knutson, 2016). This procedure is known as "timestomping". Sysmon event ID 2 indicates that a process changed a file creation time. These events were analyzed to automate the identification of timestomping. By automatically looking for multiple indicators, ransomware is less likely to evade detection and more likely to be identified quickly during an incident response.

Dashboards were reviewed to manually look for anomalous activity. Network connections were visualized on a map to pinpoint connections to suspicious or unexpected countries. Malware may initiate network connections in order to download additional capabilities, so any network connections to unexpected countries needs to be investigated. Word clouds were reviewed to determine the most common Sysmon events triggered. By understanding how normal activity looks on a system, anomalous activity stood out and drew attention to additional logs that should be investigated. These analytic dashboards helped create a timeframe of suspicious activity on a host. The start time and end time of any suspicious activity was determined and used to focus analysis efforts. Once a time window was created, other Sysmon events that occurred during that timeframe were queried to look for additional abnormal activity. A histogram of the Sysmon events triggered by the ransomware sample was created based on the start and end time of the abnormal activity. This provided a profile of the ransomware sample based on Sysmon events. The ransomware profile helped provide details regarding what actions the ransomware sample took on the infected system. This knowledge helps an incident responder identify the source of the infection in order to clean it up, as well as identify infections on other systems.

As mentioned previously, the method artifact was developed using an iterative approach as described in Chapter 3. Once suspicious activity was identified, a process was determined to automate the finding of that activity in the future. Additional alerts, dashboards, and queries were created to improve future analysis efforts. This cycle was repeated over each ransomware sample to improve the Sysmon configuration and to implement additional analytics to identify suspicious activity more quickly and more effectively. Trial and error helped determine which analytic techniques were useful in identifying ransomware. The resulting method provides valuable guidance to organizations desiring to analyze Sysmon events to identify ransomware. This method fills a gap in the community by providing precise steps to follow to analyze Sysmon logs during incident response and to profile ransomware based on the Sysmon events it triggers.

Artifact 2: An Instantiation of the Method

This research created a comprehensive instantiation of the method described above. The instantiation is a tangible example of how to apply the method in an actual environment. As March & Smith state in their often-cited design science paper, the instantiation serves as a "realization" of the method (1995). The instantiation was developed iteratively according to design science principles and the requirements introduced in Table 1. This section describes the how the instantiation was constructed and articulates the unique contributions developed within the instantiation. Implementation details are provided so that other researchers can implement the instantiation in their own environment to replicate or expand on the results obtained by this study. An architecture diagram of the implemented environment is introduced in Figure 1.

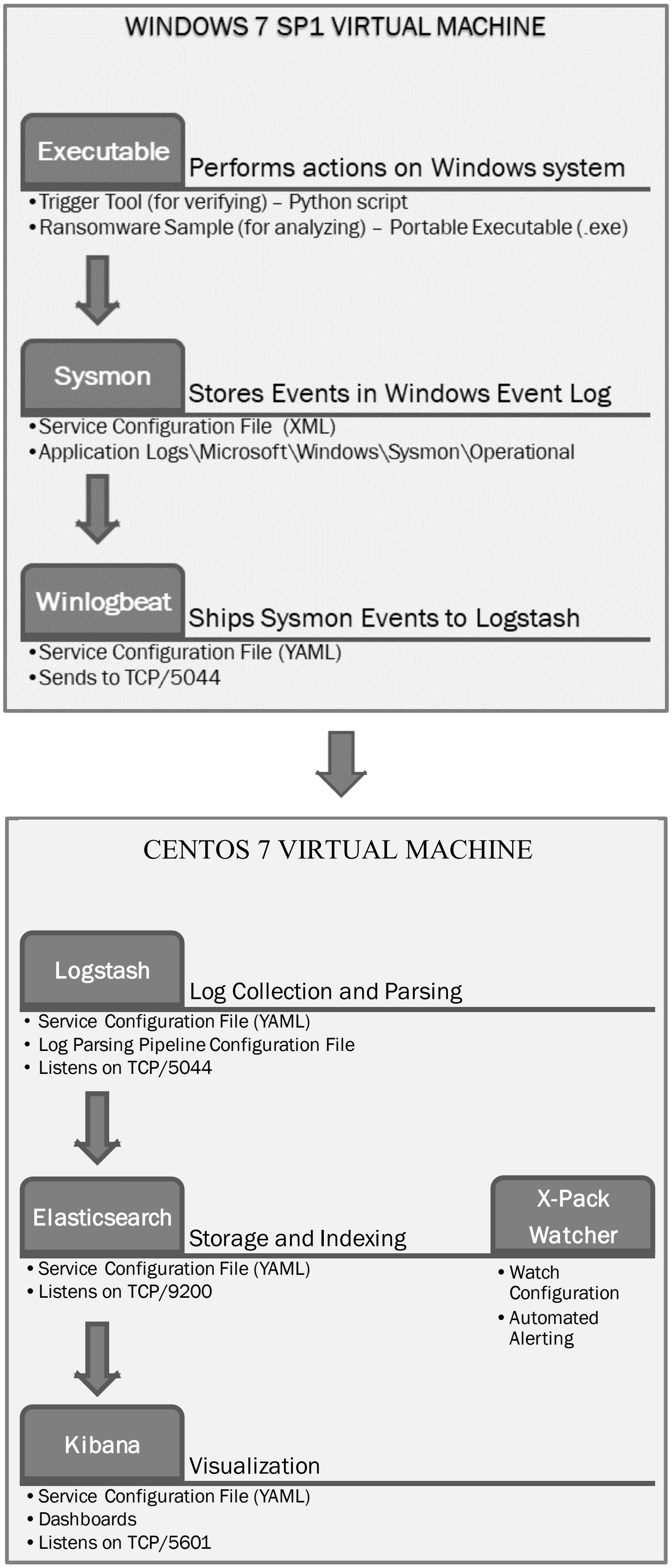

WINDOWS 7 SP1 VIRTUAL MACHINE
Executable
Performs actions on Windows system
•Trigger Tool (for verifying) – Python script
•Ransomware Sample (for analyzing) – Portable Executable (.exe)
Sysmon
Stores Events in Windows Event Log
•Service Configuration File (XML)
•Application Logs\Microsoft\Windows\Sysmon\Operational
Winlogbeat
Ships Sysmon Events to Logstash
•Service Configuration File (YAML)
•Sends to TCP/5044
CENTOS 7 VIRTUAL MACHINE
Logstash
Log Collection and Parsing
• Service Configuration File (YAML)
• Log Parsing Pipeline Configuration File
• Listens on TCP/5044
Elasticsearch
Storage and Indexing
• Service Configuration File (YAML)
• Listens on TCP/9200
X-Pack Watcher
• Watch Configuration
• Automated Alerting
Kibana
Visualization
• Service Configuration File (YAML)
• Dashboards
• Listens on TCP/5601

Figure 1. Architecture Diagram for Instantiation Artifact

The instantiation was developed on a laptop computer with virtualization software so that the results would be applicable to many organizations regardless of budget. The virtual environment consisted of a Windows endpoint system because Sysmon only runs on Windows operating systems. The endpoint system was configured with a Windows 7 Service Pack 1 operating system. The correlation server ran on a CentOS 7 operating system. An Elastic Stack served as the foundation for the log analysis platform. An Elastic Stack provides a generic data analysis platform. This research created a unique solution by leveraging the Elastic Stack to analyze Sysmon logs to identify ransomware.

The architecture can be understood by discussing the process flow that occurs from the time an action takes place on the endpoint system to the time it is visualized on the correlation server. Within the Windows 7 endpoint system, an executable performs an action on the operating system. The action is stored by the Sysmon service according to the Sysmon configuration file created by this research. On Windows 7, the Sysmon event details are stored in the Windows event log under `Application and Services Logs/Microsoft/Windows/Sysmon/Operational` (Russinovich & Garnier, 2017). The event log entries are periodically shipped across the network to the correlation server by the Winlogbeat service according to the configuration file defined in this research. The Logstash service on the correlation server receives the Sysmon events and parses the event. This research created a custom Logstash pipeline to enhance the data. Logstash enriches the data according to the custom pipeline and sends the results to Elasticsearch for storage. The Elasticsearch service stores and indexes the data so that it can be queried and retrieved. The Elasticsearch plugin X-Pack Watcher monitors the data. Four Watcher alerts were developed in this research to issue email alerts when specific suspicious activity is identified in the data. Kibana provides a visualization front-end to display the underlying data from Elasticsearch. This research created custom dashboards and specialized queries to enable focused analysis of the data. The entire architecture is provided to help organizations configure their own environments without having to start from scratch learning how all the components work together.

Open-source and freely available software was used to implement the instantiation whenever possible. This ensured that the solution is not out of reach for organizations due to budget reasons. The unique contributions of this research were made possible by configuring the open-source solution in such a way that unique analysis could be performed and documented. This study did not implement additional functionality into the open-source software, but rather leveraged open-source software to create a custom instantiation where analysis could take place and be documented for the incident response community to consume and scrutinize.

This research used VMware Workstation for the virtualization software. Although VMware Workstation is not free, other virtualization products exist that are free, such as VirtualBox (Oracle, 2018), which could replace VMware if cost was an issue. The endpoint system was running Sysmon v6.20 for logging, Winlogbeat v5.6.5 for shipping the logs to the server, and Python v3.6.3 to implement the trigger tool. The ransomware samples from the selected dataset were installed on the endpoint as well. The correlation server was a CentOS 7 virtual machine. It was running Logstash, Elasticsearch, Kibana, and the X-Pack plugin, all version 5.6.5. The Java version 8 was also used on the server because it is required by the Elastic Stack software. All the software used in the instantiation artifact is introduced in Table 8. These details are provided to facilitate others desiring to apply this research artifact within their own environments.

Table 8. Software Installed in the Instantiation Artifact

Base System:
Microsoft Windows 10.0.16299
VMware Workstation 12.5.9
Virtual Machine #1 (Endpoint System):
Microsoft Windows 7 Service Pack 1
Sysmon 6.20
Winlogbeat 5.6.5
Python 3.6.3
Ransomware samples from selected dataset
Virtual Machine #2 (Correlation Server):
CentOS 7
Logstash 5.6.5
Elasticsearch 5.6.5
Kibana 5.6.5
X-Pack 5.6.5
Java JDK 8

Sysmon

Sysmon documentation is very robust and provides details about various configuration options, but it does not provide guidance regarding which Sysmon events to monitor or how to analyze Sysmon events (Russinovich & Garnier, 2017). By default, Sysmon does not log every type of event so that it does not impact system performance when installed. Instead, the use of an Extensive Markup Language (XML) file is recommended to specify which events to include and which events to exclude. SwiftOnSecurity provides an extremely robust and well-documented configuration file that suits the needs of many organizations as discussed in Chapter 2 (2017).

SwiftOnSecurity advertises their configuration file as being a great starting point for understanding what events are possible for Sysmon to log (2017). SwiftOnSecurity is very clear that their configuration file will not capture every single event that malware could perform, but they state that it captures the most likely events that will be executed. The configuration settings are deliberately chosen to reduce performance impact while logging appropriate events (SwiftOnSecurity, 2017). Other security professionals also recommend the

SwiftOnSecurity configuration (Koopmann 2017, Russinovich 2017). For these reasons, the SwiftOnSecurity Sysmon configuration file was selected as a good fit for this research study. The SwiftOnSecurity configuration file is covered by a Creative Commons Attribution 4.0 license, which allows use to “privatize, fork, edit, teach, publish, or deploy for commercial use - with attribution in the text” (SwiftOnSecurity, 2017). Starting with the SwiftOnSecurity configuration file, this research iteratively modified the SwiftOnSecurity Sysmon configuration file to meet the needs of this research study.

Sysmon installation details are provided to guide others in how to implement Sysmon on their systems. Sysmon was installed in the instantiation using the command line introduced in Figure 2. The Sysmon configuration file was iteratively updated as this study was conducted. The final version of the Symon configuration file utilizes the recommendations from SwiftOnSecurity and also enables logging of the trigger tool artifact created later to confirm that all Sysmon events could be logged. The final version of the file is found in Appendix B. The Sysmon installation was verified by looking for the Sysmon service in the list of running services and by observing the Sysmon events in the event log `Application and Services Logs/Microsoft/Windows/Sysmon/Operational` (Russinovich & Garnier, 2017).

```
Sysmon.exe -i c:\Users\user1\Desktop\code\sysmonconfig-modified.xml
```

Figure 2. Command to Install Sysmon

The Elastic Stack

The Elastic Stack is an open source data analysis framework that was used to analyze the data in this study. The purpose of the Elastic Stack is to analyze data. Therefore, the usage of Elastic Stack for data analysis was not unique. However, fine-tuning the configuration of the Elastic Stack to enrich Sysmon logs and profile ransomware was a unique contribution of this research study. All of the configuration details used to create the Elastic Stack deployed in the instantiation are discussed below so that other researchers may replicate this study.

The Elastic Stack, formerly referred to as the ELK stack, consisted of Beats, Logstash, Elasticsearch, and Kibana (Langlois, 2016). All of these services resided on the same correlation server virtual machine, but the implementation could be expanded to multiple servers to address additional resource and storage needs. An online tutorial, along with Elastic documentation, was useful in configuring a generic Elastic Stack on the CentOS 7 server (Arul, 2017; Elastic, 2018). Once a generic Elastic Stack was working, the services were customized to suit the needs of this research study.

Winlogbeat was used to ship Sysmon logs from the endpoint to the correlation server using a Transmission Control Protocol (TCP) network connection over port 5044. Winlogbeat ran as a service installed on the endpoint system. Winlogbeat was installed using the PowerShell script `install-service-winlogbeat.ps1`, which is included in the Winlogbeat installation file. Winlogbeat uses a configuration file to specify which logs should be shipped to which locations. The Winlogbeat configuration file used in this study is found in Appendix E. Finally, Winlogbeat was started with the PowerShell command "`Start-Service winlogbeat`". The Winlogbeat service was verified to be running by checking the list of running services in Windows.

Logstash was used to parse and enrich the Sysmon logs what were received from Winlogbeat and send the results to Elasticsearch for storage. The implemented instantiation required Logstash listen on TCP port 5044 to receive the data from Winlogbeat on the endpoint system. The Logstash configuration file used in this study to configure the Logstash service is found in Appendix J. A custom developed Logstash pipeline defined how the Sysmon logs were parsed. The custom pipeline enriched the Sysmon data with additional metadata from VirusTotal and from custom Logstash dictionary files.

The Logstash pipeline was a key component of this research. It was iteratively developed as additional functionality was needed. The final pipeline configuration used in this research is found in Appendix F and is discussed below. Every Logstash pipeline contains an input section to define where the logs are coming from, a filter section to process, change, and enrich the logs as needed, and an output section to define where the logs will be sent. This study provides a detailed and unique filter section for the Logstash pipeline to perform data enrichment and custom functionality needed for the analysis developed in the method artifact.

The Logstash pipeline developed by this study implemented several tasks to enrich the Sysmon log data. The full pipeline in Appendix F includes detailed comments to document the functionality of the pipeline. The pipeline extracted SHA-256 hashes from the Sysmon events so that hashes could be queried individually and to identify specific processes. The pipeline performed Internet Protocol (IP) address geolocation on network connections. This enabled incident responders to identify the country of origin of a network connection in order to detect anomalous or unexpected network connections. The pipeline extracted file extensions from file names so that file extensions could be queried directly and common ransomware extensions could be flagged. The pipeline matched events relating to the start of a process and the termination of the process to determine the longevity of the process. Knowing when a process started and ended helped provide a timeframe to focus analytic efforts. The pipeline translated Sysmon event ID numbers into a textual description to enable quick understanding of Sysmon events while analyzing logs. The pipeline implemented a whitelist to track known good processes, and queried VirusTotal whenever an unknown process was executed. A public VirusTotal Application Programming Interface (API) key was used to look up unknown executables and determine if a process was malicious (VirusTotal, 2018b). The VirusTotal results were stored with the logs so they could be used during analytic tasks to help analysts determine if a process was malicious and if so, to determine what type of malware was executed.

One capability that made the Logstash pipeline created in this study unique was the creation of three custom dictionary files to enrich the Sysmon data. The first custom dictionary mapped Sysmon event ID numbers to a textual description of each event. This allowed for easier analysis when viewing Symon entries in Kibana. The Sysmon event dictionary file is found in Appendix G. The second custom dictionary identified commonly used ransomware extensions to tag suspicious Sysmon events if they contained a suspicious file extension. The file extensions were obtained from a MalwareBytes article (n.d.) that enumerated several ransomware extensions, along with observed file extensions discovered during this study. The ransomware extension dictionary file is found in Appendix H. The third custom dictionary provided a whitelist of known good file hashes and file paths. This dictionary was used to reduce the number of VirusTotal lookups executed by the system. By whitelisting known executables, only unknown executables were looked up in VirusTotal.

This saved time and network bandwidth. The application whitelist dictionary file is found in Appendix I. After all of the Sysmon events were processed and enriched through the Logstash pipeline, the results were then sent to Elasticsearch according to the output section of the pipeline.

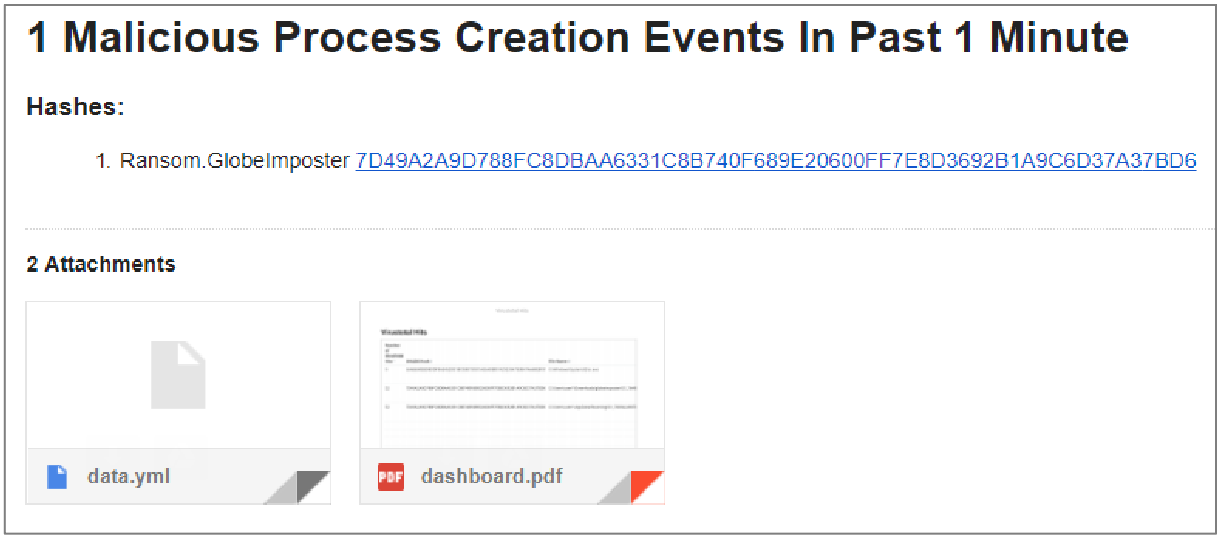
Elasticsearch was used for storing and indexing the results received from Logstash. The Elasticsearch configuration was very basic and did not require any unique customizations to meet the needs of this research study. Elasticsearch listened on TCP port 9200. The Elasticsearch configuration file used in this study is found in Appendix K. Elasticsearch only ran on one node in this study, but it could be expanded to work as a cluster with a master node and slave nodes to improve performance and speed. In a production environment, the data transferred between Logstash and Elasticsearch should be encrypted to maintain privacy.

A X-Pack plugin called Watcher was used to provide automated alerting in the instantiation. Watcher was automatically enabled once the X-Pack package from Elastic was installed with Elasticsearch and Kibana (Elastic, 2018). A trial license was used in this research. Watcher alerts have multiple alerting capabilities, but only email alerting was used in this research. Four unique Watcher alerts were created during this research to implement automated alerting and identify suspicious behavior more quickly. Watcher alerts are discussed in more detail in the next section.

Kibana was used as a web-based front-end to analyze and visualize results. Kibana listened for web connections on TCP port 5061. The Kibana configuration file used in this study is found in Appendix L. Kibana created an index of fields based on the data found in Elasticsearch. This index specifies exactly what data fields are available to query against, in addition to the data type. The final instantiation used in this research study contained 716 fields in the index. The index can be found in Appendix M. Kibana was a critical tool used to understand the Sysmon data and provide visualizations that were useful during incident response. This research created custom dashboards and custom queries to enable deep analysis of the Sysmon events. Dashboards and queries are discussed in more detail in later sections.

Alerts

Alerting was implemented using Watcher, which is a component of the Elastic X-Pack installation. This research used a trial license for X-Pack. Watcher alerts were configured to provide automated alerting when suspicious activity was identified in the Sysmon logs. Four Watcher alerts were created in this study. The first Watcher alert looked for processes marked as malicious by VirusTotal. The alert ran every minute, and if any malicious processes were found during that minute, an email alert was sent. The email contains the SHA-256 hash of the process, along with the data that triggered the alert and the dashboard summary so that analysts can quickly identify where the malicious process was executed. This alert depended on the VirusTotal data enrichment provided by the Logstash pipeline. The malicious process watcher configuration code is found in Appendix N. An example of the email alert triggered when ransomware sample G1 was executed on the endpoint system is introduced in Figure 3.



1 Malicious Process Creation Events In Past 1 Minute

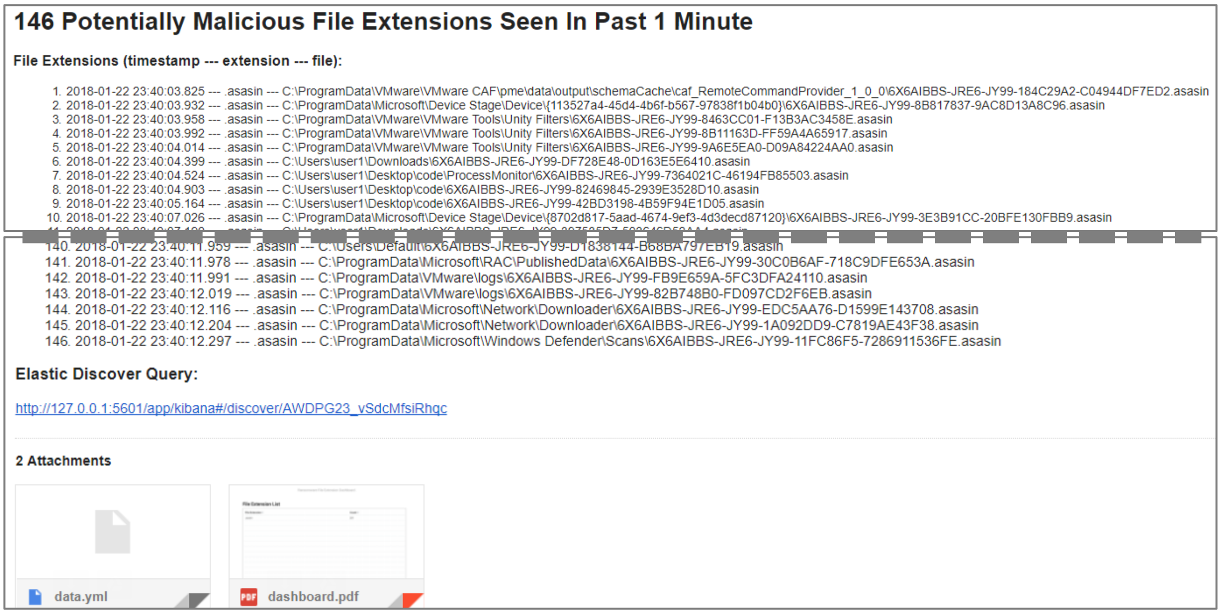
Hashes:

1. Ransom.GlobeImposter 7D49A2A9D788FC8DBAA6331C8B740F689E20600FF7E8D3692B1A9C6D37A37BD6

2 Attachments

Figure 3. Email Alert for Process Labeled Malicious by VirusTotal (Sample G1)

The second Watcher alert looked for commonly used ransomware file extensions in the Sysmon logs. The alert ran every minute, and if any suspicious extensions were found, an email alert was sent. The email contained a list of the malicious extension and file path, along with attachments containing the data that triggered the alert and the dashboard summary to help analysts quickly identify the suspicious activity. The malicious process Watcher configuration code is found in Appendix O. An example of the email alert triggered when ransomware sample L1 was executed on the endpoint system is introduced in Figure 4. The figure has been truncated for brevity to show both the beginning and ending of the email alert. The email alert shows that this sample generated 146 Sysmon events with the file extension "`.asasin`" within 1 minute.



146 Potentially Malicious File Extensions Seen In Past 1 Minute

File Extensions (timestamp --- extension --- file):

```
  1. 2018-01-22 23:40:03.825 --- .asasin --- C:\ProgramData\VMware\VMware CAF\pme\data\output\schemaCache\caf_RemoteCommandProvider_1_0_0\6X6AIBBS-JRE6-JY99-184C29A2-C04944DF7ED2.asasin
  2. 2018-01-22 23:40:03.932 --- .asasin --- C:\ProgramData\Microsoft\Device Stage\Device\{113527a4-45d4-4b6f-b567-97838f1b04b0}\6X6AIBBS-JRE6-JY99-8B817837-9AC8D13A8C96.asasin
  3. 2018-01-22 23:40:03.958 --- .asasin --- C:\ProgramData\VMware\VMware Tools\Unity Filters\6X6AIBBS-JRE6-JY99-8463CC01-F13B3AC3458E.asasin
  4. 2018-01-22 23:40:03.992 --- .asasin --- C:\ProgramData\VMware\VMware Tools\Unity Filters\6X6AIBBS-JRE6-JY99-8B11163D-FF59A4A65917.asasin
  5. 2018-01-22 23:40:04.014 --- .asasin --- C:\ProgramData\VMware\VMware Tools\Unity Filters\6X6AIBBS-JRE6-JY99-9A6E5EA0-D09A84224AA0.asasin
  6. 2018-01-22 23:40:04.399 --- .asasin --- C:\Users\user1\Downloads\6X6AIBBS-JRE6-JY99-DF728E48-0D163E5E6410.asasin
  7. 2018-01-22 23:40:04.524 --- .asasin --- C:\Users\user1\Desktop\code\ProcessMonitor\6X6AIBBS-JRE6-JY99-7364021C-46194FB85503.asasin
  8. 2018-01-22 23:40:04.903 --- .asasin --- C:\Users\user1\Desktop\code\6X6AIBBS-JRE6-JY99-82469845-2939E3528D10.asasin
  9. 2018-01-22 23:40:05.164 --- .asasin --- C:\Users\user1\Desktop\code\6X6AIBBS-JRE6-JY99-42BD3198-4B59F94E1D05.asasin
 10. 2018-01-22 23:40:07.026 --- .asasin --- C:\ProgramData\Microsoft\Device Stage\Device\{8702d817-5aad-4674-9ef3-4d3decd87120}\6X6AIBBS-JRE6-JY99-3E3B91CC-20BFE130FBB9.asasin
140. 2018-01-22 23:40:11.959 --- .asasin --- C:\Users\Default\6X6AIBBS-JRE6-JY99-D1838144-B68BA797EB19.asasin
141. 2018-01-22 23:40:11.978 --- .asasin --- C:\ProgramData\Microsoft\RAC\PublishedData\6X6AIBBS-JRE6-JY99-30C0B6AF-718C9DFE653A.asasin
142. 2018-01-22 23:40:11.991 --- .asasin --- C:\ProgramData\VMware\logs\6X6AIBBS-JRE6-JY99-FB9E659A-5FC3DFA24110.asasin
143. 2018-01-22 23:40:12.019 --- .asasin --- C:\ProgramData\VMware\logs\6X6AIBBS-JRE6-JY99-82B748B0-FD097CD2F6EB.asasin
144. 2018-01-22 23:40:12.116 --- .asasin --- C:\ProgramData\Microsoft\Network\Downloader\6X6AIBBS-JRE6-JY99-EDC5AA76-D1599E143708.asasin
145. 2018-01-22 23:40:12.204 --- .asasin --- C:\ProgramData\Microsoft\Network\Downloader\6X6AIBBS-JRE6-JY99-1A092DD9-C7819AE43F38.asasin
146. 2018-01-22 23:40:12.297 --- .asasin --- C:\ProgramData\Microsoft\Windows Defender\Scans\6X6AIBBS-JRE6-JY99-11FC86F5-7286911536FE.asasin
```

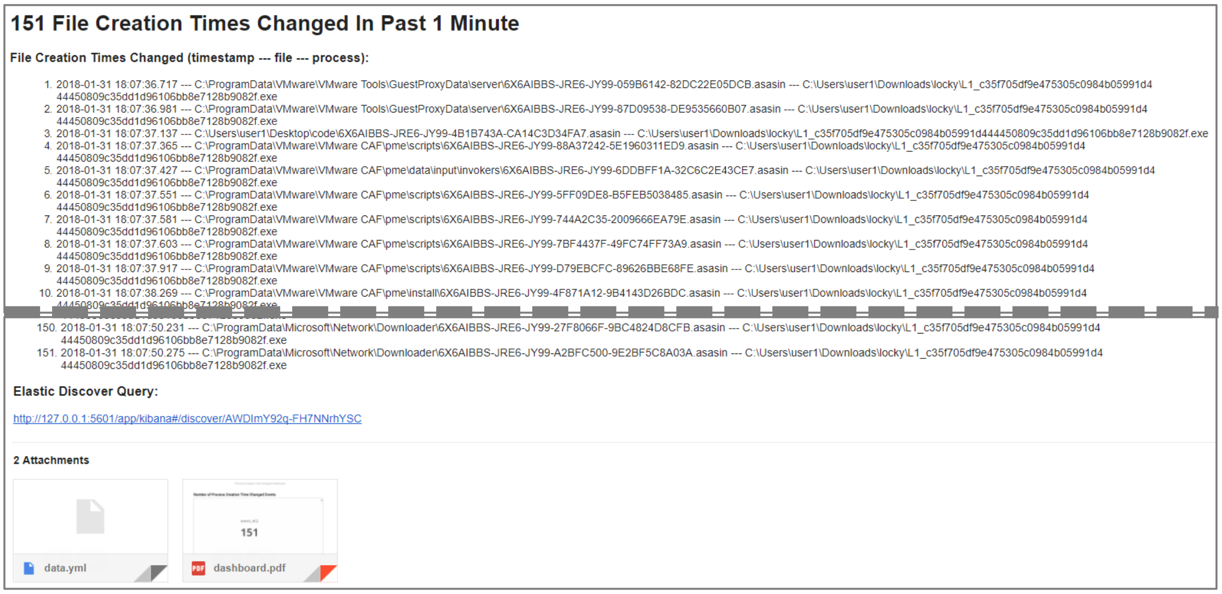
Elastic Discover Query:

http://127.0.0.1:5601/app/kibana#/discover/AWDPG23_vSdcMfsiRhqc

2 Attachments

Figure 4. Email Alert for Malicious File Extensions (Sample L1)

The third Watcher alert looked for instances when a process changed a file creation time. Malware sometimes changes the file creation time of files it creates or modifies in order to make the files blend into the filesystem so that the user does not know the files were modified. The alert runs every minute and looks for Sysmon event ID 2, which indicated that a file's creation time was changed by a process. If any of those Sysmon events occurred, an email alert is sent. The email alert listed the time of the event, the file name that was affected, and the process name that changed the file creation time. Attachments were also included containing the data that triggered the alert and a dashboard summary so that an analyst could quickly identify the activity. The malicious process watcher configuration code is found in Appendix P. An excerpt of the email alert triggered when ransomware sample L1 was executed on the endpoint system is introduced in Figure 5. The figure has been truncated for brevity to show both the beginning and ending of the email alert. The email alert shows that this sample changed the creation time of 151 files in one minute.



151 File Creation Times Changed In Past 1 Minute

File Creation Times Changed (timestamp --- file --- process):

```
1. 2018-01-31 18:07:36.717 --- C:\ProgramData\VMware\VMware Tools\GuestProxyData\server\6X6AIBBS-JRE6-JY99-059B6142-82DC22E05DCB.asasin --- C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d4
   44450809c35dd1d96106bb8e7128b9082f.exe
2. 2018-01-31 18:07:36.981 --- C:\ProgramData\VMware\VMware Tools\GuestProxyData\server\6X6AIBBS-JRE6-JY99-87D09538-DE9535660B07.asasin --- C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d4
   44450809c35dd1d96106bb8e7128b9082f.exe
3. 2018-01-31 18:07:37.137 --- C:\Users\user1\Desktop\code\6X6AIBBS-JRE6-JY99-4B1B743A-CA14C3D34FA7.asasin --- C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f.exe
4. 2018-01-31 18:07:37.365 --- C:\ProgramData\VMware\VMware CAF\pme\scripts\6X6AIBBS-JRE6-JY99-88A37242-5E1960311ED9.asasin --- C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d4
   44450809c35dd1d96106bb8e7128b9082f.exe
5. 2018-01-31 18:07:37.427 --- C:\ProgramData\VMware\VMware CAF\pme\data\input\invokers\6X6AIBBS-JRE6-JY99-6DDBFF1A-32C6C2E43CE7.asasin --- C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d4
   44450809c35dd1d96106bb8e7128b9082f.exe
6. 2018-01-31 18:07:37.551 --- C:\ProgramData\VMware\VMware CAF\pme\scripts\6X6AIBBS-JRE6-JY99-5FF09DE8-B5FEB5038485.asasin --- C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d4
   44450809c35dd1d96106bb8e7128b9082f.exe
7. 2018-01-31 18:07:37.581 --- C:\ProgramData\VMware\VMware CAF\pme\scripts\6X6AIBBS-JRE6-JY99-744A2C35-2009666EA79E.asasin --- C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d4
   44450809c35dd1d96106bb8e7128b9082f.exe
8. 2018-01-31 18:07:37.603 --- C:\ProgramData\VMware\VMware CAF\pme\scripts\6X6AIBBS-JRE6-JY99-7BF4437F-49FC74FF73A9.asasin --- C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d4
   44450809c35dd1d96106bb8e7128b9082f.exe
9. 2018-01-31 18:07:37.917 --- C:\ProgramData\VMware\VMware CAF\pme\scripts\6X6AIBBS-JRE6-JY99-D79EBCFC-89626BBE68FE.asasin --- C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d4
   44450809c35dd1d96106bb8e7128b9082f.exe
10. 2018-01-31 18:07:38.269 --- C:\ProgramData\VMware\VMware CAF\pme\install\6X6AIBBS-JRE6-JY99-4F871A12-9B4143D26BDC.asasin --- C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d4
   44450809c35dd1d96106bb8e7128b9082f.exe
150. 2018-01-31 18:07:50.231 --- C:\ProgramData\Microsoft\Network\Downloader\6X6AIBBS-JRE6-JY99-27F8066F-9BC4824D8CFB.asasin --- C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d4
     44450809c35dd1d96106bb8e7128b9082f.exe
151. 2018-01-31 18:07:50.275 --- C:\ProgramData\Microsoft\Network\Downloader\6X6AIBBS-JRE6-JY99-A2BFC500-9E2BF5C8A03A.asasin --- C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d4
     44450809c35dd1d96106bb8e7128b9082f.exe
```

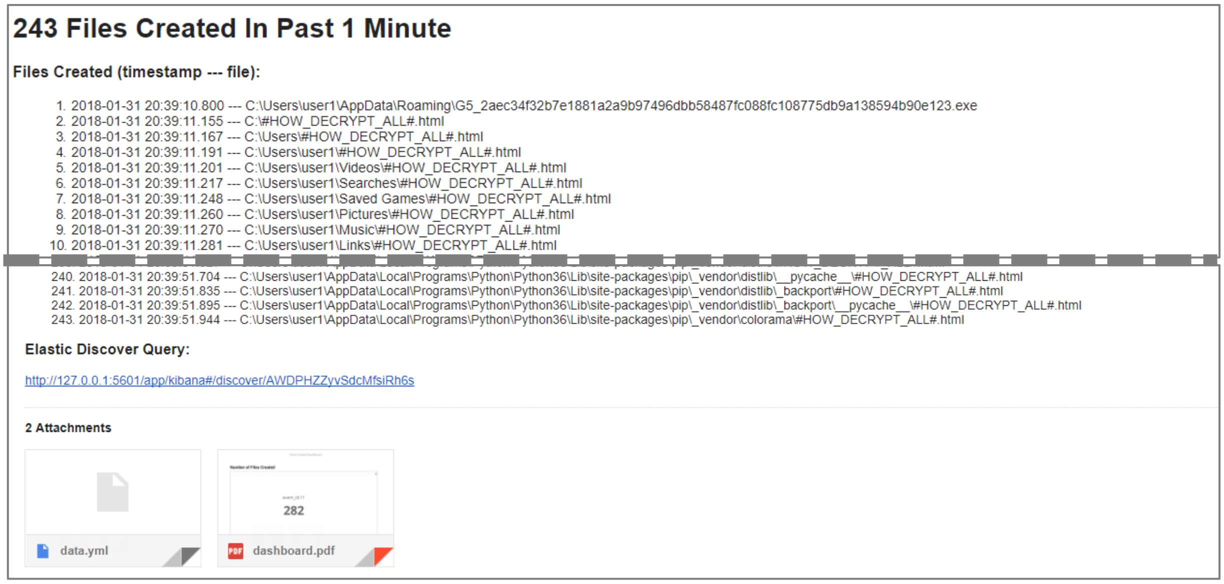
Elastic Discover Query:

http://127.0.0.1:5601/app/kibana#/discover/AWDImY92q-FH7NNrhYSC

2 Attachments

Figure 5. Email Alert for Creation Time Changed on a File (Sample L1)

The fourth Watcher alert looked for too many files created on the system in a short timeframe. The alert ran every minute and looked for Sysmon event ID 11, which indicated a file was created. The watcher alerted if more than 240 files were created in a minute. The email alert contained a list of files created, along with attachments containing the data that triggered the alert and a dashboard summary to help incident responder identify the suspicious activity quickly. The malicious process watcher configuration code is found in Appendix Q. An excerpt of the email alert sent by this watcher when ransomware sample G5 was executed on the endpoint system is introduced in Figure 6. The figure has been truncated for brevity to show both the beginning and ending of the email alert. The email alert shows that this sample created 243 new files in one minute.



243 Files Created In Past 1 Minute

Files Created (timestamp --- file):

```
  1. 2018-01-31 20:39:10.800 --- C:\Users\user1\AppData\Roaming\G5_2aec34f32b7e1881a2a9b97496dbb58487fc088fc108775db9a138594b90e123.exe
  2. 2018-01-31 20:39:11.155 --- C:\#HOW_DECRYPT_ALL#.html
  3. 2018-01-31 20:39:11.167 --- C:\Users\#HOW_DECRYPT_ALL#.html
  4. 2018-01-31 20:39:11.191 --- C:\Users\user1\#HOW_DECRYPT_ALL#.html
  5. 2018-01-31 20:39:11.201 --- C:\Users\user1\Videos\#HOW_DECRYPT_ALL#.html
  6. 2018-01-31 20:39:11.217 --- C:\Users\user1\Searches\#HOW_DECRYPT_ALL#.html
  7. 2018-01-31 20:39:11.248 --- C:\Users\user1\Saved Games\#HOW_DECRYPT_ALL#.html
  8. 2018-01-31 20:39:11.260 --- C:\Users\user1\Pictures\#HOW_DECRYPT_ALL#.html
  9. 2018-01-31 20:39:11.270 --- C:\Users\user1\Music\#HOW_DECRYPT_ALL#.html
 10. 2018-01-31 20:39:11.281 --- C:\Users\user1\Links\#HOW_DECRYPT_ALL#.html
240. 2018-01-31 20:39:51.704 --- C:\Users\user1\AppData\Local\Programs\Python\Python36\Lib\site-packages\pip\_vendor\distlib\__pycache__\#HOW_DECRYPT_ALL#.html
241. 2018-01-31 20:39:51.835 --- C:\Users\user1\AppData\Local\Programs\Python\Python36\Lib\site-packages\pip\_vendor\distlib\_backport\#HOW_DECRYPT_ALL#.html
242. 2018-01-31 20:39:51.895 --- C:\Users\user1\AppData\Local\Programs\Python\Python36\Lib\site-packages\pip\_vendor\distlib\_backport\__pycache__\#HOW_DECRYPT_ALL#.html
243. 2018-01-31 20:39:51.944 --- C:\Users\user1\AppData\Local\Programs\Python\Python36\Lib\site-packages\pip\_vendor\colorama\#HOW_DECRYPT_ALL#.html
```

Elastic Discover Query:

http://127.0.0.1:5601/app/kibana#/discover/AWDPHZZyvSdcMfsiRh6s

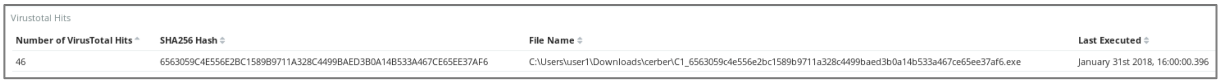
2 Attachments

Figure 6. Email Alert for Too Many Files Created (Sample G5)

Dashboards

Kibana dashboards were created to quickly visualize important data. The dashboards were designed to contain charts, graphs, maps, word clouds, and metrics to present the Sysmon logs in a meaningful way. The dashboards could be filtered based on time and data requirements to enhance the information conveyed by the dashboard. Several dashboards were created during this research to iteratively drive the development of the resulting artifacts and enhance analysis of Sysmon logs. These dashboards contribute to the overall goal of this research to provide guidance on how improve Sysmon analysis.

A custom dashboard was created to view processes labeled malicious by VirusTotal. Any potentially malicious processes should be investigated, and this dashboard was critical in pinpointing which processes are suspicious. This dashboard provided a quick starting place for analysis by providing a timeframe for when suspicious activity occurred. This dashboard was also attached to the related Watcher email alerts discussed in the previous section. The dashboard after running Cerber sample C1 on the endpoint system is introduced in Figure 7. In this scenario, 46 VirusTotal scan engines reported the process as malicious. VirusTotal results typically contain results from approximately 60 virus scanning engines. 46 VirusTotal hits are a strong indicator that the process is malicious.

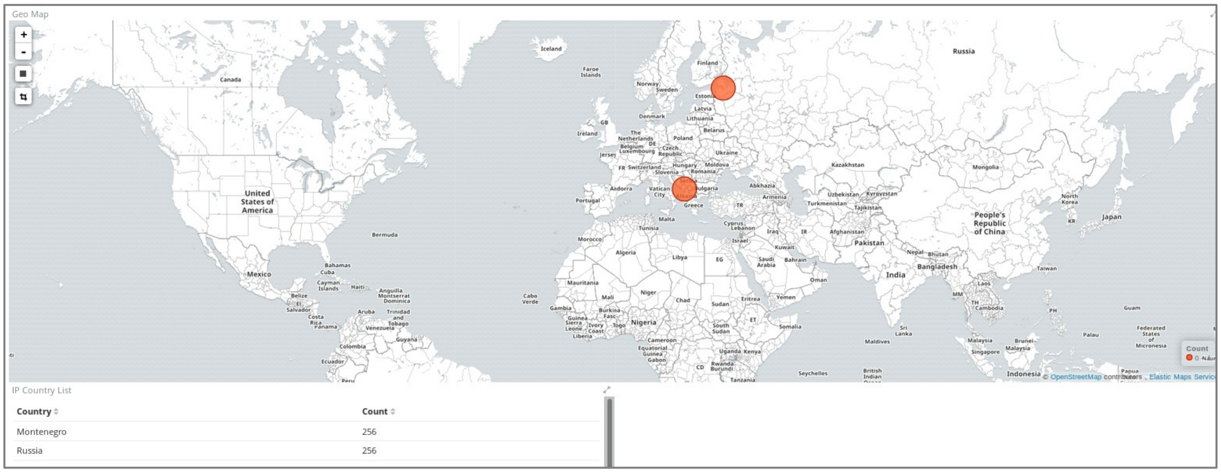


Virustotal Hits

Number of VirusTotal Hits	SHA256 Hash	File Name	Last Executed
46	6563059C4E556E2BC1589B9711A328C4499BAED3B0A14B533A467CE65EE37AF6	C:\Users\user1\Downloads\cerber\C1_6563059c4e556e2bc1589b9711a328c4499baed3b0a14b533a467ce65ee37af6.exe	January 31st 2018, 16:00:00.396

Figure 7. Dashboard for Viewing Processes Labeled Malicious by VirusTotal (Sample C1)

A custom dashboard was designed to visualize network connections. The network connections were pinpointed on a map based on the number of connections to each location. A chart was displayed underneath the map to summarize how many connections occurred to each country. This dashboard was dependent on Logstash enriching Internet Protocol (IP) addresses with geolocation data using the `geoip` Logstash filter as configured in the Logstash pipeline created by this research. The dashboard results after running Cerber ransomware sample C1 on the endpoint system is introduced in Figure 8. In this scenario, ransomware sample C1 created network connections to Montenegro and Russia. If an analyst sees unexpected network connections to suspicious countries, then they should continue the analysis process by exploring the network connection, including the process that initiated the connection and any other activity that occurred during the suspicious timeframe.



IP Country List

Country	Count
Montenegro	256
Russia	256

Figure 8. Dashboard for Viewing Network Connections (Sample C1)

A custom dashboard was created to view the frequency of Sysmon events using a word cloud. This provides a quick understanding of the Sysmon events that occurred most often during a given timeframe. The dashboard results after running Cerber ransomware sample C1 on the endpoint system is introduced in Figure 9. The results indicate that ransomware sample C1 primarily initiates network connections, but the other actions performed by the ransomware sample are also displayed, such as creating files and processes.

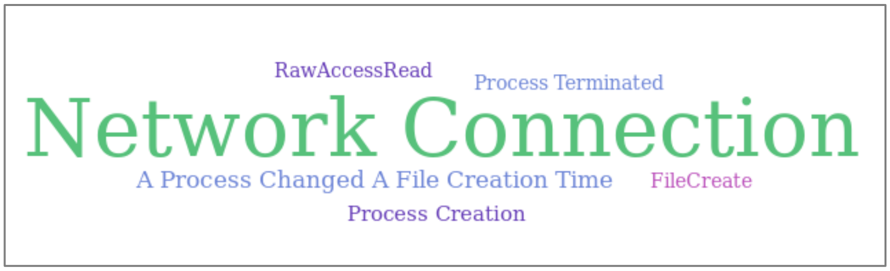
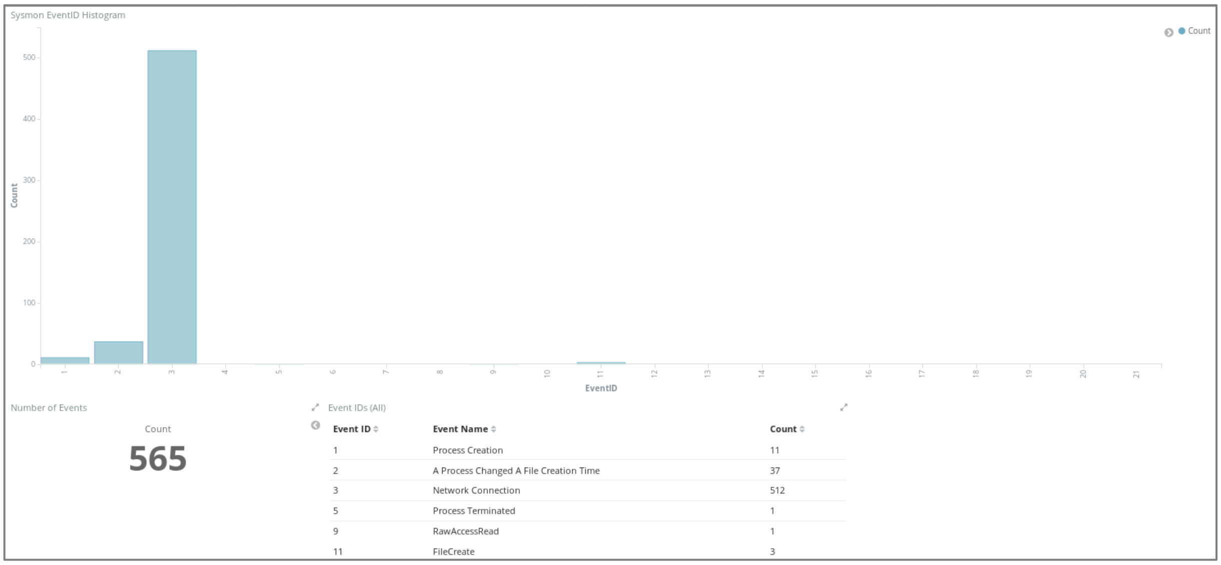


Figure 9. Dashboard for Viewing Sysmon Events Word Cloud (Sample C1)

Another dashboard was created to display a histogram of Sysmon events triggered during a given timeframe. This provides more details than the word cloud dashboard created above because it includes numerical data regarding how many Sysmon events of each type were triggered. This dashboard was useful for profiling the ransomware samples, a process that is discussed on page 54. The histogram dashboard after running sample C1 on the endpoint system is introduced in Figure 10. A total of 565 Sysmon events were triggered by this sample, consisting of six unique Sysmon event types. The network connection event was triggered most often, for a total of 512 network events. These details augment the understanding that was gained by the network connection and word cloud dashboards discussed previously as it now provides numerical data to provide context on how the ransomware sample behaved.



Number of Events

Count

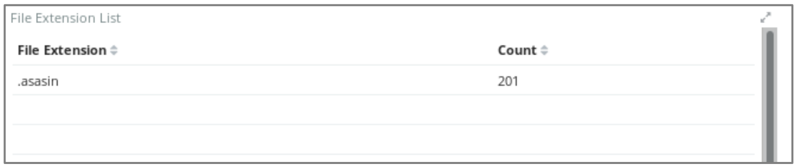
565

Event IDs (All)

Event ID	Event Name	Count
1	Process Creation	11
2	A Process Changed A File Creation Time	37
3	Network Connection	512
5	Process Terminated	1
9	RawAccessRead	1
11	FileCreate	3

Figure 10. Dashboard for Viewing Sysmon Events Histogram (Sample C1)

A ransomware file extension dashboard was created to view known ransomware file extensions. This dashboard was also attached to the related Watcher alert discussed above. The file extensions were obtained from a MalwareBytes article (n.d.) that enumerated several ransomware extensions, along with observed file extensions discovered during this study. During the ransomware profiling phase of this research, this dashboard helped determine that not all ransomware triggers the file creation Sysmon events. This observation is discussed in more detail in the Ransomware Profiling section below. The dashboard after executing ransomware sample L1 on the endpoint system is introduced in Figure 11. The dashboard shows that the ransomware sample used the file extension "`.asasin`" 201 times in the Sysmon logs.

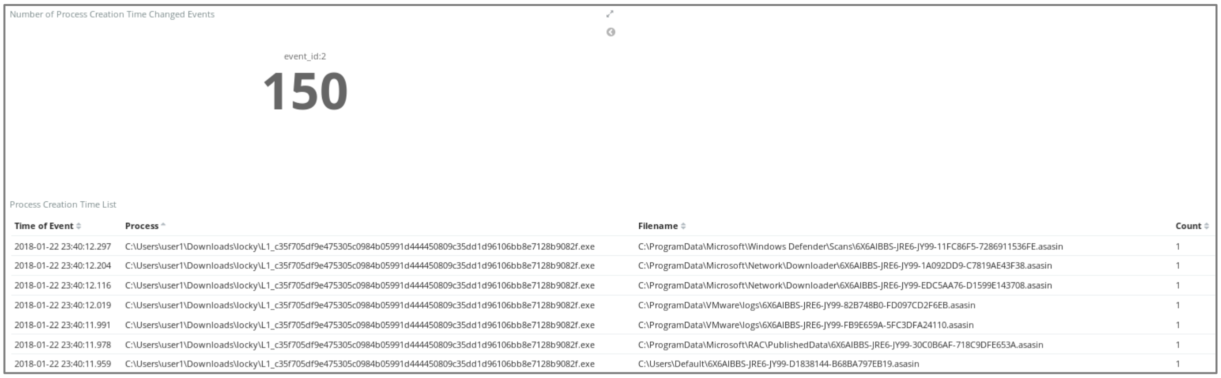


File Extension List

File Extension	Count
.asasin	201

Figure 11. Dashboard for Viewing Ransomware File Extensions (Sample L1)

A custom dashboard was created to visualize the Sysmon events triggered when a process modified the creation time of a file. This dashboard was also attached to the related Watcher email alert, as discussed in the previous section. This dashboard showed the time of the event, the process name that changed the creation time, and the filename of the file that was changed. The dashboard after executing ransomware sample L1 on the endpoint system is introduced in Figure 12. The figure was truncated at the bottom for brevity. It demonstrates that the ransomware sample L1 changed the file time of 150 files, most of them ending with the file extension "`.asasin`". This was likely done to mask the time of the infection.



Number of Process Creation Time Changed Events

event_id:2

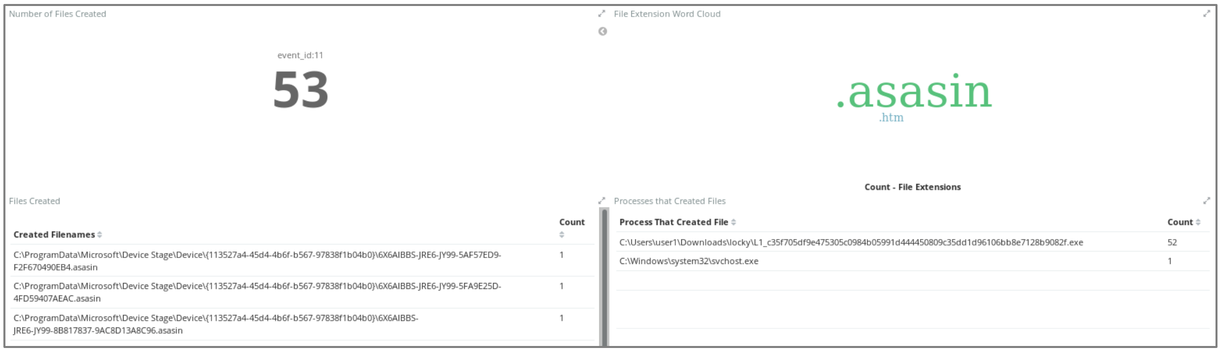
150

Process Creation Time List

Time of Event	Process	Filename	Count
2018-01-22 23:40:12.297	C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f.exe	C:\ProgramData\Microsoft\Windows Defender\Scans\6X6AIBBS-JRE6-JY99-11FC86F5-7286911536FE.asasin	1
2018-01-22 23:40:12.204	C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f.exe	C:\ProgramData\Microsoft\Network\Downloader\6X6AIBBS-JRE6-JY99-1A092DD9-C7819AE43F38.asasin	1
2018-01-22 23:40:12.116	C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f.exe	C:\ProgramData\Microsoft\Network\Downloader\6X6AIBBS-JRE6-JY99-EDC5AA76-D1599E143708.asasin	1
2018-01-22 23:40:12.019	C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f.exe	C:\ProgramData\VMware\logs\6X6AIBBS-JRE6-JY99-82B748B0-FD097CD2F6EB.asasin	1
2018-01-22 23:40:11.991	C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f.exe	C:\ProgramData\VMware\logs\6X6AIBBS-JRE6-JY99-FB9E659A-5FC3DFA24110.asasin	1
2018-01-22 23:40:11.978	C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f.exe	C:\ProgramData\Microsoft\RAC\PublishedData\6X6AIBBS-JRE6-JY99-30C0B6AF-718C9DFE653A.asasin	1
2018-01-22 23:40:11.959	C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f.exe	C:\Users\Default\6X6AIBBS-JRE6-JY99-D1838144-B68BA797EB19.asasin	1

Figure 12. Dashboard for Viewing Changes to File Creation Time (Sample L1)

Since ransomware creates newly encrypted files, monitoring file creation activity is useful in detecting ransomware. A custom dashboard was created to investigate files created on a system. This dashboard was also attached to the related Watcher email alert, as discussed in the previous section. File creations were logged by Sysmon event ID 11. The dashboard after executing ransomware sample L1 on the endpoint system is introduced in Figure 13. The figure is truncated at the bottom for brevity. The dashboard demonstrates that while the L1 sample was executing, 53 files were created. 52 of those files were created by the parent process itself. One file was created by svchost.exe, which was a process started by the ransomware process. Most of the created files ended with the file extension "`.asasin`". A keen observer will notice that the L1 sample notification files ended in "`.html`" and "`.bmp`", but those file extensions were not found in any file creation events in Sysmon logs. The Sysmon file creation event (event ID 11) was not triggered when sample L1 created the notification files. Because this research study does not investigate why Sysmon performs a certain way, this study cannot determine why the creation of those two files did not trigger Sysmon events.



Files Created

Created Filenames	Count
C:\ProgramData\Microsoft\Device Stage\Device\{113527a4-45d4-4b6f-b567-97838f1b04b0}\6X6AIBBS-JRE6-JY99-5AF57ED9-F2F670490EB4.asasin	1
C:\ProgramData\Microsoft\Device Stage\Device\{113527a4-45d4-4b6f-b567-97838f1b04b0}\6X6AIBBS-JRE6-JY99-5FA9E25D-4FD59407AEAC.asasin	1
C:\ProgramData\Microsoft\Device Stage\Device\{113527a4-45d4-4b6f-b567-97838f1b04b0}\6X6AIBBS-JRE6-JY99-8B817837-9AC8D13A8C96.asasin	1

Processes that Created Files

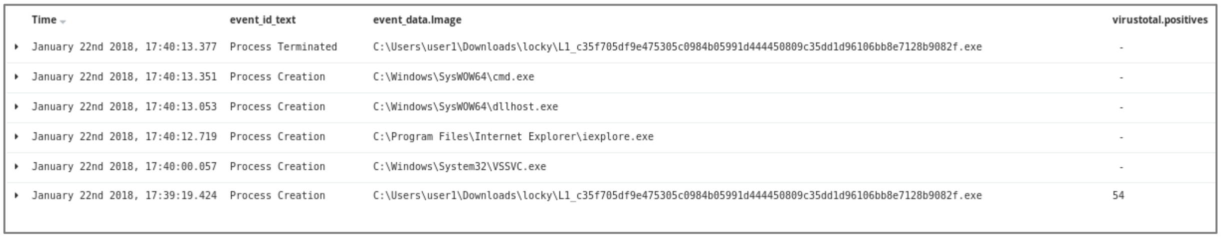
Process That Created File	Count
C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f.exe	52
C:\Windows\system32\svchost.exe	1

Figure 13. Dashboard for Viewing Created Files (Sample L1)

Queries

Kibana queries were used to search through the data to gain greater understanding of the data and look for specific events. Querying data is a critical skill during incident response, and it was essential throughout this research. Queries were filtered based on time and content as needed. Queries were saved so that useful searches could be repeated easily. The use of saved queries was an integral part in conducting the iterative analysis required to answer the research question.

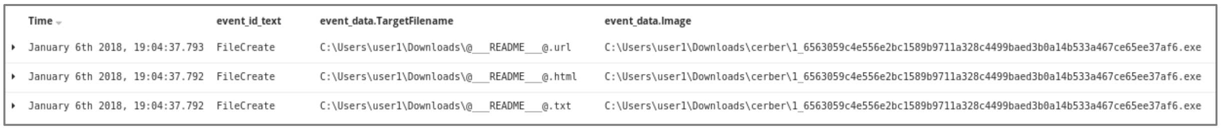
One of the queries created by this research focused on process creation and termination events. By viewing processes as they were created and terminated, the behavior of the samples could be observed. Some samples launched other processes to complete the ransomware tasks. The query result introduced in Figure 14 shows the process creation and termination Sysmon events that were triggered by ransomware sample L1. The L1 sample launched four additional processes. One of the processes launched was Internet Explorer, which was used to display the notification file "`asasin.html`" to inform the user that a ransom payment was required to decrypt the files. By reviewing the processes created by a ransomware sample, an incident responder can determine what additional actions the ransomware executed on the infected system.



Time	event_id_text	event_data.Image	virustotal.positives
January 22nd 2018, 17:40:13.377	Process Terminated	C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f.exe	-
January 22nd 2018, 17:40:13.351	Process Creation	C:\Windows\SysWOW64\cmd.exe	-
January 22nd 2018, 17:40:13.053	Process Creation	C:\Windows\SysWOW64\dllhost.exe	-
January 22nd 2018, 17:40:12.719	Process Creation	C:\Program Files\Internet Explorer\iexplore.exe	-
January 22nd 2018, 17:40:00.057	Process Creation	C:\Windows\System32\VSSVC.exe	-
January 22nd 2018, 17:39:19.424	Process Creation	C:\Users\user1\Downloads\locky\L1_c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f.exe	54

Figure 14. Query to Analyze Process Creation and Termination Events (Sample L1)

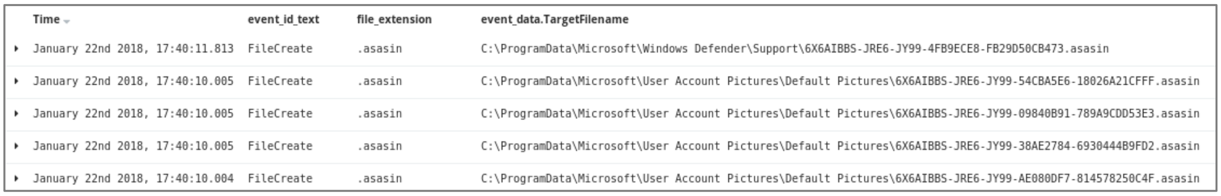
A query was created to search for file creations, which was relevant to ransomware because ransomware creates new encrypted files in order to demand a ransom payment. The query highlighted the timestamp, the created filename, and the process name that created the file. The results of this query after executing ransomware sample C1 on the endpoint system are introduced in Figure 15. Only three file creation events were triggered by sample C1. All of those were related to the notification files created by the ransomware. This query identified one additional file created by the ransomware that was not observed manually when the sample was first executed during the dataset selection process discussed on page 20. The newly discovered file was "`@__README__@.url`". Another item of interest was the fact that none of the encrypted files with file extension "`.cerber3`" were listed in this query. The ransomware must have created the encrypted files in a manner that was not logged by the Sysmon file creation event, but the reason Sysmon did not trigger this activity could not be confirmed by this design science study.



Time	event_id_text	event_data.TargetFilename	event_data.Image
January 6th 2018, 19:04:37.793	FileCreate	C:\Users\user1\Downloads\@___README___@.url	C:\Users\user1\Downloads\cerber\1_6563059c4e556e2bc1589b9711a328c4499baed3b0a14b533a467ce65ee37af6.exe
January 6th 2018, 19:04:37.792	FileCreate	C:\Users\user1\Downloads\@___README___@.html	C:\Users\user1\Downloads\cerber\1_6563059c4e556e2bc1589b9711a328c4499baed3b0a14b533a467ce65ee37af6.exe
January 6th 2018, 19:04:37.792	FileCreate	C:\Users\user1\Downloads\@___README___@.txt	C:\Users\user1\Downloads\cerber\1_6563059c4e556e2bc1589b9711a328c4499baed3b0a14b533a467ce65ee37af6.exe

Figure 15. Query to Analyze File Creations (Sample C1)

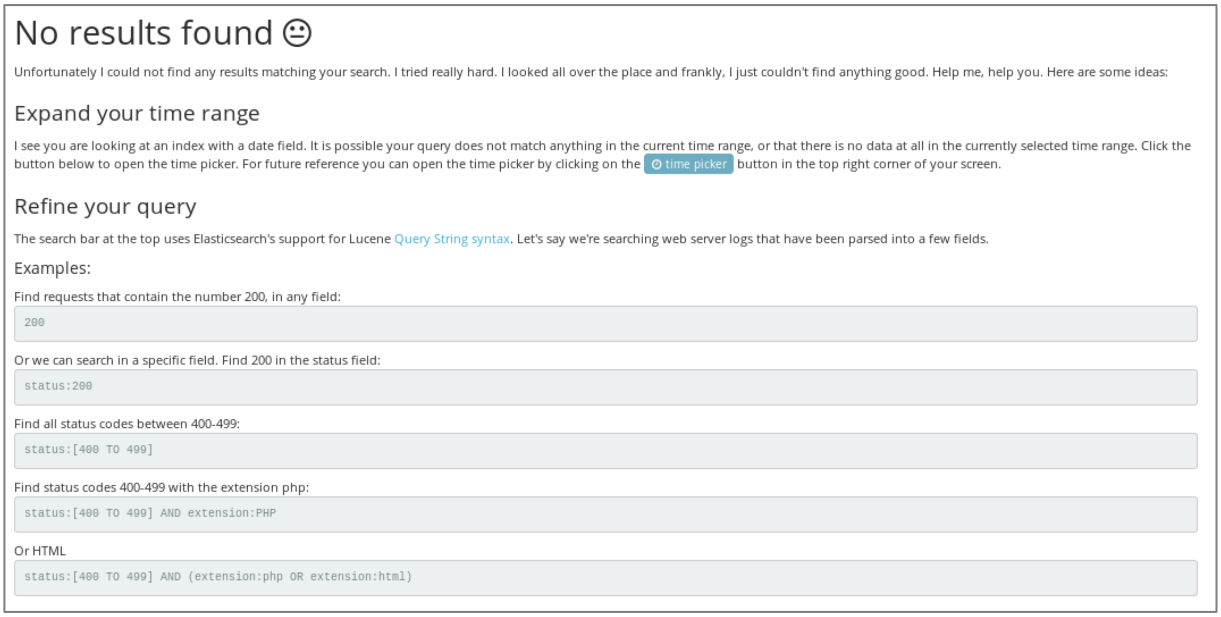
A query was created to search for common ransomware file extensions. The query displayed the timestamp, the file extension, and the filename. The results of this query after executing sample L1 are introduced in Figure 16. The creation of encrypted "`.asasin`" files by sample L1 triggered the Sysmon file creation event and was captured in the query results.



Time	event_id_text	file_extension	event_data.TargetFilename
January 22nd 2018, 17:40:11.813	FileCreate	.asasin	C:\ProgramData\Microsoft\Windows Defender\Support\6X6AIBBS-JRE6-JY99-4FB9ECE8-FB29D50CB473.asasin
January 22nd 2018, 17:40:10.005	FileCreate	.asasin	C:\ProgramData\Microsoft\User Account Pictures\Default Pictures\6X6AIBBS-JRE6-JY99-54CBA5E6-18026A21CFFF.asasin
January 22nd 2018, 17:40:10.005	FileCreate	.asasin	C:\ProgramData\Microsoft\User Account Pictures\Default Pictures\6X6AIBBS-JRE6-JY99-09840B91-789A9CDD53E3.asasin
January 22nd 2018, 17:40:10.005	FileCreate	.asasin	C:\ProgramData\Microsoft\User Account Pictures\Default Pictures\6X6AIBBS-JRE6-JY99-38AE2784-6930444B9FD2.asasin
January 22nd 2018, 17:40:10.004	FileCreate	.asasin	C:\ProgramData\Microsoft\User Account Pictures\Default Pictures\6X6AIBBS-JRE6-JY99-AE080DF7-814578250C4F.asasin

Figure 16. Query to Analyze Ransomware File Extensions (Sample L1)

The results of this query after executing sample C1 are introduced in Figure 17. The creation of encrypted "`.cerber3`" files by sample C1 did not trigger the Sysmon file creation event and were not captured in the data. This matches what was discovered when reviewing the results of the query discussed in the previous paragraph.



No results found 😐

Unfortunately I could not find any results matching your search. I tried really hard. I looked all over the place and frankly, I just couldn't find anything good. Help me, help you. Here are some ideas:

Expand your time range

I see you are looking at an index with a date field. It is possible your query does not match anything in the current time range, or that there is no data at all in the currently selected time range. Click the button below to open the time picker. For future reference you can open the time picker by clicking on the time picker button in the top right corner of your screen.

Refine your query

The search bar at the top uses Elasticsearch's support for Lucene Query String syntax. Let's say we're searching web server logs that have been parsed into a few fields.

Examples:

Find requests that contain the number 200, in any field:

```
200
```

Or we can search in a specific field. Find 200 in the status field:

```
status:200
```

Find all status codes between 400-499:

```
status:[400 TO 499]
```

Find status codes 400-499 with the extension php:

```
status:[400 TO 499] AND extension:PHP
```

Or HTML

```
status:[400 TO 499] AND (extension:php OR extension:html)
```

Figure 17. Query to Analyze Ransomware File Extensions (Sample C1)

The queries discussed above demonstrate the usefulness of the created instantiation artifact in analyzing Sysmon event to investigate ransomware incidents. The created queries provide a guided starting point for organizations that wish to analyze their Sysmon logs and understand what Sysmon events are triggered by ransomware.

Artifact 3: A Tool to Trigger All Sysmon Events

This research developed a third unique artifact which was used to verify the instantiation artifact discussed above. The third artifact was a tool that triggered every Sysmon event in order to confirm that Sysmon logging was working as expected. Sysmon events are detailed in Appendix A for reference. Although technically the third artifact is also classified as a design science instantiation (March & Smith, 1995), this research refers to this artifact as a tool in order to distinguish it from the implementation instantiation described in the section "Artifact 2: An Instantiation of the Method". The trigger tool artifact verified that Sysmon logging was working correctly and validated the accuracy of the instantiation artifact. Verifying that logging is working as expected is critical for incident response, otherwise data may not be available when it is needed most: during an incident. The literature review did not identify an existing tool to trigger all the Sysmon events and confirm that the configuration file is working as expected, so this tool provides new capabilities to the community.

The tool artifact was developed on the endpoint system in the instantiation environment. The tool was written in Python version 3.6.3 using the IDLE Python GUI (Python, 2018a). Python was used because it is an interpreted language, so other researchers can verify exactly what the trigger tool is doing by reviewing the code before executing it. The tool performs actions that cause Sysmon events to trigger, but the tool itself is benign. Other researchers can confirm that the tool does not implement malicious behavior.

The Python code for the trigger tool is provided in Appendix C. The trigger tool uses a manifest file that was written during this study to trigger three of the Sysmon events. The manifest file is provided in Appendix D. Many references were cited during the development of the Python script and the manifest file. References are documented as comments within the code in Appendix C and Appendix D. The trigger tool has external dependencies that are introduced in Table 9.

Table 9. External Dependencies for Trigger Tool Artifact

Dependency	**Purpose**
Sysmon v6.20	Sysmon must be installed and running in order to log Sysmon events.
Sysmon Configuration File `sysmonconfig-modified.xml`	Sysmon must be configured to include all the events. By default, not all Sysmon events are logged. The config file `sysmonconfig-modified.xml` is located in Appendix B.
Python 3.6	The tool to trigger all Sysmon events is written in Python 3.6. The script "trigger.py" is located in Appendix C.

Process Monitor v3.40
(Russinovich, 2017b)
	Installs a driver to trigger Sysmon Event ID 6.
	Save the files below into the same folder as Trigger Tool Python script `trigger.py:`
	`./ProcessMonitor/procmon.exe`
	`./ProcessMonitor/procmon.chm`
	`./ProcessMonitor/eula.txt`

Microsoft PowerShell v2.0	PowerShell 2.0 is included by default in Microsoft Windows 7 (Barrato, 2009).

Python v3.6 Built-in Modules	The following Python modules are built-in Python 3.6:
	`ctypes`
	`inspect`
	`os`
	`os.path`
	`shutil`
	`struct`
	`subprocess`
	`sys`
	`tempfile`
	`threading`
	`time`
	`winreg`

Manifest file `trigger.mof`	Required to trigger Sysmon Events IDs 19, 20, and 21. File is located in Appendix D Save `trigger.mof` file into the same folder as the Trigger Tool Python script "trigger.py" .

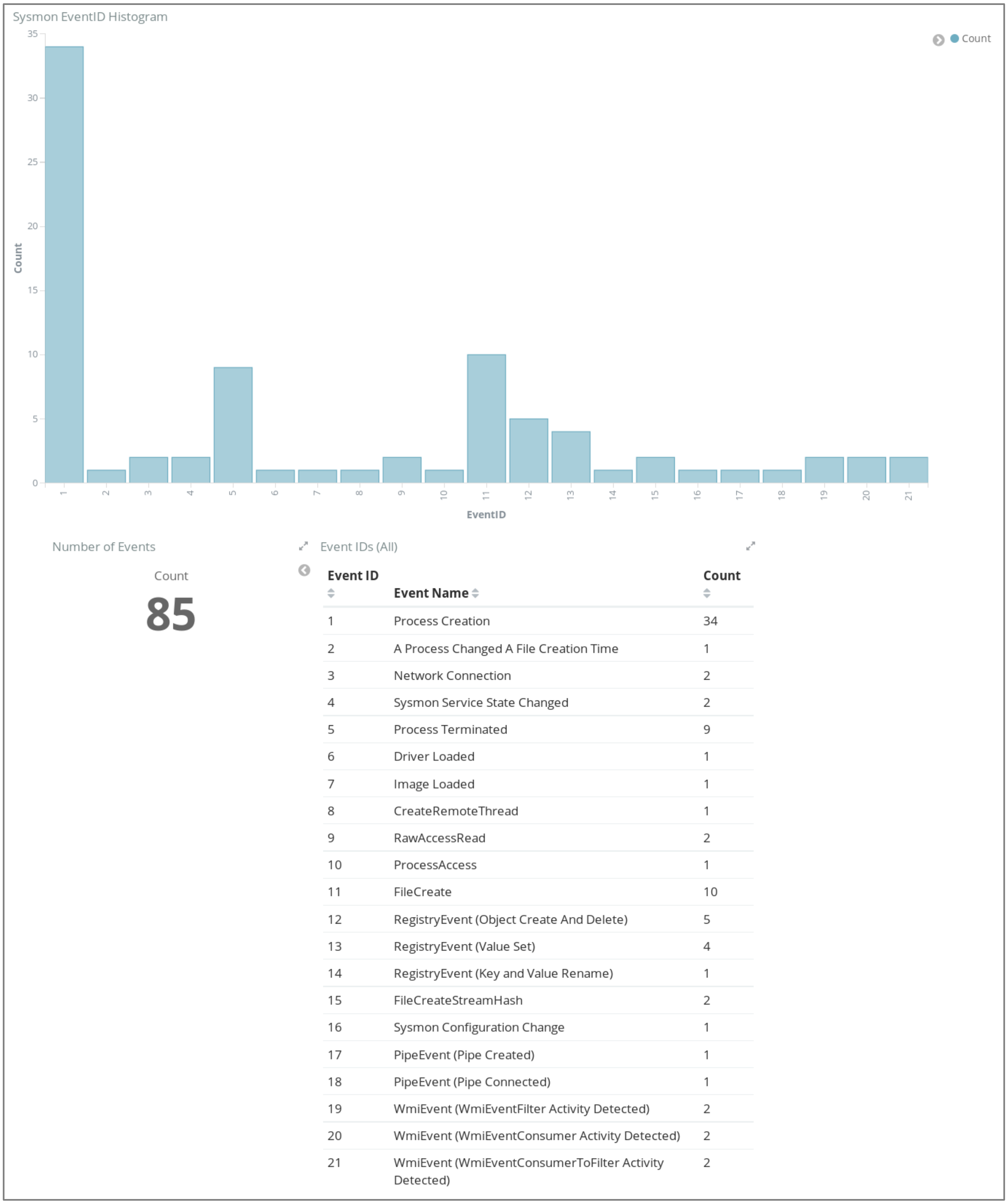
The trigger tool needs to be executed with Administrative privileges. Barring any unexpected errors, the tool cleans up after itself by removing any files or registry entries that it creates. The trigger tool provides textual output to indicate its progress during execution. The output of the trigger tool is introduced in Figure 18. The keen observer will note that the events are not triggered in numerical order. During development, it was observed that the manner in which the tool triggers certain events affected other events.

```
          VERIFY YOUR FILE LOCATIONS MATCH TO ENSURE TRIGGERS WORK:
              Sysmon Path:  C:\progra~1\Sysmon-v6.20\sysmon.exe
        Sysmon Config File:  C:\Users\user1\Desktop\code\sysmonconfig-
                                 modified.xml
           Manifest File:  C:\Users\user1\Desktop\code\trigger.mof

                              [ START ]
                         EventID 9: Triggered
                         EventID 1: Triggered
                         EventID 2: Triggered
                         EventID 3: Triggered
                         EventID 4: Triggered
                         EventID 5: Triggered
                         EventID 6: Triggered
                        EventID 10: Triggered
                        EventID 11: Triggered
                        EventID 12: Triggered
                        EventID 13: Triggered
                        EventID 14: Triggered
                        EventID 15: Triggered
                        EventID 16: Triggered
                        EventID 18: Triggered
                        EventID 17: Triggered
                        EventID 21: Triggered
                        EventID 19: Triggered
                        EventID 20: Triggered
                         EventID 7: Triggered
                         EventID 8: Triggered
                              [ DONE ]
```

Figure 18. Output of the Tool Artifact

Specifically, event IDs 7 and 8 were required to be executed last. These events indicate that a Dynamic Link Library (DLL) or thread was created in another process, which is a task that malware commonly performs to infect a system. The trigger tool does not attempt to be successful at injecting another process; it only needs to trigger the Sysmon event. Therefore, when those two Sysmon events are triggered, the system is left in an unexpected state. Although the tool does not require a reboot during execution, it is recommended to reboot the system after executing the trigger tool due to the issues noted with the process injection events.

A dashboard was created to visualize all of the Sysmon events triggered by the tool. The dashboard was used to verify that every event was triggered at least once. The dashboard is introduced in Figure 19. The dashboard contained a histogram and a numerical table listing of the events triggered. After executing the trigger tool, a total of 85 Sysmon events were generated in the instantiation environment using the Sysmon configuration file specified in Appendix B. Many events were triggered multiple times. The process creation Sysmon event was triggered most frequently, due to the fact that the Python `subprocess` module was used frequently within the tool to spawn additional processes.



Number of Events

Count

85

Event IDs (All)

Event ID	Event Name	Count
1	Process Creation	34
2	A Process Changed A File Creation Time	1
3	Network Connection	2
4	Sysmon Service State Changed	2
5	Process Terminated	9
6	Driver Loaded	1
7	Image Loaded	1
8	CreateRemoteThread	1
9	RawAccessRead	2
10	ProcessAccess	1
11	FileCreate	10
12	RegistryEvent (Object Create And Delete)	5
13	RegistryEvent (Value Set)	4
14	RegistryEvent (Key and Value Rename)	1
15	FileCreateStreamHash	2
16	Sysmon Configuration Change	1
17	PipeEvent (Pipe Created)	1
18	PipeEvent (Pipe Connected)	1
19	WmiEvent (WmiEventFilter Activity Detected)	2
20	WmiEvent (WmiEventConsumer Activity Detected)	2
21	WmiEvent (WmiEventConsumerToFilter Activity Detected)	2

Figure 19. Dashboard of Sysmon Events Generated by the Tool Artifact

Ransomware Profiling

This research focused on ransomware because of its increasing relevance as a threat to organizations as discussed in Chapter 2. All of the artifacts created above were developed in order to enable the profiling of ransomware based on the Sysmon events triggered by the ransomware. The ransomware dataset discussed on page 20 was used to demonstrate the effectiveness of this profiling technique. Each of the nine ransomware samples were executed within a virtual environment using snapshots to ensure that each sample was executed in the same manner. One ransomware sample was executed on the endpoint system, and the resulting Sysmon events were recorded in the Elastic Stack within the instantiation. After confirming that the ransomware executed as expected, the snapshot was reverted to its starting point and the next sample was executed. This cycle was repeated until all ransomware samples had been executed within the instantiation.

After all nine ransomware samples were executed, analysis of the Sysmon events was performed using the instantiation created by this research. A custom dashboard was used to explore each of the Sysmon events triggered during the execution of the samples. The start time and end time for each sample was determined based on the Sysmon events for process creation (event ID 1) and process termination (event ID 5). This provided a timeframe for when each ransomware sample was active on the system. The dashboard focused on that timeframe and displayed a histogram of all the Sysmon events each ransomware sample triggered. This histogram was used to profile the sample and determine which Sysmon events each ransomware sample triggered most frequently. The detailed histograms for each sample are found in Appendix S. The histograms provide insight into the actions each ransomware sample executed on the infected system. Knowing what a piece of malware did on an infected system helps incident responders remove the infection and identify other infected systems.

By analyzing the number of events triggered by the ransomware dataset, it was determined that only seven of the twenty-one Sysmon events were triggered by the ransomware samples. A list of the Sysmon events triggered by the ransomware dataset is introduced in Table 10.

Table 10. List of Sysmon Events Triggered by the Ransomware Dataset

Sysmon Event	Description	# of Events	Samples That Triggered Event
3	TCP/UDP network connection	2217	C1, C2, C3
11	File created	975	All
2	File creation time was changed by a process	523	C1, C2, L1, L2, L3
1	Newly created process	68	All
5	Process terminated	9	All
13	Registry value was set	6	G1, G2, G3, L1, L2, L3
9	Process opened disk or volume for read access	2	C1, C2
4	Sysmon service started or stopped	0	None
6	Driver loaded	0	None
7	Module/DLL loaded into a process	0	None
8	Thread created in another process	0	None
10	Process opened another process	0	None
12	Registry object created or deleted	0	None
14	Registry key and/or value was renamed	0	None
15	File stream created	0	None
16	Sysmon configuration changed	0	None
17	Named pipe created	0	None
18	Named pipe connected	0	None
19	Windows Management Instrumentation (WMI) Event Filter registered	0	None
20	WMI Event Consumer registered	0	None
21	WMI Event Consumer bound to WMI Event Filter	0	None

Sysmon event ID 3, which indicates an outbound network connection, was triggered the most number of times. However, only the Cerber ransomware samples triggered that event. None of the GlobeImposter or Locky samples initiated any network connections. Therefore, the Cerber samples utilized network capabilities, while GlobeImposter and Locky did not.

According to the results gathered in Table 10, the second most frequent Sysmon event triggered by the dataset was event ID 11, which is triggered when a file is created. Since ransomware encrypts existing files and saves them with a new file extension, it makes sense that this Sysmon event would be triggered frequently by ransomware.

The third most frequent Sysmon event triggered by the dataset was event ID 2, which is triggered when a process changes the creation time of a file. Malware sometimes changes file creation times to disguise the time the system was infected and to make altered files blend in with existing files on the system. All Locky samples triggered this event. Only two Cerber samples triggered this event. None of the GlobeImposter samples triggered this event.

The fourth most frequent Sysmon event triggered by the dataset was event ID 1, which is triggered when a process is created. Malware will commonly launch additional processes to conduct the desired malicious behavior. Sometimes these will be scheduled tasks; other times it will be executing system utilities, such as ping, to check what capabilities are on the system. All of the ransomware samples triggered this event. On average, the event was triggered 7.5 times by each sample in the dataset. This indicates that on average, the samples in the selected ransomware dataset launched approximately seven additional processes.

The fifth most frequent Sysmon event triggered by the dataset was event ID 5, which is triggered when a process is terminated. All samples triggered this event one time. In all cases, this event was triggered when the ransomware process itself terminated. Sometimes malware will launch other processes and terminate the originating process as soon as the other processes are launched. However, that was not the case with any of the samples in the selected dataset. In all cases in this study, the ransomware samples in the dataset first executed the desired behavior of encrypting files for ransom and notifying the user and then terminated the parent process.

The sixth most frequent Sysmon event triggered by the dataset was event ID 13, which is triggered when a registry value is set. All of the GlobeImposter and Locky samples triggered this event exactly one time. None of the Cerber samples triggered this event. Sometimes malware uses registry values to store data or to indicate that the system has been infected.

The seventh most frequent Sysmon event triggered by the dataset was event ID 9, which is triggered when a process opens a disk for read access. Only the Cerber samples C1 and C2 triggered this event, and those samples triggered the event only one time. The other samples certainly read files on the filesystem in order to encrypt the files for ransom. The other samples may have used a different method to gain access to the filesystem in a manner that was not observable by Sysmon.

According to the results generated by this study, the following conclusions could be made regarding the ransomware samples in the dataset. None of the remaining Sysmon events were triggered by any of the samples. None of the samples interacted with the Sysmon service to start, stop, or change the Sysmon service. None of the samples loaded a driver. None of the samples injected a Dynamic Link Library (DLL) or thread into another process or requested access to another running process. None of the samples created or deleted registry objects, and none of them renamed a registry key or value. None of the samples created a file stream. None of the samples created or connected to a named pipe. Finally, none of the ransomware samples in the selected dataset used Windows Management Instrumentation (WMI) filters or consumers.

The results observed when profiling ransomware are understandable considering that the task of ransomware is to encrypt files and to notify the user to pay a ransom to get the files back. An interesting finding was that only the Cerber samples initiated outbound network connections. Another interesting finding was that none of the Cerber samples set a registry value. By profiling the ransomware in this manner, insight was gained into which Sysmon events are triggered most frequently by ransomware. This can aid organizations to help focus analysis efforts when investigating ransomware incidents in their environment. The insight also helps understand what actions each ransomware sample took on the infected system, which helps an incident responder remove the infection and identify other infected systems. The result presented in this chapter are summarized in Chapter 5, where the key contributions of this research study are conveyed as well.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

This research provides tangible results to organizations on how to use Sysmon to log Microsoft Windows events to investigate ransomware infections. This chapter discusses the unique contributions that this research study made to the incident response community. Recommendations are made to guide organizations desiring to use this research in their environment. Limitations of the research application are articulated. Finally, future research ideas are presented to encourage other researchers to further explore this research topic and expand on the ideas and results presented by this study.

Contributions

This research contributes knowledge that can be applied by organizations looking to effectively implement endpoint logging on Microsoft Windows systems using Sysmon. Three artifacts were created to explore the research question and to develop capabilities within the problem space: a method, an instantiation, and a tool. By following the method designed by this research, organizations can setup Sysmon and look for ransomware in Sysmon logs. The method provides guidance that an organization can follow to implement their Sysmon logging and analysis. Organizations that have resources available can follow the method to implement their own Sysmon testing infrastructure. Organizations that are not capable of designing their own infrastructure can leverage the results of the instantiation created during this research. The instantiation artifact provides a concrete example of an environment that can be created to analyze Sysmon logs from a central location using an Elastic Stack infrastructure. The Python tool artifact enables an organization to verify that their Sysmon deployment is logging events as expected and desired.

This research contributes a Sysmon configuration file, based on SwiftOnSecurity's configuration (2017), that can be used in conjunction with the trigger tool artifact to verify that Sysmon logging is working as expected. The Sysmon configuration file is available in Appendix B. This research provides the code for the Python tool that triggers all the Sysmon events. The code is available in Appendix C and Appendix D. External dependencies for the tool are documented in Table 9. All Elastic Stack software configuration details are shared so that organizations can setup the instantiation in their own environment to replicate or expand on the results of this research. These configuration files are found in Appendix E, Appendix J, Appendix K, and Appendix L.

A key contribution of this research is the Logstash pipeline that parses the Sysmon events received from Winlogbeat and stores them in Elasticsearch. The Logstash pipeline developed in this research enriches the data to provide additional context and information to enable further analytics. The Logstash pipeline is found in Appendix F. The Logstash dictionary files created in this study to augment parsed data and provide data enrichment of the Sysmon logs are found in Appendix G, Appendix H, and Appendix I.

Another contribution is the development of email alerts that are triggered when certain activity is seen in the logs. These alerts were made functional through the Elastic X-Pack Watcher plugin. A trial license for X-Pack was used in this research. Four Watcher alerts were created. One alert is triggered whenever a process is seen that has been labeled as malicious by VirusTotal. Another alert is triggered whenever common ransomware file extensions are created on the system. A third alert is triggered whenever a process modifies the creation time of a file. The final alert is triggered whenever more than 240 files are created within a one-minute threshold. These alerts bring attention to activity which may be suspicious, which helps incident responders identify malicious activity more quickly. The code for all of the Watcher alerts can be found in Appendix N, Appendix O, Appendix P, and Appendix Q.

All of the development and artifacts created above were used to profile ransomware based on the Sysmon events triggered by the ransomware. The profiling provides empirical data regarding which Sysmon events are triggered most frequently by the ransomware dataset. This research provides insight into which Sysmon events are most useful when investigating ransomware. Details regarding the Sysmon events frequency can be found in Chapter 4. Seven Sysmon events were found to be triggered by the ransomware dataset. The events that captured network connections, file creations, file creation time changes, and newly created processes were triggered most frequently by the dataset. The dataset also triggered the events that indicate a process was terminated, a registry value was set, and a process opened a disk for read access. The ransomware dataset provided proof that the artifacts developed in this study could be used to investigate ransomware with Sysmon logs.

Limitations

This research has limitations that must be understood in order to apply the results appropriately. The first limitation is that the research is based primarily on existing, freely downloadable software. No corporate or commercially available solutions were explored to address the research problem. If expensive commercial software products were selected, then many organizations may not be able to use the results due to the expense. Because commercial products were not explored, there may exist commercial solutions to answer the research question. If an organization has a large budget, they are encouraged to explore the capabilities of commercial solutions in addition to this solution to decide what best fits their needs.

This research is not intended to replace antivirus products. Antivirus solutions are essential to preventing the execution of malware on an endpoint system. This research is intended to work alongside antivirus solutions to log activity and provide useful data during an incident response.

This research is limited by the versions of the software available at the time the artifacts were developed. This research is based on Sysmon version 6.20 and Elastic version 5.6.5. During the development of this study, Sysmon and Elastic were upgraded to newer versions. As of January 2018, Sysmon 7.01 and Elastic 6.1 were the current released versions. SwiftOnSecurity updated their Sysmon configuration file to support Sysmon version 7.01 on January 17, 2018. This research was built upon the July 13, 2017 version of the Sysmon configuration file provided by SwiftOnSecurity (2017). SwiftOnSecurity states that Sysmon 7.01 contains important filtering bug fixes (2018), but it does not elaborate on what those bugs were. If any of the software versions used in this research have errors or underlying bugs, then this implementation will also contain those errors. This implementation does not seek to add any additional functionality into the Sysmon or Elastic software. This limitation did not affect the outcomes of this research because Sysmon and Elastic are already very robust, full functioning programs and no critical bugs were identified that affected the results of this study. However, if additional functionality is desired in this research, one must ensure that Sysmon and Elastic products can support the desired functionality.

This research was limited by restrictions placed on free user accounts by VirusTotal and Hybrid Analysis. These restrictions affect the application programming interface (API) capabilities that can be used to automate data retrieval to enrich data results. Specifically, VirusTotal restricted the API to 4 queries per minute (VirusTotal, 2018). The public VirusTotal API key must not be used in commercial products or services (VirusTotal, 2018). Hybrid Analysis limited the display of certain analysis results (2017m). This research study was able to work around those limitations without affecting results. However, the limitations should be heeded if applying this solution in non-research environment to ensure that services are being used according to license agreements.

Ransomware can be delivered using a variety of infection vectors. This research did not explore the infection vector or manner in which ransomware gets executed. This research only analyzed the resulting ransomware infection, commonly called the ransomware payload (Cimpanu, 2017).

This research is limited by the shortcomings of the design science methodology, as discussed in Chapter 3. Although design science was the correct methodology to use to develop useful artifacts, this research cannot make conclusions regarding the performance or efficacy of Sysmon, Elastic, or any implemented tools. This was not a quantitative or qualitative study. Results relating to the selected ransomware dataset are provided as a proof of concept that the developed artifacts were useful in addressing the research question. Numerical results should be interpreted with the understanding that a small sample size was used. Additional quantitative and qualitative research could be performed on the results of this study to gain additional understanding of how and why Sysmon performs as it does if more concrete numerical results are desired.

Recommendations

In the process of implementing this research and developing artifacts, this research accentuated several recommendations useful to organizations regarding incident response. In light of the growing threat of malware and ransomware, it is critical that organizations get logging in place so that historical evidence exists when an incident occurs. Historical logging provides great insight into what occurred on a system and can lead to quicker understanding of the incident and more complete recovery. Specifically, this research implements an Elastic Stack solution to leverage Sysmon logging and analysis.

Organizations need to be able to detect malware as soon as possible. It is recommended that organizations implement cyber security defense in depth to provide a layered approach to protect against malware. Current antivirus products that are regularly patched are an organization's first line of defense against known malware. However, malware changes frequently and antivirus products may miss new malware variants. Therefore, logging is essential to ensure that data exists to enable incident response after an incident occurs. Specifically, this research recommends implementing automation to detect and report suspicious activity without manual intervention.

Organizations should also be looking at their logs manually so that abnormal activity stands out. Manual review of log data also serves to verify that logging is occurring as expected, so that data is available when an incident occurs. This research demonstrates how to use an Elastic Stack implementation to analyze Sysmon logs. The trigger tool artifact developed by this study can be used to verify Sysmon logging. Dashboards and queries provide an easy way to visualize logs and provide quicker understanding. This research recommends using Sysmon to log Microsoft Windows events and analyzing the logs with an Elastic Stack to stay proactive against malware threats.

Future Research

This research only touches the surface regarding capabilities that can be developed and explored when analyzing Sysmon logs. The Appendices contain source code and configuration files to help other researchers leverage the progress made by this study. Researchers are encouraged to implement their own instantiation to collect and analyze Sysmon logs to further enhance this research. This section presents several avenues in which this research could be extended.

This research could be extended to study additional malware samples. Additional GlobeImposter, Cerber, and Locky samples could be downloaded and executed within the virtual environment to increase the sample size of the ransomware dataset. Samples from other ransomware families, such as Jaff, Sage, and Mamba could be identified and profiled with this implementation. Other types of malware could be used to see what Sysmon events are triggered by malware that is not ransomware.

Capabilities could be added to this research to improve analytics and automation. Additional dashboards could be created in Kibana to highlight key data. Additional queries could be identified to point out activity of interest to an incident responder. Additional Watcher alerts could be created to automatically alert when specific situations are identified in the data. Creating these capabilities and making them available to the incident response community would provide additional value by reducing the development time required to create new analytics from scratch.

This research only analyzes the Sysmon events that are triggered by ransomware. It does not perform static or dynamic analysis on the ransomware samples to determine if the samples perform additional activities that Sysmon does not currently monitor. Future work could be performed to determine if there is additional suspicious system activity that Sysmon should detect in future versions to improve the effectiveness of Sysmon logging. The authors of Sysmon could be contacted to request additional capabilities be built into the tool.

This research used VirusTotal and Hybrid Security public API keys to enrich the data logging with additional information about processes and malware. Many data sources exist to provide context to Internet Protocol (IP) addresses, domains, malware, and threats. Future work could expand the number of data sources and the type of data enriched in the logs through the Logstash pipeline.

The trigger tool developed by this research could be improved to be more reliable and more efficient. Error handling was implemented, but additional testing should be performed to make the trigger tool script more robust. PowerShell is leveraged extensively within the Python script using the Python `subprocess` module. The trigger tool could be rewritten entirely in PowerShell instead of Python to improve efficiency. In addition, the trigger tool uses the Process Monitor software to trigger the driver installation Sysmon event. To reduce external dependencies and to reduce the footprint of the trigger tool, a custom driver could be implemented and loaded, instead of using the Process Monitor driver. Further work could also be performed to explore better ways to trigger Sysmon event IDs 7 and 8 so that a reboot is not required after execution of the tool artifact.

This research focused on a virtual environment consisting of one endpoint system and one server. Additional work could be performed to increase the scale of the implementation to include more endpoints and more servers. The Elastic Stack is frequently used on large clusters. Guidance on how this implementation performs a large networked environment would help expand use case scenarios where this solution could be deployed.

Research could be performed to investigate how to integrate other open source tools, such as Suricata, Yara, and Snort, into the implementation configured by this research (Cisco, 2018; Suricata, 2018; Yara, 2018). If malicious network connections are identified, perhaps Suricata could be updated to block the traffic. If executable files are identified on the system, perhaps Yara signatures could be run across the executables to look for malicious signatures. Perhaps Snort logs could be integrated into the system to provide additional network detection context.

This research could also be updated to evaluate a different endpoint detection tool besides Sysmon. Many logging tools exist, and the Elastic Stack solution can ingest a variety of logs. Additional Windows event logs could be added to the implementation. Other logs, such as web traffic logs or antivirus logs, could be included in the solution to provide additional context regarding ongoing activity within an organization.

Summary

This research explored the question “What methodology can an organization follow to determine which Sysmon events should be analyzed to identify ransomware in a Windows environment?” A design science research study was conducted to develop three useful artifacts to answer the research question. A method was developed to articulate steps to take to use Sysmon logging to investigate ransomware infections. The developed method provides guidance to help organizations properly leverage Sysmon logs during incident response. An instantiation was implemented to provide a comprehensive environment to apply the method and demonstrate a proof of concept. The implemented instantiation provides a realistic example that organizations can implement to start analyzing Sysmon logs quickly. A tool was created to trigger all of the Sysmon events in order to verify that logging was behaving as expected in the instantiation. The tool enables organizations to confirm that Sysmon is logging as expected to ensure that data is available when needed during an incident response.

The results of this research showed that Sysmon logging provides useful historical knowledge when investigating a ransomware infection. Using the artifacts developed in this study, ransomware activity can be investigated more quickly during incident response. A dataset was compiled consisting of publicly available samples of three major ransomware threats, GlobeImposter, Cerber, and Locky. These ransomware families were identified by Symantec as major threats to organizations in 2017, making this research extremely relevant and applicable to organizations today. The nine dataset samples were executed on the endpoint system and analyzed using the instantiation artifact. The ransomware samples were profiled using the developed artifacts to determine which Sysmon events were triggered most frequently by the ransomware. By knowing what Sysmon events are logged by ransomware, an incident responder can inspect Sysmon logs more efficiently to determine what actions took place on infected systems.

This study adds to the overall body of knowledge pertaining to incident response, particularly in the area of using Sysmon logs to investigate ransomware infections. The contributions of this research fill a gap identified in existing research regarding a lack of guidance in how to use Sysmon to investigate ransomware infections, as well as lack of a tool to verify Sysmon logging. The artifacts developed by this study fill those gaps and provide detailed source code and configuration files to enable organizations to implement, verify, and enhance the research results. Organizations can use the results of this research to improve their ability to analyze Sysmon logs. Researchers are encouraged to build upon the research created in this study to further explore and expand the use of Sysmon logging during incident response.

REFERENCES

Abrams, L. (2017, May 11). Jaff Ransomware Distributed via Necurs MALSPAM and asking for a $3,700 Ransom. *Bleeping Computer.* Retrieved from https://www.bleepingcomputer.com/news/security/jaff-ransomware-distributed-via-necurs-malspam-and-asking-for-a-3-700-ransom/.

Adm. (2015, April 23). Using NXLog with Elasticsearch and Kibana. *NXLog Ltd*. Retrieved from https://nxlog.co/using-nxlog-elasticsearch-and-kibana.

Andy. (2017, August 7). Sysmon and Neo4j. *MalwareSoup*. Retrieved from https://www.malwaresoup.com/sysmon-and-neo4j/.

Arul, M. (2017, January 20). How to Install Elastic Stack on CentOS 7. *HowtoForge.* Retrieved from https://www.howtoforge.com/tutorial/how-to-install-elastic-stack-on-centos-7.

Bandos, T. (2016, October 13). Seek Evil, and Ye Shall Find: A Guide to Cyber Threat Hunting Operations. *Digital Guardian*. Retrieved from https://digitalguardian.com/blog/seek-evil-and-ye-shall-find-guide-cyber-threat-hunting-operations.

Banon, S. (2010, February 08). You Know, for Search. *Elastic [web log]*. Retrieved from https://www.elastic.co/blog/you-know-for-search.

Barreto, J. (2009, November 25). Download for Powershell v2 for Windows 7? No need… It's already there! *Microsoft TechNet.* Retrieved from https://blogs.technet.microsoft.com/josebda/2009/11/25/download-for-powershell-v2-for-windows-7-no-need-its-already-there/.

Bryner, J. (2015, February 24). Mozilla: Tackling Security Logs with the ELK Stack. *Elastic*. Retrieved from https://www.elastic.co/elasticon/2015/sf/tackling-security-logs-with-the-elk-stack.

Check Point. (2017a). Ransomware Defense Survey. *Check Point Software Technologies, Ltd.* Retrieved from https://www.checkpoint.com/downloads/resources/ransomware-survey.pdf.

Check Point. (2017b, February). 2016 H2 Global Threat Intelligence Trends. Retrieved from http://blog.checkpoint.com/wp-content/uploads/2017/02/H2_SummaryGlobal_Report_170210_A.cleaned.pdf.

Churchill, M. (2015, February 24). Parsing Sysmon Events for IR Indicators. *Crowdstrike Blog*. Retrieved from https://www.crowdstrike.com/blog/sysmon-2/.

Chuvakin, A. (2014, July 26). Named: Endpoint Threat Detection & Response. *Gartner Blog Network.* http://blogs.gartner.com/anton-chuvakin/2013/07/26/named-endpoint-threat-detection-response/.

Chuvakin, A. (2016, April 25). EDR tool wins – only for the enlightened? *Gartner Blog.* Retrieved from http://blogs.gartner.com/anton-chuvakin/2016/04/25/edr-tool-wins-only-for-the-enlightened/.

Cimpanu, C. (2017, August 17). Ransomware Was the Most Prevalent Malware Payload Delivered via Email in Q2 2017. *Bleeping Computer.* Retrieved from https://www.bleepingcomputer.com/news/security/ransomware-was-the-most-prevalent-malware-payload-delivered-via-email-in-q2-2017.

Cisco. (2018). Snort – Network Intrusion Detection & Prevention System. Retrieved from https://www.snort.org.

Connelly, S. & Hinman, L. (2016). Elasticsearch: Getting Started [webinar]. *Elastic*. Retrieved on December 14, 2016 from https://www.elastic.co/webinars/getting-started-elasticsearch?baymax=rtp&elektra=docs&iesrc=ctr.

Costis, A. (2017, May 17). Detecting WannaCry Activity on Sysmon-Enabled Hosts. *LogRhythm [web log]*. Retrieved from https://logrhythm.com/blog/detecting-wannacry-activity-on-sysmon-enabled-hosts/.

Creswell, J. W., & Creswell, J. D. (2018). Research design: Qualitative, quantitative, and mixed methods approaches. *SAGE Publications, Inc.* Retrieved from https://books.google.com/books?id=KGNADwAAQBAJ.

Crypsisgroup. (2016, December 14). Splunkmon. *GitHub.* Retrieved from https://github.com/crypsisgroup/Splunkmon.

CyberSponse. (2015, May 12). CyberSponse to Utilize Elasticsearch to Organize Data for Incident Response. Press Release retrieved from https://cybersponse.com/press-release/press-release-05-12-15.pdf.

Dang, Q. (2015, August 04). Secure Hash Standard. *National Institute of Standards and Technology*. FIPS PUB 180-4. Retrieved from https://www.nist.gov/publications/secure-hash-standard.

Delgado, P. (2017, March 3). Advanced Sysmon Filtering Using Logstash. *Syspanda.* Retrieved from https://www.syspanda.com/index.php/2017/03/03/sysmon-filtering-using-logstash/.

Dreyfuss, J. (2015, April 8). Log Management Tools Face-Off: Splunk vs. Logstash vs. Sumo Logic. *OverOps Blog.* Retrieved from https://blog.takipi.com/log-management-tools-face-off-splunk-vs-logstash-vs-sumo-logic/.

Dubey, S. (2016, October 20). Unfolding the Mystery of Cerber Ransomware's Random File Extension. *McAfee Labs.* Retrieved from https://securingtomorrow.mcafee.com/mcafee-labs/unfolding-the-mystery-of-cerber-ransomwares-random-file-extension/.

Ducklin, P. (2016, February 17). "Locky" ransomware – what you need to know. *Sophos.* Retrieved from https://nakedsecurity.sophos.com/2016/02/17/locky-ransomware-what-you-need-to-know.

Elastic. (n.d.) Logstash Introduction. *Logstash Reference Guide version 5.2*. Retrieved from https://www.elastic.co/guide/en/logstash/current/introduction.html.

Elastic. (2017, March 7). Elastic Reaches 100 Million Downloads for the Elastic Stack. *Elastic Press Release.* Retrieved from https://www.elastic.co/about/press/elastic-reaches-100-million-downloads-for-the-elastic-stack.

Elastic. (2018). Overview Elastic Stack 5.6. *Elastic.* Retrieved from https://www.elastic.co/guide/en/elastic-stack/5.6/elastic-stack.html.

F-Secure. (2018). Crypto-ransomware. *F-Secure*. Retrieved from https://www.f-secure.com/en/web/labs_global/crypto-ransomware.

FBI. (2016, April 29). Incidents of Ransomware on the Rise. Retrieved from https://www.fbi.gov/news/stories/incidents-of-ransomware-on-the-rise.

Gilbert, H., & Handschuh, H. (2003, August). Security analysis of SHA-256 and sisters. In *International workshop on selected areas in cryptography* (pp. 175-193). Springer, Berlin, Heidelberg. Retrieved from https://link.springer.com/content/pdf/10.1007/978-3-540-24654-1_13.pdf.

Grance, T., Kent, K., & Kim, B. (2004). Computer security incident handling guide. *NIST Special Publication 800-61*. Retrieved from http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-61.pdf.

Haag, M. (2017a, March 13). Sysmon-dfir. *GitHub.* Retrieved from https://github.com/MHaggis/sysmon-dfir.

Haag, M. (2017b, April 7). Sysmon App for Splunk. *Splunk Enterprises.* Retrieved from https://splunkbase.splunk.com/app/3544/.

Hall, A. (2016, November 11). TA-Microsoft-Sysmon v5.0.0. *GitHub*. Retrieved from https://github.com/splunk/TA-microsoft-sysmon.

Hasherezade. (2016, August 16). Cerber Ransomware – New, But Mature. *Malwarebytes Labs.* Retrieved from https://blog.malwarebytes.com/threat-analysis/2016/03/cerber-ransomware-new-but-mature.

Hayes, B. (2016, October 5). Detecting Ransomware Attacks with Splunk. *Splunk Blog.* Retrieved from https://www.splunk.com/blog/2016/10/05/detecting-ransomware-attacks-with-splunk.html.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75-105.

Higgins, K. (2015, October 21). *The rebirth of endpoint security.* Retrieved from http://www.darkreading.com/endpoint/the-rebirth-of-endpoint-security/d/d-id/1322775.

Hybrid Analysis (2017a, July 10). Online File Analysis Results for ‘cerber.ex’. *Hybrid Analysis.* Retrieved from https://www.hybrid-analysis.com/sample/e67834d1e8b38ec5864cfa101b140aeaba8f1900a6e269e6a94c90fcbfe56678?environmentId=100.

Hybrid Analysis (2017b, September 13). Online File Analysis Results for ‘5a742870-sample.exe’. *Hybrid Analysis.* Retrieved from https://www.hybrid-analysis.com/sample/edf67ba035e52cd903017a24271544caba57dace039be51b1e867fdfd5252744?environmentId=100.

Hybrid Analysis (2017c, September 20). Online File Analysis Results for ‘D6500E31A2D9FB8CF4866E418D4EA3B6.1192D001’. *Hybrid Analysis.* Retrieved from https://www.hybrid-analysis.com/sample/b2282de3df95c6a9d0151ad61d2ab4e99400ca3104ce9003a0b13290260a7a55?environmentId=100.

Hybrid Analysis (2017d, October 24). Online File Analysis Results for ‘5l46zw33.exe’. *Hybrid Analysis.* Retrieved from https://www.hybrid-analysis.com/sample/4c054127056fb400acbab7825aa2754942121e6c49b0f82ae20e65422abdee4f?environmentId=100.

Hybrid Analysis (2017e, November 9). Online File Analysis Results for ‘e40d942c2d7e469d4e032d2d2c5592a6’. *Hybrid Analysis.* Retrieved from https://www.hybrid-analysis.com/sample/294f55a28930c8afed9b95d2af108a6916eeb2c79967e91f4dde48026bab15ce?environmentId=100.

Hybrid Analysis (2017f, November 10). Online File Analysis Results for ‘mirbrat.exe.1’. *Hybrid Analysis.* Retrieved from https://www.hybrid-analysis.com/sample/7d49a2a9d788fc8dbaa6331c8b740f689e20600ff7e8d3692b1a9c6d37a37bd6?environmentId=100.

Hybrid Analysis (2017g, November 15). Online File Analysis Results for ‘mirbrat.exe.1’. *Hybrid Analysis.* Retrieved from https://www.hybrid-analysis.com/sample/7d49a2a9d788fc8dbaa6331c8b740f689e20600ff7e8d3692b1a9c6d37a37bd6?environmentId=100.

Hybrid Analysis (2017h, November 23). Online File Analysis Results for ‘403577074344d4832649881daf8885fed4d9afc3e7a4b02247ceb9b51d858794’. *Hybrid Analysis.* Retrieved from https://www.hybrid-analysis.com/sample/403577074344d4832649881daf8885fed4d9afc3e7a4b02247ceb9b51d858794?environmentId=100.

Hybrid Analysis (2017i, November 23). Online File Analysis Results for ‘6563059c4e556e2bc1589b9711a328c4499baed3b0a14b533a467ce65ee37af6’. *Hybrid Analysis.* Retrieved from https://www.hybrid-analysis.com/sample/6563059c4e556e2bc1589b9711a328c4499baed3b0a14b533a467ce65ee37af6?environmentId=100.

Hybrid Analysis (2017j, December 4). Online File Analysis Results for ‘2aec34f32b7e1881a2a9b97496dbb58487fc088fc108775db9a138594b90e123’. *Hybrid Analysis.* https://www.hybrid-analysis.com/sample/2aec34f32b7e1881a2a9b97496dbb58487fc088fc108775db9a138594b90e123?environmentId=100.

Hybrid Analysis (2017k, December 4). Online File Analysis Results for ‘winserv.exe’. *Hybrid Analysis.* Retrieved from https://www.hybrid-analysis.com/sample/9bb05753775d3899470b705a33995b3e63121c4810e088c97dc7a1f4c606ae42?environmentId=100.

Hybrid Analysis (2017l, December 16). Online File Analysis Results for ‘BKAoQKtgfOM.exe’. *Hybrid Analysis.* Retrieved from https://www.hybrid-analysis.com/sample/c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f?environmentId=100.

Hybrid Analysis. (2017m, December 17). Falcon Sandbox Public API v1.1. *Hybrid Analysis.* Retrieved from https://www.hybrid-analysis.com/apikeys/info.

Hybrid Analysis. (2018). Hybrid Analysis. *Hybrid Analysis.* Retrieved from https://www.hybrid-analysis.com.

Kent, K., Chevalier, S., Grance, T., & Dang, H. (August 2006). Guide to integrating forensic techniques into incident response. *NIST Special Publication 800-86*. Retrieved from http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-86.pdf.

Kent, K., & Souppaya, M. (September 2006). Guide to computer security log management. *NIST Special Publication 800-92*. Retrieved from http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf.

Koopmann, L. (2017, February 4). Explaining and adapting Tay's Sysmon configuration. *Medium Blog.* Retrieved from https://medium.com/@lennartkoopmann/explaining-and-adapting-tays-sysmon-configuration-27d9719a89a8#.rwt51mfrg.

Kumar, R. (2014). Research methodology: A step-by-step guide for beginners. *SAGE*.

Knutson, T. (2016, March 23). Filesystem Timestamps: What Makes Them Tick? *SANS Institute.* Retrieved from https://www.sans.org/reading-room/whitepapers/forensics/filesystem-timestamps-tick-36842.

Langlois, T. (2016, June 7). Gathering insights from data: An overview of the Elastic Stack. *Opensource.com.* Retrieved from https://opensource.com/life/16/6/overview-elastic-stack.

Lewis, J. (2015, May 25). Detecting Advanced Threats with Sysmon, WEF, and Elasticsearch. *Root9b*. Retrieved from https://www.root9b.com/sites/default/files/whitepapers/R9B_blog_005_whitepaper_01.pdf.

LogRhythm. (2018). Security and Information Event Management | LogRhythm. *LogRhythm.* Retrieved from https://logrhythm.com/solutions/security/siem/.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision support systems, 15*(4), 251-266. Retrieved from https://pdfs.semanticscholar.org/5bd7/3700d5b3bafbbd41d274e8f9c4be81950f39.pdf.

March, S. T., & Storey, V. C. (2008). Design science in the information systems discipline: an introduction to the special issue on design science research. *MIS quarterly*, 725-730. Retrieved from https://pdfs.semanticscholar.org/259b/6e2968c76044be23d4e617f0b78b688dc6e1.pdf.

Malware Archaeology. (n.d.). Sample Sysmon_Config.xml File. Retrieved from https://www.malwarearchaeology.com/s/sysmon_config.7z.

MalwareBytes. (2017, March 29). Explained: Sage ransomware. *MalwareBytes Labs.* Retrieved from https://blog.malwarebytes.com/threat-analysis/2017/03/explained-sage-ransomware/.

MalwareBytes. (n.d.) Ransom.GlobeImposter. *MalwareBytes Blog.* Retrieved from https://blog.malwarebytes.com/detections/ransom-globeimposter.

McDiarmid, D. (2017, May 30). Detecting Signs of Ransomware: WannaCry and the Elastic Stack. *Elastic [Web Log].* Retrieved from https://www.elastic.co/blog/malware-analysis-wannacry-elastic-stack.

Microsoft. (n.d.). PE Format. *Microsoft Windows Dev Center.* Retrieved from https://msdn.microsoft.com/en-us/library/windows/desktop/ms680547(v=vs.85).aspx.

Morgan, S. (2018, January 28). Top 5 Cybersecurity Facts, Figures and Statistics for 2018. *CSO Online*. Retrieved from https://www.csoonline.com/article/3153707/security/top-5-cybersecurity-facts-figures-and-statistics.html.

ncis0x007. (2016, April 14). Sysmon logs at scale analyzed with Splunk. *WordPress Blog*. Retrieved from https://securitylogs.org/2016/05/07/sysmon-logs-at-scale/.

O'Brien, D. (2017, July). Ransomware 2017: An ISTR Special Report. *Symantec.* Retrieved from https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-ransomware-2017-en.pdf.

Oracle. (2018). VirtualBox. *Oracle.* Retrieved from https://www.virtualbox.org.

Palmer, D. (2017, February 21). Ransomware: Why it's a really big problem for small businesses. *ZDNet.* Retrieved from http://www.zdnet.com/article/ransomware-why-its-a-really-big-problem-for-small-businesses/.

Perez, C. (2014, August 10). Sysinternals new tool Sysmon (System Monitor). *Dark Operator Blog.* Retrieved from https://www.darkoperator.com/blog/2014/8/8/sysinternals-sysmon.

Python. (2018a, January 13). IDLE. *Python Software Foundation.* Retrieved from https://docs.python.org/3.6/library/idle.html.

Python. (2018b, January 24). The Python Standard Library. *Python Software Foundation.* Retrieved from https://docs.python.org/3.6/library/index.html

PwC. (2014, September 30). Security Incidents Continue to Rise in Cost and Frequency While Budgets Decrease, according to PwC, CIO, and CSO's The Global State of Information Security Survey 2015". Retrieved from http://www.pwc.com/us/en/press-releases/2014/global-state-of-information-security-survey-2015.html.

Quentin, J. (2017, September 19). Sysmon Events Table. *RawSec [web log]*. Retrieved from https://rawsec.lu/blog/posts/2017/Sep/19/sysmon-events-table/.

Reis, D. (2016, March 2). Solving the Endpoint Exploit Gap: Intelligence-based Exploit Detection for Endpoints. *FireEye Blog.* https://www.fireeye.com/blog/products-and-services/2016/02/solving_the_endpoint.html.

Reynolds, J. & Smith, R. F. (2017, May 24). Using Sysmon to Really See What's Happening on Endpoints. *LogRhythm [webcast]*. Retrieved from https://www.youtube.com/watch?v=M3ptscFkD1w.

Roussev, V. (2009). Hashing and data fingerprinting in digital forensics. *IEEE Security & Privacy, 7*(2). Retrieved from https://www.researchgate.net/profile/Vassil_Roussev/publication/220496865_Hashing_and_Data_Fingerprinting_in_Digital_Forensics/links/0a85e53be9e19089da000000/Hashing-and-Data-Fingerprinting-in-Digital-Forensics.pdf.

Russinovich, M. (2014, November 9). Malware Hunting with Mark Russinovich and the Sysinternals Tools. *TechEd Europe*. Video retrieved from https://www.youtube.com/watch?v=zZdHi6AAEUc.

Russinovich, M. (2015, May 12). Malware Hunting. *TechEd Europe*. Video retrieved from https://www.youtube.com/watch?v=xxf8Tz7QGjU.

Russinovich, M. (2016, March 2). Tracking Hackers on Your Network with Sysinternals Sysmon. Paper presented at the annual meeting of *RSA Conference USA 2016*, San Francisco, CA. Retrieved from https://www.rsaconference.com/writable/presentations/file_upload/hta-w05-tracking_hackers_on_your_network_with_sysinternals_sysmon.pdf.

Russinovich, M. (2017a, February 14). How to go from Responding to Hunting with Sysinternals Sysmon. Paper presented at the annual meeting of *RSA Conference USA 2017*, San Francisco, CA. Retrieved from https://www.rsaconference.com/events/us17/agenda/sessions/7516-How-to-Go-from-Responding-to-Hunting-with-Sysinternals-Sysmon.

Russinovich, M. (2017b, September 17). Process Monitor 3.40. *Microsoft*. Retrieved from https://docs.microsoft.com/en-us/sysinternals/downloads/procmon.

Russinovich, M. & Garnier, T. (2016, August 29). Sysinternals Suite. *Microsoft TechNet.* Retrieved from https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx.

Russinovich, M. & Garnier, T. (2017, November 22). Sysmon v6.20. *Microsoft TechNet.* Retrieved from https://technet.microsoft.com/en-us/sysinternals/sysmon.

Schulze, H. (2017, October 5). 2017 Ransomware Report. *Cybersecurity Insiders.* Retrieved from https://www.alienvault.com/blogs/security-essentials/2017-ransomware-report.

Shilton, K., Subramaniam, M., Vitak, J., & Winter, S. (2016). Qualitative approaches to cybersecurity research. *IConference 2016 Proceedings*. Retrieved from https://www.ideals.illinois.edu/bitstream/handle/2142/89447/Shilton495.pdf?sequence=1&isAllowed=y.

solid IT gmbh. (2018, January). DB-Engines Ranking of Search Engines. *DB-Engines.* Retrieved March 12, 2017 from http://db-engines.com/en/ranking/search+engine.

Splunk. (2017). Splunk. Retrieved from https://www.splunk.com/.

StatCounter. (2017, December). Desktop Operating System Market Share Worldwide. *StatCounter Global Stats.* Retrieved from http://gs.statcounter.com/os-market-share/desktop/worldwide/#monthly-201712-201712-bar.

Steinberg, J. (2018, January 18). 7 Types of Ransomware That You Need To Know About. *Microsoft's Modern Workplace.* Retrieved from http://josephsteinberg.com/ransomware-types/.

Stephenson, P. (2017, May 1). LogRhythm Threat Lifecycle Management Platform. *SC Magazine.* Retrieved from https://www.scmagazine.com/logrhythm-threat-lifecycle-management-platform/review/9357/.

Suricata. (2018). Suricata Open Source IDS/IPS/NSM Engine. *Suricata.* Retrieved from https://suricata-ids.org.s

SwiftOnSecurity. (2017). Sysmon-config. GitHub. Retrieved from https://github.com/SwiftOnSecurity/sysmon-config on September 19, 2017.

SwiftOnSecurity. (2018). Sysmon-config. GitHub. Retrieved from https://github.com/SwiftOnSecurity/sysmon-config on January 17, 2018.

Trend Micro. (2017, August 14). Disk-locking HDDCyrptor/Mamba Ransomware Makes a Comeback. *Trend Micro Inc.* Retrieved from https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/disk-locking-hddcryptor-mamba-ransomware-makes-a-comeback.

Trend Micro. (n.d.). Ransomware. *Trend Micro Inc.* Retrieved from https://www.trendmicro.com/vinfo/us/security/definition/ransomware.

US-CERT. (2016, September 30). Crypto Ransomware Alert TA14-295A. *United States Computer Emergency Readiness Team*. Retrieved from https://www.us-cert.gov/ncas/alerts/TA14-295A.

USC. (2018, January 18). Organizing Your Social Sciences Research Paper: Quantitative Methods. *University of Southern California Libraries.* Retrieved from http://libguides.usc.edu/writingguide/quantitative.

Vaishnavi, V. and Kuechler, W. (2004). Design science research in information systems. Retrieved from http://www.desrist.org/design-research-in-information-systems/.

Verizon. (2017). 2017 Data Breach Investigations Report. *Verizon Insights Lab.* Retrieved from http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/.

VirusTotal. (2018a). VirusTotal Intelligence. *VirusTotal.* Retrieved from https://virustotalcloud.appspot.com/nui/index.html#/intelligence-overview.

VirusTotal. (2018b). VirusTotal Public API v2.0. *VirusTotal.* Retrieved from https://www.virustotal.com/en/documentation/public-api/.

Wikipedia. (2016, October 28). Elasticsearch. Retrieved from https://en.wikipedia.org/wiki/Elasticsearch.

Yara. (2018). Yara – The pattern matching swiss knife for malware researchers. *VirusTotal.* Retrieved from https://virustotal.github.io/yara.

Zeltser, L. (2014, August 12). Using Sysinternals System Monitor (Sysmon) in a Malware Analysis Lab. *SANS Digital Forensics & Incident Response [web log]*. Retrieved from https://digital-forensics.sans.org/blog/2014/08/12/sysmon-in-malware-analysis-lab.

Zeltser, L. (2018, January 10). Malware Samples Sources for Researchers. *Zeltser Security Corp.* Retrieved from https://zeltser.com/malware-sample-sources/.

APPENDICES

APPENDIX A: SYSMON VERSION 6.20 EVENTS

EVENT ID	EVENT DESCRIPTION (Russinovich & Garnier, 2017)
1	Newly created process
2	File creation time was changed by a process
3	TCP/UDP network connection
4	Sysmon service started or stopped
5	Process terminated
6	Driver loaded
7	Module/DLL loaded into a process
8	Thread created in another process
9	Process opened disk or volume for read access
10	Process opened another process
11	File created
12	Registry object created or deleted
13	Registry value was set
14	Registry key and/or value was renamed
15	File stream created
16	Sysmon configuration changed
17	Named pipe created
18	Named pipe connected
19	Windows Management Instrumentation (WMI) Event Filter registered
20	WMI Event Consumer registered
21	WMI Event Consumer bound to WMI Event Filter
255	Sysmon Error

APPENDIX B: SYSMON CONFIGURATION (SYSMONCONFIG-MODIFIED.XML)

The Symon configuration file used in this research was based on the ‘sysmonconfig-export.xml’ configuration file from SwiftOnSecurity (2017). The configuration file was updated to capture trigger events, as well as improve ransomware detection. The configuration file used in this research can be found online at `https://github.com/dsugraduate/dsu2018/`. The configuration is included below, with line numbers displayed for reference.

```
1   <!--
2     sysmon-config | A sysmon configuration focused on default high-quality event tracing and easy customization by the community
3     Master version:   52 | Date: 2017-07-13
4     Master author:    @SwiftOnSecurity, other contributors also credited in-line or on Git.
5     Master project:   https://github.com/SwiftOnSecurity/sysmon-config
6     Master license:   Creative Commons Attribution 4.0 | You may privatize, fork, edit, teach, publish, or deploy for commercial use - with attribution in the text.
7
8     Fork version:     <N/A>
9     Fork author:      Stefani Hobratsch
10    Fork project:     A Holistic Methodology for Profiling Ransomware through Endpoint Detection
11    Fork license:     Creative Commons Attribution 4.0 | You may privatize, fork, edit, teach, publish, or deploy for commercial use - with attribution in the text.
12
13    REQUIRED: Sysmon version 6.00 or higher (due to changes in registry syntax)
14    https://technet.microsoft.com/en-us/sysinternals/bb545021.aspx
15    Note that 6.03 has important fixes for filtering
16
17    NOTE: Do not let the imposing size and complexity of this configuration scare you off building your own or customizing it.
18    This configuration is based around known high-quality event tracing, and thus looks extremely complicated.
19    Sysmon configurations only have to be a few lines, but significant effort has been invested in front-loading as
20    much filtering as possible onto the client. This is to make analysis of intrusions possible by hand, and try to
21    surface anomalous activity as quickly as possible to any technician armed only with Event Viewer.
22
23    NOTE: Sysmon is not hardened against a determined attacker with admin rights. Also, this configuration offers an attacker, willing
```

```
    to study it closely, several ways to evade some of the alerting. If you are in a high-threat environment and have significant
    security staff, you should consider a much broader log-all approach. However, in the vast majority of cases, an attacker
    will bumble along through multiple behavioral traps which this configuration monitors, especially in the first minutes.

    NOTE: "Image" is a technical term for a compiled binary file like an EXE or DLL. Also, it can match just the filename, or entire path.
          "ProcessGuid" is randomly generated, assigned, and tracked by Sysmon to assist in tracing individual process launches.
          "LoginGuid" is randomly generated, assigned, and tracked by Sysmon to assist in tracing individual user sessions.
-->
<!-- TESTING (changed version number) -->
<Sysmon schemaversion="3.40">
  <HashAlgorithms>md5,sha256</HashAlgorithms>
  <EventFiltering>

  <!--SYSMON EVENT ID 1 : PROCESS CREATION-->
          <!--DATA: UtcTime, ProcessGuid, ProcessID, Image, CommandLine, CurrentDirectory, User, LogonGuid, LogonId, TerminalSessionId, IntegrityLevel, Hashes, ParentProcessGuid, ParentProcessId, ParentImage, ParentCommandLine-->
          <ProcessCreate onmatch="exclude">
          <!--COMMENT:      All process launched will be included, except for what matches a rule below. It's best to be as specific as possible, to
                            avoid user-mode executables imitating other process names to avoid logging, or if malware drops files in an existing directory.
                            Ultimately, you must weigh CPU time checking many detailed rules, against the risk of malware exploiting the blindness created.-->
                  <!--SECTION: Microsoft Windows-->
                  <CommandLine condition="begin with">C:\Windows\system32\DllHost.exe /Processid</CommandLine> <!--Microsoft:Windows-->
                  <CommandLine condition="is">C:\Windows\system32\SearchIndexer.exe /Embedding</CommandLine> <!--Microsoft:Windows: Search Indexer-->
                  <Image condition="end with">C:\Windows\System32\CompatTelRunner.exe</Image> <!--Microsoft:Windows:Customer Experience Improvement-->
                  <Image condition="is">C:\Windows\System32\MusNotification.exe</Image> <!--Microsoft:Windows: Update popups-->
                  <Image condition="is">C:\Windows\System32\MusNotificationUx.exe</Image> <!--Microsoft:Windows: Update popups-->
                  <Image condition="is">C:\Windows\System32\audiodg.exe</Image> <!--Microsoft:Windows: Launched constantly-->
                  <Image condition="is">C:\Windows\System32\conhost.exe</Image> <!--Microsoft:Windows: Command line interface host process-->
                  <Image condition="is">C:\Windows\System32\powercfg.exe</Image> <!--Microsoft:Power configuration management-->
```

```
                <Image condition="is">C:\Windows\System32\wbem\WmiApSrv.exe</Image> <!--Microsoft:Windows: WMI performance adpater host process-->
                <Image condition="is">C:\Windows\System32\wermgr.exe</Image> <!--Microsoft:Windows:Windows error reporting/telemetry-->
                <Image condition="is">C:\Windows\SysWOW64\wermgr.exe</Image> <!--Microsoft:Windows:Windows error reporting/telemetry-->
                <Image condition="is">C:\Windows\system32\sppsvc.exe</Image> <!--Microsoft:Windows: Software Protection Service-->
                <IntegrityLevel condition="is">AppContainer</IntegrityLevel> <!--Microsoft:Windows: Don't care about sandboxed processes-->
                <ParentCommandLine condition="begin with">%%SystemRoot%%\system32\csrss.exe ObjectDirectory=\Windows</ParentCommandLine> <!--Microsoft:Windows:CommandShell: Triggered when programs use the command shell, but without attribution-->
                <ParentImage condition="is">C:\Windows\system32\SearchIndexer.exe</ParentImage> <!--Microsoft:Windows:Search: Launches many uninteresting sub-processes-->
                <!--SECTION: Microsoft:Windows:Defender-->
                <Image condition="begin with">C:\Program Files\Windows Defender</Image> <!--Microsoft:Windows:Defender in Win10-->
                <Image condition="is">C:\Windows\System32\MpSigStub.exe</Image> <!--Microsoft:Windows: Microsoft Malware Protection Signature Update Stub-->
                <Image condition="begin with">C:\Windows\SoftwareDistribution\Download\Install\AM_Base</Image> <!--Microsoft:Defender: Full signature updates-->
                <Image condition="begin with">C:\Windows\SoftwareDistribution\Download\Install\AM_Delta</Image> <!--Microsoft:Defender: Delta signature updates-->
                <Image condition="begin with">C:\Windows\SoftwareDistribution\Download\Install\AM_Engine</Image> <!--Microsoft:Defender: Engine updates-->
                <!--SECTION: Microsoft:Windows:svchost-->
                <CommandLine condition="is">C:\Windows\System32\svchost.exe -k appmodel</CommandLine> <!--Microsoft:Windows 10-->
                <CommandLine condition="is">C:\Windows\System32\svchost.exe -k dcomLaunch</CommandLine> <!--Microsoft:Windows devices-->
                <CommandLine condition="is">C:\Windows\System32\svchost.exe -k defragsvc</CommandLine> <!--Microsoft:Windows defrag-->
                <CommandLine condition="is">C:\Windows\System32\svchost.exe -k imgsvc</CommandLine> <!--Microsoft:The Windows Image Acquisition Service-->
                <CommandLine condition="is">C:\Windows\System32\svchost.exe -k localServiceAndNoImpersonation</CommandLine> <!--Microsoft:Windows: Network services-->
                <CommandLine condition="is">C:\Windows\System32\svchost.exe -k
```

```
localServiceNetworkRestricted</CommandLine> <!--Microsoft:Windows: Network
services-->
                  <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k
localSystemNetworkRestricted</CommandLine> <!--Microsoft:Windows: Network
services-->
                  <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k netsvcs</CommandLine> <!-
-Microsoft:Windows: Network services-->
                  <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k
networkServiceNetworkRestricted</CommandLine> <!--Microsoft:Windows:
Network services-->
                  <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k rPCSS</CommandLine> <!--
Microsoft:Windows Services-->
                  <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k swprv</CommandLine> <!--
Microsoft:Software Shadow Copy Provider-->
                  <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k
unistackSvcGroup</CommandLine> <!--Microsoft:Windows 10-->
                  <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k utcsvc</CommandLine> <!--
Microsoft:Windows Services-->
                  <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k
wbioSvcGroup</CommandLine> <!--Microsoft:Windows Services-->
                  <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k wsappx</CommandLine> <!--
Microsoft:Windows 10-->
                  <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
networkService</CommandLine> <!--Microsoft:Windows: Network services-->
                  <CommandLine
condition="is">C:\windows\System32\svchost.exe -k werSvcGroup</CommandLine>
<!--Microsoft:Windows: ErrorReporting-->
                  <ParentCommandLine
condition="is">C:\Windows\System32\svchost.exe -k
netsvcs</ParentCommandLine> <!--Microsoft:Windows: Network services: Spawns
Consent.exe-->
                  <ParentCommandLine
condition="is">C:\Windows\system32\svchost.exe -k
LocalSystemNetworkRestricted</ParentCommandLine> <!--Microsoft:Windows:
Network services-->
                  <!--SECTION: Microsoft:dotNet-->
                  <CommandLine condition="begin
with">C:\Windows\Microsoft.NET\Framework\v4.0.30319\ngen.exe</CommandLine>
<!--Microsoft:DotNet-->
                  <Image
condition="is">C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscorsvw.exe
</Image> <!--Microsoft:DotNet-->
                  <Image
condition="is">C:\Windows\Microsoft.NET\Framework\v4.0.30319\mscorsvw.exe</
Image> <!--Microsoft:DotNet-->
```

```
        <Image condition="is">C:\Windows\Microsoft.Net\Framework64\v3.0\WPF\PresentationFontCache.exe</Image> <!--Microsoft:Windows: Font cache service-->
        <Image condition="is">C:\Windows\Microsoft.Net\Framework64\v3.0\WPF\PresentationFontCache.exe</Image> <!--Microsoft:Windows: Font cache service-->
        <ParentCommandLine condition="contains">C:\Windows\Microsoft.NET\Framework64\v4.0.30319\ngentask.exe</ParentCommandLine>
        <ParentImage condition="is">C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscorsvw.exe </ParentImage> <!--Microsoft:DotNet-->
        <ParentImage condition="is">C:\Windows\Microsoft.NET\Framework64\v4.0.30319\ngentask.exe </ParentImage> <!--Microsoft:DotNet-->
        <ParentImage condition="is">C:\Windows\Microsoft.NET\Framework\v4.0.30319\mscorsvw.exe</ParentImage> <!--Microsoft:DotNet-->
        <ParentImage condition="is">C:\Windows\Microsoft.NET\Framework\v4.0.30319\ngentask.exe</ParentImage> <!--Microsoft:DotNet-->
        <ParentImage condition="is">C:\Windows\Microsoft.NET\Framework\v4.0.30319\ngentask.exe</ParentImage> <!--Microsoft:DotNet: Spawns thousands of ngen.exe processes-->
        <!--SECTION: Microsoft:Office-->
        <Image condition="is">C:\Program Files (x86)\Microsoft Office\Office16\MSOSYNC.EXE</Image> <!--Microsoft:Office: Background process-->
        <Image condition="is">C:\Program Files\Common Files\Microsoft Shared\OfficeSoftwareProtectionPlatform\OSPPSVC.EXE</Image> <!--Microsoft:Office: Background process-->
        <!--SECTION: Microsoft:Office:Click2Run-->
        <Image condition="is">C:\Program Files\Common Files\Microsoft Shared\ClickToRun\OfficeC2RClient.exe</Image> <!--Microsoft:Office: Background process-->
        <ParentImage condition="end with">C:\Program Files\Common Files\Microsoft Shared\ClickToRun\OfficeClickToRun.exe</ParentImage> <!--Microsoft:Office: Background process-->
        <ParentImage condition="is">C:\Program Files\Common Files\Microsoft Shared\ClickToRun\OfficeC2RClient.exe</ParentImage> <!--Microsoft:Office: Background process-->
        <!--SECTION: Google-->
        <CommandLine condition="begin with">"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --type=</CommandLine> <!--Google:Chrome: massive command-line arguments-->
        <CommandLine condition="begin with">"C:\Program Files\Google\Chrome\Application\chrome.exe" --type=</CommandLine> <!--Google:Chrome: massive command-line arguments-->
        <Image condition="begin with">C:\Program Files (x86)\Google\Update\</Image> <!--Google:Chrome: Updater-->
        <ParentImage condition="begin with">C:\Program Files (x86)\Google\Update\</ParentImage> <!--Google:Chrome: Updater-->
        <!--SECTION: Firefox-->
        <CommandLine condition="begin with">"C:\Program Files\Mozilla Firefox\plugin-container.exe" --channel</CommandLine> <!--Mozilla:Firefox: Large command-line arguments | Credit @Darkbat91 -->
```

```
				<CommandLine condition="begin with">"C:\Program Files (x86)\Mozilla Firefox\plugin-container.exe" --channel</CommandLine> <!--Mozilla:Firefox: Large command-line arguments | Credit @Darkbat91 -->
				<!--SECTION: Adobe-->
				<CommandLine condition="contains">AcroRd32.exe" /CR </CommandLine> <!--Adobe:AcrobatReader: Uninteresting sandbox subprocess-->
				<CommandLine condition="contains">AcroRd32.exe" --channel=</CommandLine> <!--Adobe:AcrobatReader: Uninteresting sandbox subprocess-->
				<Image condition="end with">C:\Program Files (x86)\Adobe\Acrobat DC\Acrobat\AcroCEF\AcroCEF.exe</Image> <!--Adobe:Acrobat: Sandbox subprocess, still evaluating security exposure-->
				<ParentImage condition="end with">C:\Program Files (x86)\Common Files\Adobe\AdobeGCClient\AGSService.exe</ParentImage>
				<Image condition="end with">C:\Program Files (x86)\Adobe\Acrobat Reader DC\Reader\AcroCEF\RdrCEF.exe</Image> <!--Adobe:AcrobatReader: Sandbox subprocess, still evaluating security exposure-->
				<Image condition="end with">C:\Program Files (x86)\Adobe\Acrobat DC\Acrobat\LogTransport2.exe</Image>
				<!--SECTION: Adobe:Flash-->
				<Image condition="end with">C:\Windows\SysWOW64\Macromed\Flash\FlashPlayerUpdateService.exe</Image> <!--Adobe:Flash: Properly hardened updater, not a risk-->
				<!--SECTION: Adobe:Updater-->
				<Image condition="end with">C:\Program Files (x86)\Common Files\Adobe\ARM\1.0\AdobeARM.exe</Image> <!--Adobe:Updater: Properly hardened updater, not a risk-->
				<ParentImage condition="end with">C:\Program Files (x86)\Common Files\Adobe\ARM\1.0\AdobeARM.exe</ParentImage> <!--Adobe:Updater: Properly hardened updater, not a risk-->
				<Image condition="end with">C:\Program Files (x86)\Common Files\Adobe\ARM\1.0\armsvc.exe</Image> <!--Adobe:Updater: Properly hardened updater, not a risk-->
				<!--SECTION: Adobe:Supporting processes-->
				<Image condition="end with">C:\Program Files (x86)\Adobe\Acrobat DC\Acrobat\AdobeCollabSync.exe</Image>
				<Image condition="end with">C:\Program Files (x86)\Common Files\Adobe\Adobe Desktop Common\HEX\Adobe CEF Helper.exe</Image>
				<Image condition="end with">C:\Program Files (x86)\Common Files\Adobe\AdobeGCClient\AdobeGCClient.exe</Image> <!--Adobe:Creative Cloud-->
				<Image condition="end with">C:\Program Files (x86)\Common Files\Adobe\OOBE\PDApp\P6\adobe_licutil.exe</Image> <!--Adobe:License utility-->
				<Image condition="end with">C:\Program Files (x86)\Common Files\Adobe\OOBE\PDApp\P7\adobe_licutil.exe</Image> <!--Adobe:License utility-->
				<ParentImage condition="end with">C:\Program Files (x86)\Common Files\Adobe\OOBE\PDApp\P7\adobe_licutil.exe</ParentImage> <!--Adobe:License utility-->
				<Image condition="end with">C:\Program Files (x86)\Common Files\Adobe\OOBE\PDApp\UWA\updaterstartuputility.exe</Image>
				<ParentImage condition="is">C:\Program Files (x86)\Common Files\Adobe\OOBE\PDApp\UWA\updaterstartuputility.exe</ParentImage>
				<!--SECTION: Adobe:Creative Cloud-->
```

```
            <Image condition="end with">C:\Program Files (x86)\Adobe\Adobe Creative Cloud\ACC\Creative Cloud.exe</Image>
            <ParentImage condition="end with">C:\Program Files (x86)\Adobe\Adobe Creative Cloud\ACC\Creative Cloud.exe</ParentImage>
            <ParentImage condition="end with">C:\Program Files (x86)\Adobe\Adobe Creative Cloud\CCXProcess\CCXProcess.exe</ParentImage>
            <ParentImage condition="end with">C:\Program Files (x86)\Adobe\Adobe Creative Cloud\CoreSync\CoreSync.exe</ParentImage>
            <!--SECTION: Drivers-->
            <CommandLine condition="begin with">"C:\Program Files\DellTPad\ApMsgFwd.exe" -s{</CommandLine>
            <Image condition="begin with">C:\Program Files\NVIDIA Corporation\</Image> <!--Nvidia:Driver: routine actions-->
            <Image condition="begin with">C:\Program Files\Realtek\</Image> <!--Realtek:Driver: routine actions-->
            <ParentImage condition="end with">C:\Program Files\DellTPad\HidMonitorSvc.exe</ParentImage>
            <ParentImage condition="end with">C:\Program Files\Realtek\Audio\HDA\RtkAudioService64.exe</ParentImage> <!--Realtek:Driver: routine actions-->
            <!--SECTION: Dropbox-->
            <Image condition="end with">C:\Program Files (x86)\Dropbox\Update\DropboxUpdate.exe</Image> <!--Dropbox:Updater: Lots of command-line arguments-->
            <ParentImage condition="end with">C:\Program Files (x86)\Dropbox\Update\DropboxUpdate.exe</ParentImage>
            <!--SECTION: Dell-->
            <ParentImage condition="image">C:\Program Files (x86)\Dell\CommandUpdate\InvColPC.exe</ParentImage> <!--Dell:CommandUpdate: Detection process-->
        </ProcessCreate>

    <!--SYSMON EVENT ID 2 : FILE CREATION TIME RETROACTIVELY CHANGED IN THE FILESYSTEM-->
        <!--DATA: UtcTime, ProcessGuid, ProcessId, Image, TargetFilename, CreationUtcTime, PreviousCreationUtcTime-->
        <FileCreateTime onmatch="include">
            <Image condition="begin with">C:\Users</Image> <!--Look for timestomping processes in user area-->
            <TargetFilename condition="end with">trigger.txt</TargetFilename> <!--Look for Trigger -->
        </FileCreateTime>
        <FileCreateTime onmatch="exclude">
            <Image condition="image">OneDrive.exe</Image> <!--OneDrive constantly changes file times-->
            <Image condition="contains">setup</Image> <!--Ignore setups-->
        </FileCreateTime>



    <!--SYSMON EVENT ID 3 : NETWORK CONNECTION INITIATED-->
        <!--DATA: UtcTime, ProcessGuid, ProcessId, Image, User, Protocol, Initiated, SourceIsIpv6, SourceIp, SourceHostname, SourcePort, SourcePortName, DestinationIsIpV6, DestinationIP, DestinationHostname, DestinationPort, DestinationPortName-->
        <NetworkConnect onmatch="include">
        <!--COMMENT:	Takes a very conservative approach to network logging, limit to extremely high-signal events.-->
        <!--TECHNICAL:	For the DestinationHostname, Sysmon uses the GetNameInfo API, which may not always have the information or may be a CDN. Using that field is best-effort only.-->
        <!--TECHNICAL:	These exe's do not initiate their connections, and cannot be included: BITSADMIN-->
                <Image condition="end with">trigger.exe</Image>  <!--Look for Trigger -->
                <!--Suspicious sources-->
                <Image condition="begin with">C:\Users</Image> <!--Tools downloaded by users can use other processes for networking, but this is a very valuable indicator.-->
                <Image condition="begin with">C:\ProgramData</Image> <!--Normally, network communications should be sourced from "Program Files" not from ProgramData, something to look at-->
                <Image condition="begin with">C:\Windows\Temp</Image> <!--Suspicious anything would communicate from the system-level temp directory-->
                <!--Suspicious Windows tools-->
                <Image condition="image">at.exe</Image> <!--Microsoft:Windows: Remote task scheduling | Credit @ion-storm -->
                <Image condition="image">certutil.exe</Image> <!--Microsoft:Windows: Certificate tool can contact outbound | Credit @ion-storm and @FVT [ https://twitter.com/FVT/status/834433734602530817 ] -->
                <Image condition="image">cmd.exe</Image> <!--Microsoft:Windows: Command prompt-->
                <Image condition="image">cscript.exe</Image> <!--Microsoft:WindowsScriptingHost: | Credit @Cyb3rOps [ https://gist.github.com/Neo23x0/a4b4af9481e01e749409 ] -->
                <Image condition="image">java.exe</Image> <!--Java: Monitor usage of vulnerable application | Credit @ion-storm -->
                <Image condition="image">mshta.exe</Image> <!--Microsoft:Windows: HTML application executes scripts without IE protections | Credit @ion-storm [ https://en.wikipedia.org/wiki/HTML_Application ] -->
                <Image condition="image">msiexec.exe</Image> <!--Microsoft:Windows: Can install from http:// paths | Credit @vector-sec -->
                <Image condition="image">net.exe</Image> <!--Microsoft:Windows: "net use"/"net view" used by attackers to surveil and connect with file shares from command line | Credit @ion-storm -->
                <Image condition="image">notepad.exe</Image> <!--Microsoft:Windows: [ https://blog.cobaltstrike.com/2013/08/08/why-is-notepad-exe-connecting-to-the-internet/ ] -->
                <Image condition="image">powershell.exe</Image> <!--Microsoft:Windows: PowerShell interface-->
                <Image condition="image">qwinsta.exe</Image> <!--Microsoft:Windows: Remotely query login sessions on a server or workstation | Credit @ion-storm -->
                <Image condition="image">reg.exe</Image> <!--Microsoft:Windows: Remote Registry | Credit @ion-storm -->
                <Image condition="image">regsvr32.exe</Image> <!--Microsoft:Windows: [ https://subt0x10.blogspot.com/2016/04/bypass-application-whitelisting-script.html ] -->
                <Image condition="image">rundll32.exe</Image> <!--Microsoft:Windows: [ https://blog.cobaltstrike.com/2016/07/22/why-is-rundll32-exe-connecting-to-the-internet/ ] -->
```

```
			<Image condition="image">sc.exe</Image> <!--Microsoft:Windows: Remotely change Windows service settings from command line | Credit @ion-storm -->
			<Image condition="image">wmic.exe</Image> <!--Microsoft:WindowsManagementInstrumentation: Credit @Cyb3rOps [ https://gist.github.com/Neo23x0/a4b4af9481e01e749409 ] -->
			<Image condition="image">wscript.exe</Image> <!--Microsoft:WindowsScriptingHost: | Credit @arekfurt -->
			<!--Relevant 3rd Party Tools: Remote Access-->
			<Image condition="image">psexec.exe</Image> <!--Sysinternals:PsExec client side | Credit @Cyb3rOps -->
			<Image condition="image">psexesvc.exe</Image> <!--Sysinternals:PsExec server side | Credit @Cyb3rOps -->
			<Image condition="image">vnc.exe</Image> <!-- VNC client | Credit @Cyb3rOps -->
			<Image condition="image">vncviewer.exe</Image> <!-- VNC client | Credit @Cyb3rOps -->
			<Image condition="image">vncservice.exe</Image> <!-- VNC server | Credit @Cyb3rOps -->
			<Image condition="image">winexesvc.exe</Image> <!--Winexe service executable | Credit @Cyb3rOps -->
			<Image condition="image">\AA_v</Image> <!-- Ammy Admin service executable (e.g. AA_v3.0.exe AA_v3.5.exe ) | Credit @Cyb3rOps -->
			<!-- Often exploited services -->
			<Image condition="image">omniinet.exe</Image> <!-- HP Data Protector https://www.cvedetails.com/vulnerability-list/vendor_id-10/product_id-20499/HP-Data-Protector.html | Credit @Cyb3rOps -->
			<Image condition="image">hpsmhd.exe</Image> <!-- HP System Management Homepage https://www.cvedetails.com/vulnerability-list/vendor_id-10/product_id-7244/HP-System-Management-Homepage.html | Credit @Cyb3rOps -->
			<!--Malware related-->
			<Image condition="image">tor.exe</Image> <!--Tor [ https://www.hybrid-analysis.com/sample/800bf028a23440134fc834efc5c1e02cc70f05b2e800bbc285d7c92a4b126b1c?environmentId=100 ] -->
			<!--Ports: Suspicious-->
			<DestinationPort condition="is">22</DestinationPort> <!--SSH protocol-->
			<DestinationPort condition="is">23</DestinationPort> <!--Telnet protocol-->
			<DestinationPort condition="is">25</DestinationPort> <!--SMTP mail protocol-->
			<DestinationPort condition="is">3389</DestinationPort> <!--Microsoft:Windows:RDP-->
			<DestinationPort condition="is">5800</DestinationPort> <!--VNC protocol-->
			<DestinationPort condition="is">5900</DestinationPort> <!--VNC protocol-->
			<!--Ports: Proxy-->
			<DestinationPort condition="is">1080</DestinationPort> <!--Socks proxy port | Credit @ion-storm-->
			<DestinationPort condition="is">3128</DestinationPort> <!--Socks proxy port | Credit @ion-storm-->
			<DestinationPort condition="is">8080</DestinationPort> <!--Socks proxy port | Credit @ion-storm-->
			<!--Ports: Tor-->
```

```
            <DestinationPort condition="is">1723</DestinationPort> <!--Tor protocol | Credit @ion-storm-->
            <DestinationPort condition="is">4500</DestinationPort> <!--Tor protocol | Credit @ion-storm-->
            <DestinationPort condition="is">9001</DestinationPort> <!--Tor protocol [ http://www.computerworlduk.com/tutorial/security/tor-enterprise-2016-blocking-malware-darknet-use-rogue-nodes-3633907/ ] -->
            <DestinationPort condition="is">9030</DestinationPort> <!--Tor protocol [ http://www.computerworlduk.com/tutorial/security/tor-enterprise-2016-blocking-malware-darknet-use-rogue-nodes-3633907/ ] -->
        </NetworkConnect>
        <NetworkConnect onmatch="exclude">
            <Image condition="image">OneDrive.exe</Image> <!--Microsoft:OneDrive-->
            <Image condition="image">Spotify.exe</Image> <!--Spotify-->
            <Image condition="end with">AppData\Roaming\Dropbox\bin\Dropbox.exe</Image> <!--Dropbox-->
            <!--SECTION: Microsoft-->
            <Image condition="image">OneDriveStandaloneUpdater.exe</Image> <!--Microsoft:OneDrive-->
            <DestinationHostname condition="end with">microsoft.com</DestinationHostname> <!--Microsoft:Update delivery-->
            <DestinationHostname condition="end with">microsoft.com.akadns.net</DestinationHostname> <!--Microsoft:Update delivery-->
            <DestinationHostname condition="end with">microsoft.com.nsatc.net</DestinationHostname> <!--Microsoft:Update delivery-->
        </NetworkConnect>

    <!--SYSMON EVENT ID 4 : RESERVED FOR SYSMON STATUS MESSAGES, THIS LINE IS INCLUDED FOR DOCUMENTATION PURPOSES ONLY-->
        <!--DATA: UtcTime, State, Version, SchemaVersion-->
        <!--Cannot be filtered.-->

    <!--SYSMON EVENT ID 5 : PROCESS ENDED-->
        <!--DATA: UtcTime, ProcessGuid, ProcessId, Image-->
        <ProcessTerminate onmatch="include">
        <!--COMMENT:    Useful data in building infection timelines.-->
            <Image condition="begin with">C:\Users</Image> <!--Process terminations by user binaries-->
            <Image condition="end with">trigger.exe</Image>  <!--Look for Trigger -->
        </ProcessTerminate>

    <!--SYSMON EVENT ID 6 : DRIVER LOADED INTO KERNEL-->
        <!--DATA: UtcTime, ImageLoaded, Hashes, Signed, Signature, SignatureStatus-->
        <DriverLoad onmatch="exclude">
        <!--COMMENT:    Because drivers with bugs can be used to escalate to kernel permissions, be extremely selective
                about what you exclude from monitoring. Low event volume, little incentive to exclude.-->
```

```
              <Signature condition="contains">microsoft</Signature> <!--Exclude signed Microsoft drivers-->
              <Signature condition="contains">windows</Signature> <!--Exclude signed Microsoft drivers-->
              <Signature condition="begin with">Intel </Signature> <!--Exclude signed Intel drivers-->
        </DriverLoad>

  <!--SYSMON EVENT ID 7 : DLL (IMAGE) LOADED BY PROCESS-->
        <!--DATA: UtcTime, ProcessGuid, ProcessId, Image, ImageLoaded, Hashes, Signed, Signature, SignatureStatus-->
        <ImageLoad onmatch="include">
        <!--COMMENT:      Can cause high system load, disabled by default, important examples included below.-->
              <ImageLoaded condition="end with">trigger.dll</ImageLoaded>  <!--Look for Trigger -->

              <!-- <ImageLoaded condition="contains">system.automation</ImageLoaded> -->
              <!-- <ImageLoaded condition="image">wshom.ocx</ImageLoaded> -->
              <!-- <ImageLoaded condition="image">vbscript.dll</ImageLoaded> -->
              <!-- <ImageLoaded condition="image">javascript.dll</ImageLoaded> -->
              <!-- <ImageLoaded condition="contains">msxml4</ImageLoaded> -->
              <!-- <ImageLoaded condition="image">hal.dll</ImageLoaded> -->
              <!-- <ImageLoaded condition="image">scrrun.dll</ImageLoaded> -->
              <!-- <ImageLoaded condition="contains">npjpi</ImageLoaded> -->
              <!-- <ImageLoaded condition="image">jp2iexp.dll</ImageLoaded> -->
              <!-- Mimikatz -->
                    <!--NOTES: [ https://securityriskadvisors.com/blog/post/detecting-in-memory-mimikatz/ ] -->
              <!-- <ImageLoaded condition="image">wdigest.dll</ImageLoaded> -->
        </ImageLoad>

  <!--SYSMON EVENT ID 8 : REMOTE THREAD CREATED-->
        <!--DATA: UtcTime, SourceProcessGuid, SourceProcessId, SourceImage, TargetProcessId, TargetImage, NewThreadId, StartAddress, StartModule, StartFunction-->
        <CreateRemoteThread onmatch="exclude">
        <!--COMMENT:      Monitor for processes injecting code into other processes. Often used by malware to cloak their actions.
                    Exclude mostly-safe sources and log anything else.-->
              <SourceImage condition="is">C:\Windows\System32\wbem\WmiPrvSE.exe</SourceImage>
              <SourceImage condition="is">C:\Windows\System32\svchost.exe</SourceImage>
```

```
            <SourceImage condition="is">C:\Windows\System32\wininit.exe</SourceImage>
            <SourceImage condition="is">C:\Windows\System32\csrss.exe</SourceImage>
            <SourceImage condition="is">C:\Windows\System32\services.exe</SourceImage>
            <SourceImage condition="is">C:\Windows\System32\winlogon.exe</SourceImage>
            <SourceImage condition="is">C:\Windows\System32\audiodg.exe</SourceImage>
            <StartModule condition="is">C:\windows\system32\kernel32.dll</StartModule>
            <TargetImage condition="end with">Google\Chrome\Application\chrome.exe</TargetImage>
        </CreateRemoteThread>

    <!--SYSMON EVENT ID 9 : RAW DISK ACCESS-->
        <!--DATA: UtcTime, ProcessGuid, ProcessId, Image, Device-->
        <RawAccessRead onmatch="include">
        <!--COMMENT:      Monitor for raw sector-level access to the disk, often used to bypass access control lists or access locked files.
                        Disabled by default since including even one entry here activates this component. Reward/performance/rule maintenance decision.
                        Encourage you to experiment with this feature yourself.-->
        <!--COMMENT:      You will likely want to set this to a full capture on domain controllers, where no process should be doing raw reads.-->
            <Image condition="end with">powershell.exe</Image>
            <Image condition="end with">wmic.exe</Image>
        </RawAccessRead>

    <!--SYSMON EVENT ID 10 : INTER-PROCESS ACCESS-->
        <!--DATA: UtcTime, SourceProcessGuid, SourceProcessId, SourceThreadId, SourceImage, TargetProcessGuid, TargetProcessId, TargetImage, GrantedAccess, CallTrace-->
        <ProcessAccess onmatch="include"> <!--TEST-->
        <!--COMMENT:      Monitor for processes accessing other process' memory. This can be valuable, but can cause a huge number of events.-->
            <SourceImage condition="end with">trigger.exe</SourceImage>  <!--Look for Trigger -->
        </ProcessAccess>

    <!--SYSMON EVENT ID 11 : FILE CREATED-->
        <!--DATA: UtcTime, ProcessGuid, ProcessId, Image, TargetFilename, CreationUtcTime-->
        <FileCreate onmatch="include">
            <TargetFilename condition="contains">\Start Menu</TargetFilename> <!--Microsoft:Windows: Startup links and shortcut modification-->
            <TargetFilename condition="contains">\Startup</TargetFilename> <!--Microsoft:Office: Changes to user's autoloaded files under AppData-->
```

```
			<TargetFilename condition="contains">\Content.Outlook\</TargetFilename> <!--Microsoft:Outlook: attachments--> <!--PRIVACY WARNING-->
			<TargetFilename condition="contains">\Downloads\</TargetFilename> <!--Downloaded files. Does not include "Run" files in IE--> <!--PRIVACY WARNING-->
			<TargetFilename condition="end with">.application</TargetFilename> <!--Microsoft:ClickOnce: [ https://blog.netspi.com/all-you-need-is-one-a-clickonce-love-story/ ] -->
			<TargetFilename condition="end with">.appref-ms</TargetFilename> <!--Microsoft:ClickOnce application | Credit @ion-storm -->
			<TargetFilename condition="end with">.bat</TargetFilename> <!--Batch scripting-->
			<TargetFilename condition="end with">.cmd</TargetFilename> <!--Batch scripting: Batch scripts can also use the .cmd extension | Credit: @mmazanec -->
			<TargetFilename condition="end with">.cmdline</TargetFilename> <!--Microsoft:dotNet: Executed by cvtres.exe-->
			<TargetFilename condition="end with">.docm</TargetFilename> <!--Microsoft:Office:Word: Macro-->
			<TargetFilename condition="end with">.exe</TargetFilename> <!--Executable-->
			<TargetFilename condition="end with">.hta</TargetFilename> <!--Scripting-->
			<TargetFilename condition="end with">.pptm</TargetFilename> <!--Microsoft:Office:Word: Macro-->
			<TargetFilename condition="end with">.ps1</TargetFilename> <!--PowerShell [ More information: http://www.hexacorn.com/blog/2014/08/27/beyond-good-ol-run-key-part-16/ ] -->
			<TargetFilename condition="end with">.sys</TargetFilename> <!--System driver files-->
			<TargetFilename condition="end with">.vbs</TargetFilename> <!--VisualBasicScripting-->
			<TargetFilename condition="end with">.xlsm</TargetFilename> <!--Microsoft:Office:Word: Macro-->
			<TargetFilename condition="begin with">C:\Users\Default</TargetFilename> <!--Microsoft:Windows: Changes to default user profile-->
			<TargetFilename condition="begin with">C:\Windows\System32\Drivers</TargetFilename> <!--Microsoft: Drivers dropped here-->
			<TargetFilename condition="begin with">C:\Windows\SysWOW64\Drivers</TargetFilename> <!--Microsoft: Drivers dropped here-->
			<TargetFilename condition="begin with">C:\Windows\System32\GroupPolicy\Machine\Scripts</TargetFilename> <!--Group policy [ More information: http://www.hexacorn.com/blog/2017/01/07/beyond-good-ol-run-key-part-52/ ] -->
			<TargetFilename condition="begin with">C:\Windows\System32\GroupPolicy\User\Scripts</TargetFilename> <!--Group policy [ More information: http://www.hexacorn.com/blog/2017/01/07/beyond-good-ol-run-key-part-52/ ] -->
```

```
                  <TargetFilename condition="begin with">C:\Windows\System32\Tasks</TargetFilename> <!--Microsoft:ScheduledTasks-->
                  <TargetFilename condition="begin with">C:\Windows\System32\Wbem</TargetFilename> <!--Microsoft:WMI: [ More information: http://2014.hackitoergosum.org/slides/day1_WMI_Shell_Andrei_Dumitrescu.pdf ] -->
                  <TargetFilename condition="begin with">C:\Windows\SysWOW64\Wbem</TargetFilename> <!--Microsoft:WMI: [ More information: http://2014.hackitoergosum.org/slides/day1_WMI_Shell_Andrei_Dumitrescu.pdf ] -->
                  <TargetFilename condition="begin with">C:\Windows\System32\WindowsPowerShell</TargetFilename> <!--Microsoft:Powershell: Look for modifications for persistence [ https://www.malwarearchaeology.com/cheat-sheets ] -->
                  <TargetFilename condition="begin with">C:\Windows\SysWOW64\WindowsPowerShell</TargetFilename> <!--Microsoft:Powershell: Look for modifications for persistence [ https://www.malwarearchaeology.com/cheat-sheets ] -->
                  <TargetFilename condition="begin with">C:\Windows\Tasks\</TargetFilename> <!--Microsoft:ScheduledTasks-->
                  <!-- TEST -->
                  <TargetFilename condition="end with">.402</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end with">.4035</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end with">.4090</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end with">.4091</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end with">.452</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end with">.707</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end with">.725</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end with">.726</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end with">.911</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end with">.f41o1</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end with">.2cXpCihgsVxB3</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
```

```
                <TargetFilename condition="end with">.3ncrypt3d</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.au1crypt</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.BONUM</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.BRT92</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.BUSH</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.C8B089F</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.CHAK</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.clinTON</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.crypt</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.FIX</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.fuck</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.goro</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.gotham</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.granny</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.happ</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.lpcrestore</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.keepcalm</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.LIN</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.MAKB</TargetFilename> <!--
```

```
https://blog.malwarebytes.com/detections/ransome-globeimposter -->

                  <TargetFilename condition="end
with">.medal</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.mtk118</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.needdecrypt</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.needkeys</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.NIGGA</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.nWcrypt</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.paycyka</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.pizdec</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.pscrypt</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.ReaGAN</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.rumblegoodboy</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.s1crypt</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.scorp</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.sea</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.skunk</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.Trump</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.UNLIS</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                  <TargetFilename condition="end
with">.vdul</TargetFilename> <!--
https://blog.malwarebytes.com/detections/ransome-globeimposter -->
```

```
                <TargetFilename condition="end with">.wallet</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="contains">.write_</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.YAYA</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.zuzya</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">..doc</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->
                <TargetFilename condition="end with">.xlsm</TargetFilename> <!-- https://blog.malwarebytes.com/detections/ransome-globeimposter -->

                <TargetFilename condition="end with">.cerber3</TargetFilename> <!-- Reverse.it sample -->
                <TargetFilename condition="end with">.coded</TargetFilename> <!-- Reverse.it sample -->
                <TargetFilename condition="end with">.STN</TargetFilename> <!-- Reverse.it sample -->
                <TargetFilename condition="end with">.AK47</TargetFilename> <!-- Reverse.it sample -->
                <TargetFilename condition="end with">.asasin</TargetFilename> <!-- Reverse.it sample -->
                <TargetFilename condition="end with">how_to_back_files.html</TargetFilename> <!-- Reverse.it sample -->
                <TargetFilename condition="contains">RECOVER-FILES</TargetFilename> <!-- Reverse.it sample -->
                <TargetFilename condition="contains">DECRYPT</TargetFilename> <!-- Commonly seen in Ransomware -->
                <!-- TEST -->
        </FileCreate>
        <FileCreate onmatch="exclude">
                <!--SECTION: Microsoft:Office:Click2Run-->
                <Image condition="is">C:\Program Files\Common Files\Microsoft Shared\ClickToRun\OfficeC2RClient.exe</Image> <!-- Microsoft:Office Click2Run-->
                <!--SECTION: Microsoft:Windows-->
                <Image condition="is">C:\Windows\System32\smss.exe</Image> <!-- Microsoft:Windows: Session Manager SubSystem: Creates swapfile.sys,pagefile.sys,hiberfile.sys-->
                <Image condition="is">C:\Windows\system32\CompatTelRunner.exe</Image> <!-- Microsoft:Windows: Windows 10 app, creates tons of cache files-->
                <Image condition="is">\\?\C:\Windows\system32\wbem\WMIADAP.EXE</Image> <!-- Microsoft:Windows: WMI Performance updates-->
                <TargetFilename condition="begin with">C:\Windows\System32\DriverStore\Temp\</TargetFilename> <!-- Microsoft:Windows: Temp files by DrvInst.exe-->
```

```
            <TargetFilename condition="begin with">C:\Windows\System32\wbem\Performance\</TargetFilename> <!--Microsoft:Windows: Created in wbem by WMIADAP.exe-->
            <TargetFilename condition="end with">WRITABLE.TST</TargetFilename> <!-- Microsoft:Windows: Created in wbem by svchost-->
            <!--SECTION: Microsoft:Windows:Updates-->
            <TargetFilename condition="begin with">C:\$WINDOWS.~BT\Sources\SafeOS\SafeOS.Mount\</TargetFilename> <!--Microsoft:Windows: Feature updates containing lots of .exe and .sys-->
            <Image condition="begin with">C:\WINDOWS\winsxs\amd64_microsoft-windows</Image> <!--Microsoft:Windows: Windows update-->
            <!--SECTION: Dell-->
            <Image condition="is">C:\Program Files (x86)\Dell\CommandUpdate\InvColPC.exe</Image>
            <!--SECTION: Intel-->
            <Image condition="is">C:\Windows\system32\igfxCUIService.exe</Image> <!--Intel: Drops bat and other files in \Windows in normal operation-->
        </FileCreate>

    <!--SYSMON EVENT ID 12 & 13 & 14 : REGISTRY MODIFICATION-->
        <!--NOTE:   It may appear this section is missing important entries, but many of them match multiple areas, so look carefully to see if something is already covered.-->
        <!--NOTE:   "contains" conditions below are formatted to reduce CPU load, so they may appear written inconsistently, but this is on purpose from tuning.-->
        <!--NOTE:   "contains" works by finding the first letter, then matching the second, etc, so the first letters should be as low-occurance as possible.-->
        <!--NOTE:   Windows writes hundreds or thousands of registry keys a minute, so just because you're not changing stuff, doesn't mean these rules aren't being run.-->
        <!--NOTE:   You don't have to spend a lot of time worrying about this, CPUs are fast, but it's something to consider. Every rule and condition type has a cost.-->
        <!--DATA: EventType, UtcTime, ProcessGuid, ProcessId, Image, TargetObject, Details, NewName-->
        <!--TECHNICAL:    Possible prefixes are HKLM, HKCR, and HKEY_USERS-->
        <!--CRITICAL:     Schema version 3.30 and higher use HKLM and HKEY_USERS and HKCR and CurrentControlSet instead of REGISTRY\MACHINE\ and \REGISTRY\USER\ and ControlSet001-->

        <RegistryEvent onmatch="include">
            <!--Autorun or Startups-->
                <!--ADDITIONAL REFERENCE: [ http://www.ghacks.net/2016/06/04/windows-automatic-startup-locations/ ] -->
                <!--ADDITIONAL REFERENCE: [ https://view.officeapps.live.com/op/view.aspx?src=https://arsenalrecon.com/downloads/resources/Registry_Keys_Related_to_Autorun.ods ] -->
                <!--ADDITIONAL REFERENCE: [ http://www.silentrunners.org/launchpoints.html ] -->
                <!-- TEST -->

                <TargetObject condition="contains">Trigger</TargetObject> <!--Look for Trigger -->
                <TargetObject condition="contains">\CurrentVersion\Run</TargetObject> <!--Microsoft:Windows: Run keys, incld RunOnce, RunOnceEx, RunServices, RunServicesOnce [Also covers terminal server] -->
                <TargetObject condition="contains">\Group Policy\Scripts</TargetObject> <!--Microsoft:Windows: Group policy scripts-->
                <TargetObject condition="contains">\Windows\System\Scripts</TargetObject> <!--Microsoft:Windows: Logon, Loggoff, Shutdown-->
                <TargetObject condition="contains">\Policies\Explorer\Run</TargetObject> <!--Microsoft:Windows | Credit @ion-storm-->
                <TargetObject condition="end with">\ServiceDll</TargetObject> <!--Microsoft:Windows: Points to a service's DLL [ https://blog.cylance.com/windows-registry-persistence-part-1-introduction-attack-phases-and-windows-services ] -->
                <TargetObject condition="end with">\ImagePath</TargetObject> <!--Microsoft:Windows: Points to a service's EXE [ https://github.com/crypsisgroup/Splunkmon/blob/master/sysmon.cfg ] -->
                <TargetObject condition="end with">\Start</TargetObject> <!--Microsoft:Windows: Services start mode changes (Disabled, Automatically, Manual)-->
                <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify\</TargetObject> <!--Microsoft:Windows: Autorun location [ https://www.cylance.com/windows-registry-persistence-part-2-the-run-keys-and-search-order ] -->
                <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit\</TargetObject> <!--Microsoft:Windows: Autorun location [ https://www.cylance.com/windows-registry-persistence-part-2-the-run-keys-and-search-order ] -->
                <TargetObject condition="begin with">HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows NT\CurrentVersion\Drivers32</TargetObject> <!--Microsoft:Windows: Legacy driver loading | Credit @ion-storm -->
                <TargetObject condition="begin with">HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\BootExecute</TargetObject> <!--Microsoft:Windows: Autorun | Credit @ion-storm | [ https://www.cylance.com/windows-registry-persistence-part-2-the-run-keys-and-search-order ] -->
                <!--CLSID launch commands and file association changes-->
                <TargetObject condition="contains">\Explorer\FileExts\</TargetObject> <!--Microsoft:Windows: Changes to file extension mapping-->
                <TargetObject condition="contains">\shell\install\command\</TargetObject> <!--Microsoft:Windows: Sensitive subkey under file associations and CLSID that map to launch command-->
                <TargetObject condition="contains">\shell\open\command\</TargetObject> <!--Microsoft:Windows: Sensitive subkey under file associations and CLSID that map to launch command-->
                <TargetObject condition="contains">\shell\open\ddeexec\</TargetObject> <!--Microsoft:Windows: Sensitive subkey under file associations and CLSID that map to launch command-->
                <!--Windows COM-->
                <TargetObject condition="end with">\InprocServer32\(Default)</TargetObject> <!--Microsoft:Windows:COM Object Hijacking [ https://blog.gdatasoftware.com/2014/10/23941-com-object-hijacking-the-discreet-way-of-persistence ] | Credit @ion-storm -->
                <!--Windows shell hijack-->
                <TargetObject condition="contains">\Classes\*\</TargetObject> <!--Microsoft:Windows:Explorer: [ http://www.silentrunners.org/launchpoints.html ] -->
                <TargetObject condition="contains">\Classes\AllFilesystemObjects\</TargetObject> <!--Microsoft:Windows:Explorer: [ http://www.silentrunners.org/launchpoints.html ] -->
                <TargetObject condition="contains">\Classes\Directory\</TargetObject> <!--Microsoft:Windows:Explorer: [ https://stackoverflow.com/questions/1323663/windows-shell-context-menu-option ] -->
                <TargetObject condition="contains">\Classes\Drive\</TargetObject> <!--Microsoft:Windows:Explorer: [ https://stackoverflow.com/questions/1323663/windows-shell-context-menu-option ] -->
                <TargetObject condition="contains">\Classes\Folder\</TargetObject> <!--Microsoft:Windows:Explorer: ContextMenuHandlers, DragDropHandlers, CopyHookHandlers, [ https://stackoverflow.com/questions/1323663/windows-shell-context-menu-option ] -->
                <TargetObject condition="contains">\ContextMenuHandlers\</TargetObject> <!--Microsoft:Windows: [ http://oalabs.openanalysis.net/2015/06/04/malware-persistence-hkey_current_user-shell-extension-handlers/ ] -->
                <TargetObject condition="contains">\CurrentVersion\Shell</TargetObject> <!--Microsoft:Windows: Shell Folders, ShellExecuteHooks, ShellIconOverloadIdentifers, ShellServiceObjects, ShellServiceObjectDelayLoad [ http://oalabs.openanalysis.net/2015/06/04/malware-persistence-hkey_current_user-shell-extension-handlers/ ] -->
                <TargetObject condition="begin with">HKLM\Software\Microsoft\Windows\CurrentVersion\explorer\ShellExecuteHooks</TargetObject> <!--Microsoft:Windows: ShellExecuteHooks-->
                <TargetObject condition="begin with">HKLM\Software\Microsoft\Windows\CurrentVersion\explorer\ShellServiceObjectDelayLoad</TargetObject> <!--Microsoft:Windows: ShellExecuteHooks-->
                <!--AppPaths hijacking-->
                <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\</TargetObject> <!--Microsoft:Windows: Credit to @Hexacorn [ http://www.hexacorn.com/blog/2013/01/19/beyond-good-ol-run-key-part-3/ ] -->
```

```
                  <TargetObject condition="begin with">HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\AppCertDlls\</TargetObject> <!--Microsoft:Windows: Credit to @Hexacorn [ http://www.hexacorn.com/blog/2013/01/19/beyond-good-ol-run-key-part-3/ ] -->
                  <!--Terminal service boobytraps-->
                  <TargetObject condition="begin with">HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp\InitialProgram</TargetObject>
                  <!--Group Policy interity-->
                  <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GPExtensions\</TargetObject> <!--Microsoft:Windows: Group Policy internally uses a plugin architecture that nothing should be modifying-->
                  <!--Winsock and Winsock2-->
                  <TargetObject condition="begin with">HKLM\SYSTEM\CurrentControlSet\Services\WinSock\</TargetObject> <!--Microsoft:Windows: Wildcard, includes Winsock and Winsock2-->
                  <TargetObject condition="end with">\ProxyServer</TargetObject> <!--Microsoft:Windows: System and user proxy server-->
                  <!--Credential providers-->
                  <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\Credential Provider</TargetObject> <!--Wildcard, includes Credental Providers and Credential Provider Filters-->
                  <TargetObject condition="begin with">HKLM\SYSTEM\CurrentControlSet\Control\Lsa\</TargetObject>
                  <TargetObject condition="begin with">HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SecurityProviders</TargetObject> <!--Microsoft:Windows: Changes to WDigest-UseLogonCredential for password scraping [ https://www.trustedsec.com/april-2015/dumping-wdigest-creds-with-meterpreter-mimikatzkiwi-in-windows-8-1/ ] -->
                  <!--Networking-->
                  <TargetObject condition="begin with">HKLM\SYSTEM\CurrentControlSet\Control\NetworkProvider\Order\</TargetObject> <!--Microsoft:Windows: Order of network providers that are checked to connect to destination [ https://www.malwarearchaeology.com/cheat-sheets ] -->
                  <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles</TargetObject> <!--Microsoft:Windows: | Credit @ion-storm -->
                  <!--DLLs that get injected into every process launch-->
                  <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\Appinit_Dlls\</TargetObject> <!--Microsoft:Windows: [ https://msdn.microsoft.com/en-us/library/windows/desktop/dd744762(v=vs.85).aspx ] -->
                  <TargetObject condition="begin with">HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Windows\Appinit_Dlls\</TargetObject> <!--Microsoft:Windows: [ https://msdn.microsoft.com/en-us/library/windows/desktop/dn280412(v=vs.85).aspx ] -->
                  <!--Office-->
```

```xml
        <TargetObject condition="contains">\Microsoft\Office\Outlook\Addins\</TargetObject> <!--Microsoft:Office: Outlook add-ins-->
        <!--IE-->
        <TargetObject condition="contains">\Internet Explorer\Toolbar\</TargetObject> <!--Microsoft:InternetExplorer: Machine and user-->
        <TargetObject condition="contains">\Internet Explorer\Extensions\</TargetObject> <!--Microsoft:InternetExplorer: Machine and user-->
        <TargetObject condition="contains">\Browser Helper Objects\</TargetObject> <!--Microsoft:InternetExplorer: Machine and user [ https://msdn.microsoft.com/en-us/library/bb250436(v=vs.85).aspx ] -->
        <!--Magic registry keys-->
        <TargetObject condition="contains">{AB8902B4-09CA-4bb6-B78D-A8F59079A8D5}\</TargetObject> <!--Microsoft:Windows: Thumbnail cache autostart [ http://blog.trendmicro.com/trendlabs-security-intelligence/poweliks-levels-up-with-new-autostart-mechanism/ ] -->
        <!--Infection artifacts-->
        <TargetObject condition="end with">\UrlUpdateInfo</TargetObject> <!--Microsoft:ClickOnce: [ https://subt0x10.blogspot.com/2016/12/mimikatz-delivery-via-clickonce-with.html ] -->
        <TargetObject condition="end with">\InstallSource</TargetObject> <!--Microsoft:Windows: Source folder for certain program and componenent installations-->
        <!--Windows UAC tampering-->
        <TargetObject condition="begin with">HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\EnableLUA</TargetObject> <!--Detect: UAC Tampering | Credit @ion-storm -->
        <TargetObject condition="begin with">HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\LocalAccountTokenFilterPolicy</TargetObject> <!--Detect: UAC Tampering | Credit @ion-storm -->
        <!--Microsoft Firewall modifications-->
        <TargetObject condition="begin with">HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\AuthorizedApplications\List</TargetObject> <!--Windows Firewall authorized applications | Credit @ion-storm -->
        <!--Microsoft Security Center tampering | Credit @ion-storm -->
        <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Security Center\AllAlertsDisabled</TargetObject>
        <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Security Center\AntiVirusDisableNotify</TargetObject>
        <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Security Center\DisableMonitoring</TargetObject>
        <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Security Center\FirewallDisableNotify</TargetObject>
        <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Security Center\FirewallOverride</TargetObject>
```

```
                <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Security Center\UacDisableNotify</TargetObject>
                <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Security Center\UpdatesDisableNotify</TargetObject>
                <!--Windows Defender tampering | Credit @ion-storm -->
                <TargetObject condition="begin with">HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\DisableAntiSpyware</TargetObject>
                <TargetObject condition="begin with">HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\DisableAntiVirus</TargetObject>
                <TargetObject condition="begin with">HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection\DisableBehaviorMonitoring</TargetObject>
                <TargetObject condition="begin with">HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection\DisableOnAccessProtection</TargetObject>
                <TargetObject condition="begin with">HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection\DisableScanOnRealtimeEnable</TargetObject>
                <TargetObject condition="begin with">HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet\SpyNetReporting</TargetObject>
                <!--Windows internals integrity monitoring-->
                <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\</TargetObject> <!--Microsoft:Windows: Malware likes changing IFEO-->
                <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\</TargetObject> <!--Microsoft:Windows:UAC: Detect malware changes to UAC prompt level-->
                <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\</TargetObject> <!--Microsoft:Windows: Event log system integrity and ACLs-->
                <TargetObject condition="begin with">HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\</TargetObject> <!--Microsoft:Defender: Detect changes to Defender administrative settings to monitor for disablement-->
                <TargetObject condition="begin with">HKLM\SYSTEM\CurrentControlSet\Control\Safeboot\</TargetObject> <!--Microsoft:Windows: Services approved to load in safe mode-->
                <TargetObject condition="begin with">HKLM\SYSTEM\CurrentControlSet\Control\Winlogon\</TargetObject> <!--Microsoft:Windows: Providers notified by WinLogon-->
                <TargetObject condition="end with">\FriendlyName</TargetObject> <!--Microsoft:Windows: New devices connected and remembered-->
                <TargetObject condition="is">HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\InProgress\(Default)</TargetObject> <!--Microsoft:Windows: See when WindowsInstaller is engaged-->
        </RegistryEvent>
        <RegistryEvent onmatch="exclude">
        <!--COMMENT:      Remove low-information noise-->
```

```
            <!--SECTION: Microsoft binaries-->
            <Image condition="end with">Office\root\integration\integrator.exe</Image> <!--Microsoft:Office: C2R client-->
            <Image condition="image">C:\WINDOWS\system32\backgroundTaskHost.exe</Image> <!--Microsoft:Windows: Changes association registry keys-->
            <Image condition="is">C:\Program Files\Common Files\Microsoft Shared\ClickToRun\OfficeClickToRun.exe</Image> <!--Microsoft:Office: C2R client-->
            <Image condition="is">C:\Program Files\Windows Defender\MsMpEng.exe</Image> <!--Microsoft:Windows:Defender-->
            <Image condition="is">C:\Windows\SystemApps\Microsoft.Windows.Cortana_cw5n1h2txyewy\SearchUI.exe</Image> <!--Microsoft:Cortana-->
            <!--Misc-->
            <TargetObject condition="end with">Toolbar\WebBrowser</TargetObject> <!--Microsoft:IE: Extraneous activity-->
            <TargetObject condition="end with">Toolbar\WebBrowser\ITBar7Height</TargetObject> <!--Microsoft:IE: Extraneous activity-->
            <TargetObject condition="end with">Toolbar\ShellBrowser\ITBar7Layout</TargetObject> <!--Microsoft:Windows:Explorer: Extraneous activity-->
            <TargetObject condition="end with">Internet Explorer\Toolbar\Locked</TargetObject> <!--Microsoft:Windows:Explorer: Extraneous activity-->
            <TargetObject condition="end with">ShellBrowser</TargetObject> <!--Microsoft:InternetExplorer: Noise-->
            <TargetObject condition="end with">\CurrentVersion\Run</TargetObject> <!--Microsoft:Windows: Remove noise from the "\Windows\CurrentVersion\Run" wildcard-->
            <TargetObject condition="end with">\CurrentVersion\RunOnce</TargetObject> <!--Microsoft:Windows: Remove noise from the "\Windows\CurrentVersion\Run" wildcard-->
            <TargetObject condition="end with">\CurrentVersion\App Paths</TargetObject> <!--Microsoft:Windows: Remove noise from the "\Windows\CurrentVersion\App Paths" wildcard-->
            <TargetObject condition="end with">\CurrentVersion\Image File Execution Options</TargetObject> <!--Microsoft:Windows: Remove noise from the "\Windows\CurrentVersion\Image File Execution Options" wildcard-->
            <TargetObject condition="end with">\CurrentVersion\Shell Extensions\Cached</TargetObject> <!--Microsoft:Windows: Remove noise from the "\CurrentVersion\Shell Extensions\Cached" wildcard-->
            <TargetObject condition="end with">\CurrentVersion\Shell Extensions\Approved</TargetObject> <!--Microsoft:Windows: Remove noise from the "\CurrentVersion\Shell Extensions\Approved" wildcard-->
            <TargetObject condition="end with">}\PreviousPolicyAreas</TargetObject> <!--Microsoft:Windows: Remove noise from \Winlogon\GPExtensions by svchost.exe-->
            <TargetObject condition="contains">\Control\WMI\Autologger\</TargetObject> <!--Microsoft:Windows: Remove noise from monitoring "\Start"-->
            <TargetObject condition="end with">HKLM\SYSTEM\CurrentControlSet\Services\UsoSvc\Start</TargetObject> <!--Microsoft:Windows: Remove noise from monitoring "\Start"-->
```

```
                <TargetObject condition="end with">\Lsa\OfflineJoin\CurrentValue</TargetObject> <!--Microsoft:Windows: Sensitive value during domain join-->
                <TargetObject condition="end with">\Components\TrustedInstaller\Events</TargetObject> <!--Microsoft:Windows: Remove noise monitoring Winlogon-->
                <TargetObject condition="end with">\Components\TrustedInstaller</TargetObject> <!--Microsoft:Windows: Remove noise monitoring Winlogon-->
                <TargetObject condition="end with">\Components\Wlansvc</TargetObject> <!--Microsoft:Windows: Remove noise monitoring Winlogon-->
                <TargetObject condition="end with">\Components\Wlansvc\Events</TargetObject> <!--Microsoft:Windows: Remove noise monitoring Winlogon-->
                <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\</TargetObject> <!--Microsoft:Windows: Remove noise monitoring installations run as system-->
                <TargetObject condition="end with">\Directory\shellex</TargetObject> <!--Microsoft:Windows: Remove noise monitoring Classes-->
                <TargetObject condition="end with">\Directory\shellex\DragDropHandlers</TargetObject> <!--Microsoft:Windows: Remove noise monitoring Classes-->
                <TargetObject condition="end with">\Drive\shellex</TargetObject> <!--Microsoft:Windows: Remove noise monitoring Classes-->
                <TargetObject condition="end with">\Drive\shellex\DragDropHandlers</TargetObject> <!--Microsoft:Windows: Remove noise monitoring Classes-->
                <TargetObject condition="contains">_Classes\AppX</TargetObject> <!--Microsoft:Windows: Remove noise monitoring "Shell\open\command"--> <!--Win8+-->
                <TargetObject condition="begin with">HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Publishers\</TargetObject> <!--Microsoft:Windows: SvcHost Noise-->
                <Image condition="is">C:\Windows\SystemApps\Microsoft.Windows.Cortana_cw5n1h2txyewy\SearchUI.exe</Image> <!--Microsoft:Windows: Remove noise from Windows 10 Cortana | Credit @ion-storm--> <!--Win10-->
                <!--Bootup Control noise-->
                <TargetObject condition="end with">HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Audit</TargetObject> <!--Microsoft:Windows:lsass.exe: Boot noise--> <!--Win8+-->
                <TargetObject condition="end with">HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Audit\AuditPolicy</TargetObject> <!--Microsoft:Windows:lsass.exe: Boot noise--> <!--Win8+-->
                <TargetObject condition="end with">HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Audit\PerUserAuditing\System</TargetObject> <!--Microsoft:Windows:lsass.exe: Boot noise--> <!--Win8+-->
                <TargetObject condition="end with">HKLM\SYSTEM\CurrentControlSet\Control\Lsa\SspiCache</TargetObject> <!--Microsoft:Windows:lsass.exe: Boot noise--> <!--Win8+-->
                <TargetObject condition="end with">HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Domains</TargetObject> <!--Microsoft:Windows:lsass.exe: Boot noise--> <!--Win8+-->
                <TargetObject condition="end with">HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\Audit</TargetObject> <!--Microsoft:Windows:lsass.exe: Boot noise--> <!--Win8+-->
                <!--Sevices autostart noise-->
                <TargetObject condition="end with">\services\clr_optimization_v2.0.50727_32\Start</TargetObject> <!--Microsoft:dotNet: Windows 7-->
                <TargetObject condition="end with">\services\clr_optimization_v2.0.50727_64\Start</TargetObject> <!--Microsoft:dotNet: Windows 7-->
                <TargetObject condition="end with">\services\clr_optimization_v4.0.30319_32\Start</TargetObject> <!--Microsoft:dotNet: Windows 10-->
                <TargetObject condition="end with">\services\clr_optimization_v4.0.30319_64\Start</TargetObject> <!--Microsoft:dotNet: Windows 10-->
                <TargetObject condition="end with">\services\DeviceAssociationService\Start</TargetObject> <!--Microsoft:Windows: Remove noise from monitoring "\Start"-->
                <TargetObject condition="end with">\services\BITS\Start</TargetObject> <!--Microsoft:Windows: Remove noise from monitoring "\Start"-->
                <TargetObject condition="end with">\services\TrustedInstaller\Start</TargetObject> <!--Microsoft:Windows: Remove noise from monitoring "\Start"-->
                <TargetObject condition="end with">\services\tunnel\Start</TargetObject> <!--Microsoft:Windows: Remove noise from monitoring "\Start"-->
                <TargetObject condition="end with">\services\UsoSvc\Start</TargetObject> <!--Microsoft:Windows: Remove noise from monitoring "\Start"-->
                <!--FileExts noise filtering-->
                <TargetObject condition="contains">\OpenWithProgids</TargetObject> <!--Microsoft:Windows: Remove noise from monitoring "FileExts"-->
                <TargetObject condition="end with">\OpenWithList</TargetObject> <!--Microsoft:Windows: Remove noise from monitoring "FileExts"-->
                <TargetObject condition="end with">\UserChoice</TargetObject> <!--Microsoft:Windows: Remove noise from monitoring "FileExts"-->
                <TargetObject condition="end with">\UserChoice\ProgId</TargetObject> <!--Microsoft:Windows: Remove noise from monitoring "FileExts"--> <!--Win8+-->
                <TargetObject condition="end with">\UserChoice\Hash</TargetObject> <!--Microsoft:Windows: Remove noise from monitoring "FileExts"--> <!--Win8+-->
                <TargetObject condition="end with">\OpenWithList\MRUList</TargetObject> <!--Microsoft:Windows: Remove noise from monitoring "FileExts"-->
                <TargetObject condition="end with">} 0xFFFF</TargetObject> <!--Microsoft:Windows: Remove noise from explorer.exe from monitoring ShellCached binary keys--> <!--Win8+-->
                <!--SECTION: 3rd party-->
```

```
			<Image condition="is">C:\Program Files\WIDCOMM\Bluetooth Software\btwdins.exe</Image> <!--Constantly writes to HKLM-->
			<Image condition="is">C:\Program Files (x86)\Webroot\WRSA.exe</Image> <!--Webroot-->
		</RegistryEvent>

	<!--SYSMON EVENT ID 15 : ALTERNATE DATA STREAM CREATED-->
		<!--DATA: UtcTime, ProcessGuid, ProcessId, Image, TargetFilename, CreationUtcTime, Hash-->
		<FileCreateStreamHash onmatch="include">
		<!--COMMENT:	Any files created with an NTFS Alternate Data Stream which match these rules will be hashed and logged.
				[ https://blogs.technet.microsoft.com/askcore/2013/03/24/alternate-data-streams-in-ntfs/ ]
				ADS's are used by browsers and email clients to mark files as originating from the Internet or other foreign sources.
				[ https://textslashplain.com/2016/04/04/downloads-and-the-mark-of-the-web/ ] -->
			<TargetFilename condition="contains">Content.Outlook</TargetFilename> <!--Microsoft:Outlook: Attachments--> <!--PRIVACY WARNING-->
			<TargetFilename condition="contains">Downloads</TargetFilename> <!--Downloaded files. Does not include "Run" files in IE-->
			<TargetFilename condition="contains">Temp\7z</TargetFilename> <!--7zip extractions-->
			<TargetFilename condition="end with">.bat</TargetFilename> <!--Batch scripting-->
			<TargetFilename condition="end with">.cmd</TargetFilename> <!--Batch scripting | Credit @ion-storm -->
			<TargetFilename condition="end with">.hta</TargetFilename> <!--Scripting-->
			<TargetFilename condition="end with">.lnk</TargetFilename> <!--Shortcut file | Credit @ion-storm -->
			<TargetFilename condition="end with">.ps1</TargetFilename> <!--Powershell-->
			<TargetFilename condition="end with">.ps2</TargetFilename> <!--Powershell-->
			<TargetFilename condition="end with">.reg</TargetFilename> <!--Registry File-->
			<TargetFilename condition="end with">.vb</TargetFilename> <!--VisualBasicScripting files-->
			<TargetFilename condition="end with">.vbe</TargetFilename> <!--VisualBasicScripting files-->
			<TargetFilename condition="end with">.vbs</TargetFilename> <!--VisualBasicScripting files-->
		</FileCreateStreamHash>

	<!--SYSMON EVENT ID 16 : SYSMON CONFIGURATION CHANGE, THIS LINE IS INCLUDED FOR DOCUMENTATION PURPOSES ONLY-->
		<!--DATA: UtcTime, Configuration, ConfigurationFileHash-->
		<!--Cannot be filtered.-->

	<!--SYSMON EVENT ID 17 & 18 : PIPE CREATED / PIPE CONNECTED-->
		<!--DATA: UtcTime, ProcessGuid, ProcessId, PipeName, Image-->

		<PipeEvent onmatch="include">
		<!-- TESTING -->
			<!--ADDITIONAL REFERENCE: [ https://www.cobaltstrike.com/help-smb-beacon ] -->
			<!--ADDITIONAL REFERENCE: [ https://blog.cobaltstrike.com/2015/10/07/named-pipe-pivoting/ ] -->
			<PipeName condition="end with">trigger</PipeName> <!-- Look for Trigger -->
		</PipeEvent>

	<!-- TESTING -->
	<!--SYSMON EVENT ID 19: WMI_FILTER-->
		<!--DATA: EventType, UtcTime, Operation, User, EventNamespace, Name, Query-->
	<!--SYSMON EVENT ID 20: WMI_CONSUMER-->
		<!--DATA: EventType, UtcTime, Operation, User, Name, Type, Destination -->
	<!--SYSMON EVENT ID 21: WMI_BINDING-->
		<!--DATA: EventType, UtcTime, Operation, User, Consumer, Filter -->
		<WmiEvent onmatch="exclude">
			<!-- Nothing is excluded, so trigger will fire -->
		</WmiEvent>

	</EventFiltering>
</Sysmon>
```

APPENDIX C: TRIGGER TOOL PYTHON SCRIPT (TRIGGER.PY)

The Python script for the trigger tool created in this research can be found online at `https://github.com/dsugraduate/dsu2018/`. The Python script may be used to trigger all 21 Sysmon events listed in Appendix A. The script must be run using Python version 3 with Administrator privileges. The script is included below for convenience, with line numbers displayed for reference.

```
1   import ctypes
2   import inspect
3   import os
4   import os.path
5   import shutil
6   import struct
7   import subprocess
8   import sys
9   import tempfile
10  import threading
11  import time
12  import winreg
13
14  ##
15  ## Function Name:
16  ##   has_admin()
17  ##
18  ## Purpose:
19  ##   Checks if user has admin privileges
20  ##
21  ## References:
22  ##   https://stackoverflow.com/questions/2946746/python-checking-if-a-user-has-administrator-privileges
23  ##
24  def has_admin():
25      if os.name == 'nt':
26          try:
27              # only windows users with admin privileges can read the C:\windows\temp
28              temp = os.listdir(os.sep.join([os.environ.get('SystemRoot','C:\\windows'),'temp']))
29          except:
30              return (os.environ['USERNAME'],False)
31          else:
32              return (os.environ['USERNAME'],True)
33      else:
34          if 'SUDO_USER' in os.environ and os.geteuid() == 0:
35              return (os.environ['SUDO_USER'],True)
```

```python
        else:
            return (os.environ['USERNAME'],False)

##
## Function Name:
##   trigger1()
##
## Purpose:
##   Triggers Sysmon Event ID 1
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 1: Process creation
##     The process creation event provides extended information about a newly created process.
##     The full command line provides context on the process execution.
##     The ProcessGUID field is a unique value for this process across a domain to make event correlation easier.
##     The hash is a full hash of the file with the algorithms in the HashType field.
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <ProcessCreate onmatch="exclude">
##      <!--COMMENT:      All process launched will be included, except for what matches a rule below... Make sure you don't have any rules that exclude cmd.exe -->
##   </ProcessCreate>
##
## References:
##   https://docs.python.org/3/library/subprocess.html#subprocess.TimeoutExpired
##   https://stackoverflow.com/questions/847850/cross-platform-way-of-getting-temp-directory-in-python
##
def trigger1():
    tempdir = tempfile.gettempdir()
    trigger = tempdir + "\\1trigger.exe"
    result = shutil.copyfile("C:\\Windows\\System32\\cmd.exe", trigger)
    time.sleep(1)
    cmd = [trigger]
    try:
        process = subprocess.run(cmd, timeout=5)
    except subprocess.TimeoutExpired:
        pass
    except:
        print("EventID 1: ERROR")
        return
    result = os.remove(trigger)
    print("EventID 1: Triggered")
    return

##
## Function Name:
##   trigger2()
##
```

```python
## Purpose:
##   Triggers Sysmon Event ID 2
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 2: A process changed a file creation time
##     The change file creation time event is registered when a file creation time is explicitly modified by a process.
##     This event helps tracking the real creation time of a file.
##     Attackers may change the file creation time of a backdoor to make it look like it was installed with the operating system.
##     Note that many processes legitimately change the creation time of a file; it does not necessarily indicate malicious activity.
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <FileCreateTime onmatch="include">
##     <TargetFilename condition="end with">trigger.txt</TargetFilename> <!--Look for Trigger file -->
##   </FileCreateTime>
##
## References:
##   https://technologytales.com/2014/10/29/changing-file-timestamps-using-windows-powershell/
##   https://stackoverflow.com/questions/847850/cross-platform-way-of-getting-temp-directory-in-python
##
def trigger2():
    tempdir = tempfile.gettempdir()
    f = open(tempdir + "\\2trigger.txt","w+")

    cmd1 = ["powershell.exe", "$(Get-Item " + f.name + ").creationtime"]
    cmd2 = ["powershell.exe", "$(Get-Item " + f.name + ").creationtime=$(Get-Date '1/1/1950')"]
    cmd3 = ["powershell.exe", "$(Get-Item " + f.name + ").creationtime"]
    try:
        ret = subprocess.run(cmd1, timeout=20, stdout=subprocess.PIPE, stderr=subprocess.STDOUT)
        ret = subprocess.run(cmd2, timeout=20, stdout=subprocess.PIPE, stderr=subprocess.STDOUT)
        ret = subprocess.run(cmd3, timeout=20, stdout=subprocess.PIPE, stderr=subprocess.STDOUT)
    except subprocess.TimeoutExpired:
        print("EventID 2: TIMEOUT")
        return
    finally:
        f.close()
        result = os.remove(f.name)

    print("EventID 2: Triggered")
    return

##
## Function Name:
```

```
##   trigger3()
##
## Purpose:
##   Triggers Sysmon Event ID 3
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 3: Network connection
##     The network connection event logs TCP/UDP connections on the machine. It is disabled by default.
##     Each connection is linked to a process through the ProcessId and ProcessGUID fields.
##     The event also contains the source and destination host names IP addresses, port numbers and IPv6 status.
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <NetworkConnect onmatch="include">
##     <Image condition="image">powershell.exe</Image> <!--Microsoft:Windows: PowerShell interface-->
##   </NetworkConnect>
##
## References:
##   https://learn-powershell.net/2011/02/11/using-powershell-to-query-web-site-information/
##
def trigger3():
    cmd = ["powershell.exe", "$wc = New-Object system.Net.WebClient; $wc.Headers.Add('User-Agent','Mozilla/5.0 (Windows NT 6.2; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36'); $wp = $wc.downloadString('http://dsu.edu'); exit;"]
    try:
        ret = subprocess.run(cmd, timeout=120)
    except subprocess.TimeoutExpired:
        print("EventID 3: TIMEOUT")
        return
    print("EventID 3: Triggered")
    return

##
## Function Name:
##   trigger4()
##
## Purpose:
##   Triggers Sysmon Event ID 4
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 4: Sysmon service state changed
##     The service state change event reports the state of the Sysmon service (started or stopped).
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <!-- This event cannot be filtered. -->
```

```
##
## References:
##   https://docs.python.org/3/library/subprocess.html
##
def trigger4():
    mycwd = os.getcwd()
    cmd1 = ["powershell.exe", "Restart-Service", "sysmon"]
    try:
        ret = subprocess.run(cmd1, timeout=60, cwd=mycwd, shell=False)
    except subprocess.TimeoutExpired:
        print("EventID 4: TIMEOUT")
        return

    print("EventID 4: Triggered")
    return

##
## Function Name:
##   trigger5()
##
## Purpose:
##   Triggers Sysmon Event ID 5
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 5: Process terminated
##     The process terminate event reports when a process terminates. It provides the UtcTime, ProcessGuid and ProcessId of the process.
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <ProcessTerminate onmatch="include">
##     <Image condition="begin with">C:\Users</Image> <!--Process terminations by user binaries-->
## OR
##     <Image condition="end with">trigger.exe</Image> <!-- Look for Trigger -->
##   </ProcessTerminate>
##
## References:
##   https://docs.python.org/3/library/subprocess.html
##  https://stackoverflow.com/questions/847850/cross-platform-way-of-getting-temp-directory-in-python
##
def trigger5():
    tempdir = tempfile.gettempdir()
    trigger = tempdir + "\\5trigger.exe"
    result = shutil.copyfile("C:\\Windows\\System32\\calc.exe", trigger)
    time.sleep(2)
    cmd = [trigger]
    try:
        ret = subprocess.run(cmd, timeout=1)
    except subprocess.TimeoutExpired:
        pass

    result = os.remove(trigger)
    time.sleep(2)
    print("EventID 5: Triggered")
    return

##
## Function Name:
##   trigger6()
##
## Purpose:
##   Triggers Sysmon Event ID 6
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 6: Driver loaded
##     The driver loaded events provides information about a driver being loaded on the system.
##     The configured hashes are provided as well as signature information.
##     The signature is created asynchronously for performance reasons and indicates if the file was removed after loading.
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <DriverLoad onmatch="exclude">
##     <!--COMMENT: Because drivers with bugs can be used to escalate to kernel permissions, be extremely selective
##                  about what you exclude from monitoring. Low event volume, little incentive to exclude.-->
##     <Signature condition="contains">microsoft</Signature> <!--Exclude signed Microsoft drivers-->
##     <Signature condition="contains">windows</Signature> <!--Exclude signed Microsoft drivers-->
##     <Signature condition="begin with">Intel </Signature> <!--Exclude signed Intel drivers-->
##   </DriverLoad>
##
## References:
##   https://docs.microsoft.com/en-us/sysinternals/downloads/procmon
##   https://docs.python.org/3/library/subprocess.html
##
def trigger6():

    mycwd = os.getcwd()
    cmd1 = [mycwd + "\\ProcessMonitor\\Procmon.exe", "/AcceptEula", "/Minimized", "/Runtime", "3"]
    try:
        ret = subprocess.run(cmd1, timeout=30, cwd=mycwd, shell=False)
    except subprocess.TimeoutExpired:
        print("EventID 6: TIMEOUT")
        return

    print("EventID 6: Triggered")
    return

##
```

```
## Function Name:
##   trigger7()
##
## Purpose:
##   Triggers Sysmon Event ID 7
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 7: Image loaded
##     The image loaded event logs when a module is loaded in a specific process.
##     This event is disabled by default and needs to be configured with the –l option. It indicates the process in which the module is loaded, hashes and signature information.
##     The signature is created asynchronously for performance reasons and indicates if the file was removed after loading.
##     This event should be configured carefully, as monitoring all image load events will generate a large number of events.
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <ImageLoad onmatch="include">
##     <ImageLoaded condition="end with">trigger.dll</ImageLoaded> <!-- Look for Trigger -->
##   </ImageLoad>
##
## References:
##   https://stackoverflow.com/questions/32732751/unload-dll-loaded-in-python
##   https://stackoverflow.com/questions/847850/cross-platform-way-of-getting-temp-directory-in-python
##
def trigger7():
    tempdir = tempfile.gettempdir()
    trigger = tempdir + "\\7trigger.dll"
    result = shutil.copyfile("C:\\Windows\\System32\\user32.dll", trigger)
    time.sleep(2)
    try:
        m = ctypes.cdll.LoadLibrary(trigger)
        _ctypes.FreeLibrary(trigger)
        time.sleep(5)
    except OSError as e:
        # Ignore dll initialization errors... we just want to trigger loading
        if e.winerror == 1114:
            pass
        else:
            print("EventID 7: ERROR", e)
            return
    except:
        print("EventID 7: ERROR", sys.exc_info())
        return

    result = os.remove(trigger)
    print("EventID 7: Triggered")
    return

##
## Function Name:
##   trigger8()
##
## Purpose:
##   Triggers Sysmon Event ID 8
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 8: CreateRemoteThread
##     The CreateRemoteThread event detects when a process creates a thread in another process.
##     This technique is used by malware to inject code and hide in other processes.
##     The event indicates the source and target process.
##     It gives information on the code that will be run in the new thread: StartAddress, StartModule and StartFunction.
##     Note that StartModule and StartFunction fields are inferred, they might be empty if the starting address is outside loaded modules or known exported functions.
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <CreateRemoteThread onmatch="exclude">
##     <!--COMMENT: Monitor for processes injecting code into other processes. Often used by malware to cloak their actions.
##                  Exclude mostly-safe sources and log anything else.-->
##   </CreateRemoteThread>
##
## References:
##   https://github.com/infodox/python-dll-injection/blob/master/dll_inject.py
##   https://stackoverflow.com/questions/7989922/opening-a-process-with-popen-and-getting-the-pid
##   https://www.christophertruncer.com/injecting-shellcode-into-a-remote-process-with-python/
##   https://docs.python.org/3/library/subprocess.html
##  https://stackoverflow.com/questions/847850/cross-platform-way-of-getting-temp-directory-in-python
##
def trigger8():
    tempdir = tempfile.gettempdir()
    trigger = tempdir + "\\8trigger.exe"
    result = shutil.copyfile("C:\\Windows\\System32\\calc.exe", trigger)
    time.sleep(3)

    process = subprocess.Popen(trigger, shell=False)

    page_rwx_value = 0x40
    process_all = 0x1F0FFF
    memcommit = 0x00001000
    kernel32_variable = ctypes.windll.kernel32
    shellcode = "\x90\x90\x90\x90"
    process_id = process.pid
    shellcode_length = len(shellcode)
    process_handle = kernel32_variable.OpenProcess(process_all, False, process_id)
    result = memory_allocation_variable = kernel32_variable.VirtualAllocEx(process_handle, 0, shellcode_length, memcommit, page_rwx_value)
    result = kernel32_variable.WriteProcessMemory(process_handle, memory_allocation_variable, shellcode, shellcode_length, 0)
    result = kernel32_variable.CreateRemoteThread(process_handle, None, 0, memory_allocation_variable, 0, 0, 0)

    result = process.kill()
    time.sleep(4)
    result = os.remove(trigger)
    print("EventID 8: Triggered")
    return

##
## Function Name:
##   trigger9()
##
## Purpose:
##   Triggers Sysmon Event ID 9
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 9: RawAccessRead
##     The RawAccessRead event detects when a process conducts reading operations from the drive using the \\.\ denotation.
##     This technique is often used by malware for data exfiltration of files that are locked for reading, as well as to avoid file access auditing tools.
##     The event indicates the source process and target device.
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <RawAccessRead onmatch="include">
##     <Image condition="end with">powershell.exe</Image>
##     <Image condition="end with">wmic.exe</Image>
##   </RawAccessRead>
##
## References:
##   https://ardamis.com/2012/08/21/getting-a-list-of-logical-and-physical-drives-from-the-command-line/
##   https://docs.python.org/3/library/subprocess.html
##
def trigger9():
    cmd = ["powershell.exe", "wmic.exe", "diskdrive", "list"]
    try:
        result = subprocess.run(cmd, timeout=60)
    except:
        print("EventID 9: ERROR", sys.exc_info())
        return
    time.sleep(5)
    print("EventID 9: Triggered")
    return

##
## Function Name:
##   trigger10()
##
## Purpose:
##   Triggers Sysmon Event ID 10
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 10: ProcessAccess
##     The process accessed event reports when a process opens another process, an operation that’s often followed by information queries or reading and writing the address space of the target process.
##     This enables detection of hacking tools that read the memory contents of processes like Local Security Authority (Lsass.exe) in order to steal credentials for use in Pass-the-Hash attacks.
##     Enabling it can generate significant amounts of logging if there are diagnostic utilities active that repeatedly open processes to query their state, so it generally should only be done so with filters that remove expected accesses.
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <ProcessAccess onmatch="include"> <!--TEST-->
##     <SourceImage condition="end with">trigger.exe</SourceImage> <!-- Look for trigger -->
##   </ProcessAccess>
##
## References:
##   https://docs.python.org/3/library/subprocess.html
##
def trigger10():
    mycwd = os.getcwd()
    tempdir = tempfile.gettempdir()
    trigger = tempdir + "\\10trigger.exe"
    result = shutil.copyfile("C:\\Windows\\System32\\cmd.exe", trigger)
    cmd = [trigger, "/c " + trigger + " /c echo trigger"]
    try:
        result = subprocess.run(cmd, timeout=60, cwd=mycwd, stdout=subprocess.PIPE, shell=False, check=True)
    except:
        print("EventID 10: ERROR", sys.exc_info())
        return
    result = os.remove(trigger)
    print("EventID 10: Triggered")
    return

##
## Function Name:
##   trigger11()
##
## Purpose:
##   Triggers Sysmon Event ID 11
```

```
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 11: FileCreate
##     File create operations are logged when a file is created or overwritten.
##     This event is useful for monitoring autostart locations, like the Startup folder, as well as temporary and download directories, which are common places malware drops during initial infection.
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <FileCreate onmatch="include">
##     <TargetFilename condition="end with">.exe</TargetFilename> <!--Executable-->
##   </FileCreate>
##
## References:
##  https://stackoverflow.com/questions/847850/cross-platform-way-of-getting-temp-directory-in-python
##
def trigger11():
    tempdir = tempfile.gettempdir()
    trigger = tempdir + "\\11trigger.exe"
    result = shutil.copyfile("C:\\Windows\\System32\\CMD.EXE", trigger)
    result = os.remove(trigger)
    print("EventID 11: Triggered")
    return

##
## Function Name:
##   trigger121314()
##
## Purpose:
##   Triggers Sysmon Event ID 12, 13, and 14
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 12: RegistryEvent (Object create and delete)
##     Registry key and value create and delete operations map to this event type, which can be useful for monitoring for changes to Registry autostart locations, or specific malware registry modifications.
##
##   Event ID 13: RegistryEvent (Value Set)
##     This Registry event type identifies Registry value modifications. The event records the value written for Registry values of type DWORD and QWORD.
##
##   Event ID 14: RegistryEvent (Key and Value Rename)
##     Registry key and value rename operations map to this event type, recording the new name of the key or value that was renamed.
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <RegistryEvent onmatch="include">
##     <TargetObject condition="contains">\CurrentVersion\Run</TargetObject>
##   </RegistryEvent>
##
## References:
##   https://stackoverflow.com/questions/37357411/adding-an-exe-file-to-registry-on-windows-run-at-startup-via-python
##   https://stackoverflow.com/questions/2632199/how-do-i-get-the-path-of-the-current-executed-file-in-python
##   http://blogs.microsoft.co.il/pavely/2015/09/29/regrenamekey-hidden-registry-api/
##   https://www.blog.pythonlibrary.org/2010/03/20/pythons- winreg-editing-the-windows-registry/
##
def trigger121314():
    keyVal = r'Software\CurrentVersion\Run'
    try:
        key = winreg.OpenKey(winreg.HKEY_CURRENT_USER, keyVal, 0, winreg.KEY_ALL_ACCESS)
    except:
        key = winreg.CreateKey(winreg.HKEY_CURRENT_USER, keyVal)
    winreg.SetValueEx(key, "Trigger", 0, winreg.REG_SZ, "trigger1213")
    winreg.DeleteValue(key, "Trigger")

    keyVal = r'Software\CurrentVersion\Run\14Trigger'
    try:
        key2 = winreg.OpenKey(winreg.HKEY_CURRENT_USER, keyVal, 0, winreg.KEY_ALL_ACCESS)
    except:
        key2 = winreg.CreateKey(winreg.HKEY_CURRENT_USER, keyVal)

    m = ctypes.cdll.LoadLibrary("c:\\windows\\system32\\advapi32")
    ret = m.RegRenameKey(key2.handle,0,r"New14Trigger")

    winreg.DeleteKey(key, r"New14Trigger")
    winreg.CloseKey(key)
    winreg.CloseKey(key2)

    print("EventID 12: Triggered")
    print("EventID 13: Triggered")
    print("EventID 14: Triggered")
    return

##
## Function Name:
##   trigger15()
##
## Purpose:
##   Triggers Sysmon Event ID 15
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 15: FileCreateStreamHash
##     This event logs when a named file stream is created, and it generates events that log the hash of the contents of the file to which the stream is assigned (the unnamed stream), as well as the contents of the named stream. There are malware variants that drop their executables or
```

```
configuration settings via browser downloads, and this event is aimed at capturing that based on the browser attaching a Zone.Identifier “mark of the web” stream.
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <FileCreateStreamHash onmatch="include">
##     <TargetFilename condition="end with">.bat</TargetFilename> <!--Batch scripting-->
##   </FileCreateStreamHash>
##
## References:
##   http://www.powertheshell.com/ntfsstreams/
##   https://www.irongeek.com/i.php?page=security/altds
##
def trigger15():
    cmd1 = ["cmd.exe", "/c echo howdy > 15trigger.bat"]
    cmd2 = ["cmd.exe", "/c echo trigger>15trigger.bat:triggerADS.bat"]
    cmd3 = ["cmd.exe", "/c del 15trigger.bat"]

    try:
        ret = subprocess.run(cmd1, timeout=20)
        ret = subprocess.run(cmd2, timeout=20)
        ret = subprocess.run(cmd3, timeout=20)
    except subprocess.TimeoutExpired:
        print("EventID 15: TIMEOUT")
        return

    print("EventID 15: Triggered")
    return

##
## Function Name:
##   trigger16()
##
## Purpose:
##   Triggers Sysmon Event ID 16
##
## Input:
##   Path to Sysmon Executable
##   Path to Desired Sysmon Configuration File
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 16: Sysmon Configuration File Changed
##     Sysmon configuration file changed
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <!--Cannot be filtered.-->
##
## References:
##   https://docs.python.org/3/library/subprocess.html
##
def trigger16(sysmon_path, config_path):

    mycwd = os.getcwd()
    cmd1 = [sysmon_path, "-c", config_path]
    try:
        ret = subprocess.run(cmd1, timeout=30, cwd=mycwd, shell=False)
    except subprocess.TimeoutExpired:
        print("EventID 16: TIMEOUT")
        return

    print("EventID 16: Triggered")
    return

##
## Function Name:
##   trigger1718(), pipeserver(), pipeclient()
##
## Purpose:
##   Triggers Sysmon Event ID 17 and 18
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 17: PipeEvent (Pipe Created)
##     This event generates when a named pipe is created. Malware often uses named pipes for interprocess communication.
##
##   Event ID 18: PipeEvent (Pipe Connected)
##     This event logs when a named pipe connection is made between a client and a server.
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <PipeEvent onmatch="include">
##     <PipeName condition="end with">trigger</PipeName> <!-- Look for Trigger -->
##   </PipeEvent>
##
## References:
##   https://docs.python.org/3/library/subprocess.html
##   https://stackoverflow.com/questions/1430446/create-a-temporary-fifo-named-pipe-in-python
##   https://stackoverflow.com/questions/13319679/createnamedpipe-in-python
##   https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/sans-dfir-2015.pdf
## http://www.bogotobogo.com/python/Multithread/python_multithreading_creating_threads.php
##
def pipeserver():

    cmd = ["cmd.exe", "/c powershell [reflection.Assembly]::LoadWithPartialName('system.core'); $pipe = new-object System.IO.Pipes.NamedPipeServerStream('\\\\.\\pipe\\18trigger'); $pipe.WaitForConnection(); $pipe.Dispose();"]
    try:
        ret = subprocess.run(cmd, timeout=30, shell=False)
    except subprocess.TimeoutExpired:
        print("EventID 17: TIMEOUT")
        return
    print("EventID 17: Triggered")
    return

def pipeclient():

    cmd = ["cmd.exe", "/c powershell [reflection.Assembly]::LoadWithPartialName('system.core'); $pipe=new-object System.IO.Pipes.NamedPipeClientStream('\\\\.\\pipe\\18trigger'); $pipe.Connect(); $pipe.Dispose();"]
    try:
        ret = subprocess.run(cmd, timeout=30, shell=False)

    except subprocess.TimeoutExpired:
        print("EventID 18: TIMEOUT")
        return
    print("EventID 18: Triggered")
    return

def trigger1718():
    c = threading.Thread(target=pipeclient)
    c.start()
    time.sleep(5)
    s = threading.Thread(target=pipeserver)
    s.start()
    time.sleep(5)

    return

##
## Function Name:
##   trigger192021()
##
## Purpose:
##   Triggers Sysmon Event ID 19, 20, and 21
##
## Input:
##   Path to trigger.mof file
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 19: WmiEvent (WmiEventFilter activity detected)
##     When a WMI event filter is registered, which is a method used by malware to execute, this event logs the WMI namespace, filter name and filter expression.
##
##   Event ID 20: WmiEvent (WmiEventConsumer activity detected)
##     This event logs the registration of WMI consumers, recording the consumer name, log, and destination.
##
##   Event ID 21: WmiEvent (WmiEventConsumerToFilter activity detected)
##      When a consumer binds to a filter, this event logs the consumer name and filter path.
##
```

```
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   <WmiEvent onmatch="exclude">
##     <!-- <Name condition="contains">Trigger</Name> Trigger event to verify functionality -->
##   </WmiEvent>
##
## References:
##   https://gist.github.com/mattifestation/aff0cb8bf66c7f6ef44a
##   https://soykablog.wordpress.com/2013/04/17/removing-permanent-wmi-event-registrations-trevor-sullivans-tech-room/
##   https://learn-powershell.net/2013/08/14/powershell-and-events-permanent-wmi-event-subscriptions/
##
def trigger192021(mof_path):

    ## CREATE WMI EVENTS
    cmd1 = ["cmd.exe", "/c", "mofcomp.exe", mof_path]
    try:
        ret = subprocess.run(cmd1, timeout=30, shell=False)
    except subprocess.TimeoutExpired:
        print("EventID 19: TIMEOUT")
        print("EventID 20: DID NOT RUN")
        print("EventID 21: DID NOT RUN")
        return

    time.sleep(5)

    ## REMOVE THE WMI EVENTS WE CREATED, TRIGGERS DELETE EVENTS

    cmd = "powershell.exe"
    args="\"Get-WmiObject -Namespace 'root/subscription' -Class '__FilterToConsumerBinding' -Filter 'Filter=\\\"__EventFilter.Name=\\\\\\\"TriggerFilter\\\\\\\"\\\"' | Remove-WmiObject\" "
    ret = ctypes.windll.shell32.ShellExecuteW(None, u"runas", cmd, args, None, 0)
    time.sleep(5)

    print("EventID 21: Triggered")

    cmd = "powershell.exe"
    args = "\"Get-WmiObject -Namespace 'root/subscription' -Class '__EventFilter' -Filter 'Name=\\\"TriggerFilter\\\"' | Remove-WmiObject\" "
    ret = ctypes.windll.shell32.ShellExecuteW(None, u"runas", cmd, args, None, 0)
    time.sleep(5)

    print("EventID 19: Triggered")

    cmd = "powershell.exe"
    args= "\"Get-WmiObject -Namespace 'root/subscription' -Class 'CommandLineEventConsumer' -Filter 'Name=\\\"TriggerConsumer\\\"' | Remove-WmiObject\" "
    ret = ctypes.windll.shell32.ShellExecuteW(None, u"runas", cmd, args, None, 0)
    time.sleep(5)

    print("EventID 20: Triggered")
    return

##
## Function Name:
##   trigger255()
##
## Purpose:
##   Triggers Sysmon Event ID 255
##
## Sysmon Event Details (from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon):
##   Event ID 255: Error
##      This event is generated when an error occurred within Sysmon.
##      They can happen if the system is under heavy load and certain tasked could not be performed or a bug exists in the Sysmon service.
##      You can report any bugs on the Sysinternals forum or over Twitter (@markrussinovich).
##
## Sysmon Configuration to Match Trigger (https://github.com/SwiftOnSecurity/sysmon-config/blob/master/sysmonconfig-export.xml):
##   Not Implemented
##
def trigger255():
  print("EventID 255: Not Implemented")
  return

##
## Function Name:
##
## Purpose:
##   Triggers all Sysmon events
##
## Input:
##   Modify "sysmon_path", "config_path", and "mof_path" to match your environment locations and names.
##
def main():

    # CHECK FOR ADMIN RIGHTS (your mileage may vary)
    if not has_admin():
        print("ADMIN RIGHTS REQUIRED!!!")
        return

    # GET DETAILS ABOUT CURRENT FILE AND DIRECTORY
    filename = inspect.getframeinfo(inspect.currentframe()).filename
    dirname = os.path.dirname(os.path.abspath(filename))


    # MODIFY THESE VALUES TO MATCH YOUR ENVIRONMENT LOCATIONS AND NAMES
    sysmon_path = "C:\\progra~1\\Sysmon-v6.20\\sysmon.exe"
    config_path = dirname + os.path.sep + "sysmonconfig-modified.xml"
    mof_path = dirname + os.path.sep + "trigger.mof"

    print("VERIFY YOUR FILE LOCATIONS MATCH TO ENSURE TRIGGERS WORK:")
    print("Sysmon Path: ", sysmon_path)
    print("Sysmon Config File: ", config_path)
    print("Manifest File: ", mof_path)
    print("")

    # TRIGGER ALL EVENTS (numerically out of order to prevent timing issues)
    print("[ START ]")
    r = trigger9()
    r = trigger1()
    r = trigger2()
    r = trigger3()
    r = trigger4()
    r = trigger5()
    r = trigger6()
    r = trigger10()
    r = trigger11()
    r = trigger121314()
    r = trigger15()
    r = trigger16(sysmon_path, config_path)
    r = trigger1718()
    r = trigger192021(mof_path)
    r = trigger7()
    r = trigger8()
    print("[ DONE ]")

    return

if __name__== "__main__":
    main()
```

APPENDIX D: TRIGGER MANIFEST FILE (TRIGGER.MOF)

The Python script for the trigger tool uses a manifest file to trigger the Windows Management Instrumentation (WMI) events in Sysmon. The manifest file can be found online at `https://github.com/dsugraduate/dsu2018/`. The file is included below, with line numbers displayed for reference.

```
1   #pragma classflags ("updateonly", "forceupdate")
2   #pragma namespace("\\\\.\\root\\subscription")
3
4   instance of __EventFilter as $MyEventFilter {
5     EventNamespace = "Root\\Cimv2";
6     Name  = "TriggerFilter";
7     Query = "Select * From __InstanceModificationEvent Where
TargetInstance Isa 'Win32_LocalTime' And TargetInstance.Second=5";
8     QueryLanguage = "WQL";
9   };
10
11  instance of CommandLineEventConsumer as $MyConsumer {
12    Name = "TriggerConsumer";
13    CommandLineTemplate = "c:\\windows\\system32\\ping.exe 8.8.8.8";
14    RunInteractively = False;
15    WorkingDirectory = "c:\\windows\\temp";
16  };
17
18  instance of __FilterToConsumerBinding {
19    Consumer = $MyConsumer;
20    Filter   = $MyEventFilter;
21  };
```

APPENDIX E: WINLOGBEAT CONFIGURATION (WINLOGBEAT.YML)

The Winlogbeat configuration file used in this research can be found online at `https://github.com/dsugraduate/dsu2018/`. Many common default options were used from the default YML configuration example provided when Winlogbeat is installed. The only changes made to the default configuration values are displayed below.

```
#======================= Winlogbeat specific options
==========================
winlogbeat.event_logs:
  - name: Microsoft-Windows-Sysmon/Operational
    ignore_older: 72h

#================================ General
=====================================

# The name of the shipper that publishes the network data. It can be used
to group
# all the transactions sent by a single shipper in the web interface.
name: "win7"

# The tags of the shipper are included in their own field with each
# transaction published.
tags: ["windows", "testing"]

# Optional fields that you can specify to add additional information to the
# output.
fields:
  env: testing

#----------------------------- Logstash output ----------------------------
----
output.logstash:
  # The Logstash hosts
  hosts: ["10.1.1.12:5044"]
```

APPENDIX F: LOGSTASH PIPELINE (BEATS.CONF)

The configuration file for the Logstash pipeline created in this research can be found online at `https://github.com/dsugraduate/dsu2018/`. The configuration is included below, with line numbers displayed for reference.

```
1   input {
2     beats {
3       port => 5044
4     }
5   }
6
7   filter {
8
9     # MATCH FILE/IMAGE HASHES (MD5 and SHA256)
10    if ( [event_id] == 1 or [event_id] == 6 or [event_id] == 7 ) {
11      grok {
12        match => [ "[event_data][Hashes]", "MD5=(?<MD5>[0-9A-
F]{32}),SHA256=(?<SHA256>[0-9A-F]{64})" ]
13      }
14    }
15
16    # MATCH SYSMON CONFIG FILE HASH (SHA1 ONLY)
17    if ( [event_id] == 16 ) {
18      grok {
19        match => [ "[event_data][ConfigurationFileHash]",
"SHA1=(?<sysmon_config_SHA1>[0-9A-F]{40})" ]
20      }
21    }
22
23    # GEOIP ON DESTINATION IP
24    if ("" in [event_data][DestinationIp]) {
25      if ( [event_data][DestinationIp] != "127.0.0.1" ) {
26        geoip {
27          source => "[event_data][DestinationIp]"
28        }
29      }
30    }
31
32    # EXTRACT EXTENSION FROM TARGET FILENAME
33    if ( [event_id] == 2 or [event_id] == 11 or [event_id] == 15 ) {
34      grok {
35        #match => [ "[event_data][TargetFilename]",
"(\.(?<file_extension>[.]*[0-9A-Za-z]*)$)" ]
36        match => [ "[event_data][TargetFilename]",
"((?<file_extension>[.]+[0-9A-Za-z]*)$)" ]
37      }
38      translate {
39          field => "file_extension"
40          destination => "file_extension_ransomware"
41          fallback => "false"
```

```
        dictionary_path => '/etc/logstash/conf.d/dict/ransomware-extensions.yaml'
      }
  }

  # FIND MATCHING START DATA
  if ( [event_id] == 5 ) {
    elasticsearch {
      id => "process_duration"
      hosts => ["localhost:9200"]
      query => 'event_id:1 AND event_data.ProcessGuid:"%{[event_data][ProcessGuid]}"'
      fields => { "@timestamp" => "started" }
    }
    date {
      match => ["started", "ISO8601"]
      target => "[started]"
    }
    ruby {
      code => "event.set('duration_seconds', event.get('@timestamp') - event.get('started') ) "
    }
  }

  # TRANSLATE SYSMON EVENT ID INTO EVENT NAME
  translate {
    field => "event_id"
    destination => "event_id_text"
    dictionary_path => '/etc/logstash/conf.d/dict/sysmon_event_ids.yaml'
  }

  # IF WE WANT TO QUERY VT
  if ("" in [SHA256]) {
#  if ([SHA256] == "asdf") {

    # ONLY QUERY VT ON PROCESS CREATION EVENTS THAT AREN'T WHITELISTED
    if ( [event_id] == 1 or [event_id] == 6 or [event_id] == 7 ) {
      translate {
        field => "SHA256"
        destination => "whitelisted"
        fallback => "false"
        dictionary_path => '/etc/logstash/conf.d/dict/whitelist.yaml'
      }

      if ([whitelisted] == "false")
      {
        rest {
          request => {
            url => "http://www.virustotal.com/vtapi/v2/file/report"
            method => "GET"
            headers => {
              "Accept-Encoding" => "gzip, deflate"
              "User-Agent" => "gzip"
            }
            params => {
              "apikey" => "INSERT_YOUR_API_KEY_HERE"
              "resource" => "%{SHA256}"
            }
          }
          json => true
          target => "virustotal"
          fallback => {                        # hash describing a default in case of error
            "virustotal_error" => "true"
          }
        }
      }
    }
  }
}

output {
  elasticsearch {
    hosts => localhost
    manage_template => false
    index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"
    document_type => "%{[@metadata][type]}"
  }
}

```

APPENDIX G: LOGSTASH DICTIONARY (SYSMON_EVENT_IDS.YAML)

The dictionary file for the inclusion of Sysmon Event ID text used in the Logstash pipeline can be found online at `https://github.com/dsugraduate/dsu2018/`. The file is included below, with line numbers displayed for reference.

```yaml
"1": "Process Creation"
"2": "A Process Changed A File Creation Time"
"3": "Network Connection"
"4": "Sysmon Service State Changed"
"5": "Process Terminated"
"6": "Driver Loaded"
"7": "Image Loaded"
"8": "CreateRemoteThread"
"9": "RawAccessRead"
"10": "ProcessAccess"
"11": "FileCreate"
"12": "RegistryEvent (Object Create And Delete)"
"13": "RegistryEvent (Value Set)"
"14": "RegistryEvent (Key and Value Rename)"
"15": "FileCreateStreamHash"
"16": "Sysmon Configuration Change"
"17": "PipeEvent (Pipe Created)"
"18": "PipeEvent (Pipe Connected)"
"19": "WmiEvent (WmiEventFilter Activity Detected)"
"20": "WmiEvent (WmiEventConsumer Activity Detected)"
"21": "WmiEvent (WmiEventConsumerToFilter Activity Detected)"
"255": "Error"
```

APPENDIX H: LOGSTASH DICTIONARY (RANSOMWARE_EXTENSIONS.YAML)

The dictionary file used in this research identify common ransomware file extensions during the Logstash pipeline can be found online at `https://github.com/dsugraduate/dsu2018/`. The file is included below, with line numbers displayed for reference.

```
".726": "true"
".8899": "true"
".asasin": "true"
".cerber3": "true"
".coded": "true"
".crypt": "true"
".AK47": "true"
".STN": "true"
```

APPENDIX I: LOGSTASH DICTIONARY (WHITELIST.YAML)

The dictionary file for the whitelist of SHA256 hashes used in the Logstash pipeline can be found online at `https://github.com/dsugraduate/dsu2018/`. The file is included below, with line numbers displayed for reference.

```
1  "36414C7E57AFA6136D77FD47F4C55102E35F2475FBCD719728DA7D14B1590E2A":
"C:\\Windows\\SysWOW64\\reg.exe"
2  "DB24550C3183FC38F9440134322F124447DFE0A3564490180418305D7899D159":
"C:\\Windows\\SysWOW64\\attrib.exe"
3  "17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE":
"C:\\Windows\\SysWOW64\\cmd.exe"
4  "FECD6785984DBB61C6C0EA8A3D8DAF034346E47C88ECA90564F855C2548E40B2":
"C:\\Windows\\System32\\mobsync.exe"
5  "34DF739526C114BB89470B3B650946CBF7335CB4A2206489534FB05C1FC143A8":
"C:\\Windows\\System32\\ipconfig.exe"
6  "9DFD80610CBBC9188F6C6BC85C87016B0AE42254FC289C2B578E85282BDD9C23":
"C:\\Windows\\System32\\taskhost.exe"
7  "93B2ED4004ED5F7F3039DD7ECBD22C7E4E24B6373B4D9EF8D6E45A179B13A5E8":
"C:\\Windows\\System32\\svchost.exe"
8  "66C371914C6F262906C875DC2B489C7F41CDFFD94EB3C7D9A2ED2CDC1192EC11":
"C:\\Program Files\\Internet Explorer\\iexplore.exe"
9  "E09BF4D27555EC7567A598BA89CCC33667252CEF1FB0B604315EA7562D18AD10":
"C:\\Windows\\SysWOW64\\vssadmin.exe"
10 "AE255C2230F699D207323DFB289004C12B53D5CF6B12312DB11D845FD87315EC":
"C:\\Program Files (x86)\\Internet Explorer\\iexplore.exe"
11 "47B801E623254CF0202B3591CB5C019CABFB52F123C7D47E29D19B32F1F2B915":
"C:\\Windows\\System32\\VSSVC.exe"
12 "14262982A64551FDE126339B22B993B6E4AED520E53DD882E67D887B6B66F942":
"C:\\Windows\\System32\\PING.EXE"
13 "74B3152A28D4F1A4FFF46B279ABF0EEF666DF0F8245EFAE1E71E6E375889FF70":
"C:\\Windows\\System32\\taskkill.exe"
14 "DA3AD32583644BD20116F0479C178F7C7C0B730728F4C02A438C0D19378C83D9":
"C:\\Windows\\System32\\wbem\\WMIC.exe"
15 "C6A91CBA00BF87CDB064C49ADAAC82255CBEC6FDD48FD21F9B3B96ABF019916B":
"C:\\Windows\\System32\\calc.exe"
16 "D5BC504277172BE5C54B60AD5C13209DC1F729131DEF084DE3EC8C72E54C58EF":
"C:\\Windows\\explorer.exe"
17 "9DFD80610CBBC9188F6C6BC85C87016B0AE42254FC289C2B578E85282BDD9C23":
"C:\\Windows\\System32\\taskhost.exe"
18 "F34F231D117CCDFEBB9CB35C8D6FDFA7051DA27FDC1204FCCFF361FC0B13A0FF":
"C:\\Windows\\System32\\wbem\\WmiPrvSE.exe"
19 "2FBBEC4CACB5161F68D7C2935852A5888945CA0F107CF8A1C01F4528CE407DE3":
"C:\\Windows\\System32\\msdtc.exe"
20 "F5C5758B5A1B09D44A0CDA1D4EAB5E35D4AC6A78AD39A69011F4A82CEAD958FF":
"C:\\Program Files\\Winlogbeat\\winlogbeat.exe"
21 "ED9FB40C3CB5BA0A9AC3ADE80B503F5D7128016C75852E612A6C838F04401EA3":
"C:\\Program Files\\VMware\\VMware Tools\\vmtoolsd.exe"
```

```
"2E2CE2F3FBB2A00AD374AF9419B41F8DD09244E299D5FB32326B0B775857ECF7":
"C:\\Windows\\Sysmon.exe"
"1398EF88CDE0195F9CBEA696591E6731209540320A622D4B777992AFC7DBFD1E":
"C:\\Windows\\Sysmon.exe"
"11C194D9ADCE90027272C627D7FBF3BA5025FF0F7B26A8333F764E11E1382CF9":
"C:\\Windows\\System32\\userinit.exe"
"415A23E6B3A446186FEBD84D86222C4FB654E9FD2BFDD8D877C6AB8C52D60C7B":
"C:\\Windows\\System32\\EOSNotify.exe"
"AF0A85066A7983878DC1C663811CE61C6CA1912DC956184F878B7B82DB93C651":
"C:\\Windows\\System32\\spoolsv.exe"
"F9B6DDF62B4175093DD38C00520C7F0D52FBAB0077A8ED1391DD5188E400F481":
"C:\\Windows\\System32\\lsass.exe"
"CB1C6018FC5C15483AC5BB96E5C2E2E115BB0C0E1314837D77201BAB37E8C03A":
"C:\\Windows\\System32\\csrss.exe"
"405F03534BE8B45185695F68DEB47D4DAF04DCD6DF9D351CA6831D3721B1EFC4":
"C:\\Windows\\System32\\rundll32.exe"
"C7FD161906C7226E86C7AE00506A1C7862D21ED4BB3FEF34A4B20C999A5A3E2A":
"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell_ise.exe"
"CF50CE4474AA811D4AB00B44610A112244C2267018DF15BBF352A8FE4119C904":
"C:\\Users\\user1\\AppData\\Local\\Programs\\Python\\Python36\\pythonw.exe"
"3B2F41EFDA68C82D9D50AF329AC9B403C806CBE74F87917CDB350E542ADDA017":
"C:\\Windows\\System32\\wbem\\unsecapp.exe"
"A8FDBA9DF15E41B6F5C69C79F66A26A9D48E174F9E7018A371600B866867DAB8":
"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe"
"B212E59E4C7FE77F6F189138D9D8B151E50EB83A35D6EADFB1E4BB0B4262C484":
"C:\\Windows\\System32\\mmc.exe"
"7AA73B8E7D4D700C164D0410DCF84EA1CCCB0F7DD513E47A2EF0DAE5F16CAE45":
"C:\\Windows\\System32\\winlogon.exe"
"3E51D778DB271D53DA0B1F075B5F515D4A38F70F9C6F083B2646DAE8B9E0281A":
"C:\\Windows\\System32\\smss.exe"
"A0E2EB71358AD1BEA9C0ECBBF5B1AEAC3EAB769A73DD799320A98A42D0F1D2BE":
"C:\\Program Files\\Sysmon\\Sysmon.exe"
"DB06C3534964E3FC79D2763144BA53742D7FA250CA336F4A0FE724B75AAFF386":
"C:\\Windows\\System32\\cmd.exe"
"2F1A0D69B87CA32BB5DE2582CDD8B795823EB268A57B9D179609BBC5D046E865":
"C:\\Program Files\\Winlogbeat\\winlogbeat.exe"
"435603112D830DB82A3622F7A3387AE91BA828435A7A8DBA35F010C6F8B792E3":
"C:\\Program Files\\VMware\\VMware Tools\\VMware
CAF\\pme\\bin\\ManagementAgentHost.exe"
"9D67EFA96796E24CBEACE6190B29B8DD03271326B83D87582E4F8515D753A429":
"C:\\Program Files\\VMware\\VMware Tools\\VMware VGAuth\\VGAuthService.exe"
"09AB0535A54C2E2962F0FD06988D99060F8CECA39B07AC00A63204C773B95893":
"C:\\Windows\\System32\\LogonUI.exe"
"D205B2C163E78AB42A5D67D7664EF6B75EA0374FF0924467D624F9DB0611F0AD":
"C:\\Windows\\System32\\lsm.exe"
"A86D6A6D1F5A0EFCD649792A06F3AE9B37158D48493D2ECA7F52DCC1CB9B6536":
"C:\\Windows\\System32\\services.exe"
"2B658D0A86220608C9F8FAC30494A0368778FD7DA5E6529257913B45A52C4A55":
"C:\\Program Files\\VMware\\VMware Tools\\vmacthlp.exe"
"C4E98F07170CEC69CACDD5CEDB8927E48A2A299CB1B8CDA87526E768AF6174F0":
"C:\\Windows\\System32\\wininit.exe"
"7BC847CE6C2D29C334F0D1600BBBDE3933FF45F6BEE5186F442E6270A3F9EC4E":
"C:\\Windows\\System32\\autochk.exe"
"435603112D830DB82A3622F7A3387AE91BA828435A7A8DBA35F010C6F8B792E3":
"C:\\Program Files\\VMware\\VMware Tools\\VMware
CAF\\pme\\bin\\ManagementAgentHost.exe"
```

```
"9D67EFA96796E24CBEACE6190B29B8DD03271326B83D87582E4F8515D753A429":
"C:\\Program Files\\VMware\\VMware Tools\\VMware VGAuth\\VGAuthService.exe"
"A4567E548F8DA31C6EF694E79DA6019C7E24308A8D18081E423BED0D551EE393":
"C:\\Windows\\System32\\wbem\\mofcomp.exe"
"2C7257C7D9D0064574296C5B96C6834B4FA2679E04097C2DD21B7AE3BB6940DD":
"C:\\Users\\user1\\AppData\\Local\\Temp\\Procmon64.exe"
"978F95FC63699F1E769663E32BAE596148742B3038F743A51AF346CA4D2F56D0":
"C:\\Users\\user1\\Desktop\\code\\ProcessMonitor\\Procmon.exe"
"8481A8EC19CB656CE328C877D5817D317203BA34424A2E9D169DDCE5BF2CD2B0":
"C:\\Windows\\System32\\DeviceDisplayObjectProvider.exe"
"6DCE7D58EBB0D705FCB4179349C441B45E160C94E43934C5ED8FA1964E2CD031":
"C:\\Windows\\System32\\schtasks.exe"
"61BD24487C389FC2B939CE000721677CC173BDE0EDCAFCCFF81069BBD9987BFD":
"C:\\Windows\\System32\\sdclt.exe"
"7EE656884090AEDB0A615E0641ECC250D6204CB9570CA02216F2B7D5F381E021":
"C:\\Windows\\System32\\wsqmcons.exe"
"67E045FD25809E8CA486B1D17EB33667835FBC04974DD65DC07FCFC7E9A3D254":
"C:\\Windows\\System32\\CompatTel\\diagtrackrunner.exe"
"CB1822A981E9821D571AF16B7E37BEBA5FEB8E3DEDCDD0461119AF9AAC0358B3":
"C:\\Windows\\SysWOW64\\taskkill.exe"
"F7AD4B09AFB301CE46DF695B22114331A57D52E6D4163FF74787BF68CCF44C78":
"C:\\Windows\\SysWOW64\\dllhost.exe"
"4DE7FA20E3224382D8C4A81017E5BDD4673AFBEF9C0F017E203D7B78977FBF8C":
"C:\\Windows\\System32\\vssadmin.exe"
"CF45DC3EFB09309F4A2F7275A8D011A0D0CF65B3DD69A867A708A78A1076BBB7":
"C:\\Program Files (x86)\\Notepad++\\notepad++.exe"
"933E1778B2760B3A9194C2799D7B76052895959C3CAEDEFB4E9D764CBB6AD3B5":
"C:\\Windows\\System32\\notepad.exe"
"C9FF4DE365930BC063A082FF3DFEEC50247E9137F80DCA0B08C573326F659F58":
"C:\\Windows\\System32\\drivers\\vmmemctl.sys"
"42AF63EA6C594D3610DC3A201E564E676891828CC29B16AF127CCB51A3E5B9BB":
"C:\\Windows\\System32\\drivers\\vmhgfs.sys"
"2044B0B94C70368F43D5D31399BD3AB516482FFD5A3948D9271CF778FDA26A6E":
"C:\\Windows\\System32\\drivers\\vmusbmouse.sys"
"4FB7B3A7F4CDE02A906D56386CBA474FB996717734F134C769B618A373F42A1F":
"C:\\Windows\\System32\\drivers\\vm3dmp.sys"
"9D703AE54D05CEE7E8D61C3E1ECE1498905E5C41BD700AE8E560F99AEBFA87EA":
"C:\\Windows\\System32\\drivers\\vmmouse.sys"
"D0528DDE112DD388F6B9279505485EED7D5CC706E2206F6374BF0E92AD1BEC7F":
"C:\\Windows\\System32\\drivers\\vmrawdsk.sys"
"4CA10DBA7FF487FDB3F1362A3681D7D929F5AA1262CDFD31B04C30826983FB1D":
"C:\\Windows\\SysWOW64\\PING.EXE"
"B56AFE7165AD341A749D2D3BD925D879728A1FE4A4DF206145C1A69AA233F68B":
"C:\\Windows\\SysWOW64\\notepad.exe"
"949485BA939953642714AE6831D7DCB261691CAC7CBB8C1A9220333801F60820":
"C:\\Windows\\SysWOW64\\mshta.exe"
"661F5D4CE4F0A6CB32669A43CE5DEEC6D5A9E19B2387F22C5012405E92169943":
"C:\\Windows\\SysWOW64\\netsh.exe"
"62E449589E1F082E8DE3FA4D775871E1C66A272E3BD1FE5CC33EEEB40351CD13":
"C:\\Windows\\System32\\Defrag.exe"
"F583C6D572ABFDC9EFBA48480814ADD63FF3626A23C2880449E482D89AF24427":
"C:\\Program Files\\7-Zip\\7zG.exe"
"AAF48B577885D37A63604E01D5190C1F36705B5E26C4231ABEE42E86F11EDEB0":
"C:\\Windows\\System32\\aitagent.exe"
"CE29FAB2DE75FBB40A2DA29C9ED547CBEF758C58E0EBCE461CD56E42194EA6B0":
"C:\\Program Files\\VMware\\VMware Tools\\VMwareResolutionSet.exe"
```

```
"3B9AD8E2C1D03FF941A7C9192A605F31671B107DEF6FF503A71A0FB2C5BBD659":
"C:\\Windows\\System32\\net.exe"
"021D7CE4D95A1F8811AD5085ED091C2066C544152DEF2D370EFF464381B7D2ED":
"C:\\Windows\\System32\\lpremove.exe"
"0F084CCC40CBF7C3C7472DDAD609B5FD31AACAFA44E23F9EC7E9E2184713B986":
"C:\\Windows\\System32\\net1.exe"
"57E3E3EC6F548394C932BFCCAC03F02EA909B4DF332921E19454C71D7DBD362D":
"C:\\Program Files\\Sysmon-v6.10\\Sysmon.exe"
```

APPENDIX J: LOGSTASH CONFIGURATION (LOGSTASH.YML)

The Logstash configuration file used in this research can be found online at `https://github.com/dsugraduate/dsu2018/`. The default Logstash configuration example file available when Logstash is first installed was used in this research. The only changes made from the file were the values **`path.data`** and **`xpack.monitoring.enabled`**, as indicated below.

```
# ------------ Data path ------------------
#
# Which directory should be used by logstash and its plugins
# for any persistent needs. Defaults to LOGSTASH_HOME/data
#
path.data: /var/lib/logstash
#

# disable xpack settings
xpack.monitoring.enabled: false
```

APPENDIX K: ELASTICSEARCH CONFIGURATION (ELASTICSEARCH.YML)

The Elasticsearch configuration file used in this research can be found online at `https://github.com/dsugraduate/dsu2018/`. The default Elasticsearch configuration example file available when Elasticsearch is first installed was used. The only changes made in the file were the values shown below.

.

```
#
# Lock the memory on startup:
#
bootstrap.memory_lock: true

# Set the bind address to a specific IP (IPv4 or IPv6):
#
network.host: 127.0.0.1

#
# disable x-pack security
xpack.security.enabled: false

xpack.watcher.enabled: true

xpack.notification.email.account:
      gmail_account:
          profile: gmail
          smtp:
             auth: true
             starttls.enable: true
             host: smtp.gmail.com
             port: 587
             user: <insert_email_address>
             password: <insert_password>

#disable for now to get tables to show up. Change in production.
xpack.notification.email.html.sanitization.enabled: false
```

APPENDIX L: KIBANA CONFIGURATION (KIBANA.YML)

The Kibana configuration file used in this research can be found online at `https://github.com/dsugraduate/dsu2018/`. All defaults were used from the Kibana configuration example file provided when Kibana is installed, except for the xpack setting shown below.

```
# disable xpack security
xpack.security.enabled: false
```

APPENDIX M: KIBANA INDEX FOR WINLOGBEAT LOGS

The Kibana Winlogbeat index used in this research can be found online at `https://github.com/dsugraduate/dsu2018/`. The index is included below, with line numbers displayed for reference. This index was automatically generated by Elasticsearch as Winlogbeat events were parsed and stored in Elasticsearch.

```
{
  "winlogbeat-2018.01.23": {
    "aliases": {},
    "mappings": {
      "_default_": {
        "properties": {
          "geoip": {
            "dynamic": "true",
            "properties": {
              "ip": {
                "type": "ip"
              },
              "latitude": {
                "type": "float"
              },
              "location": {
                "type": "geo_point"
              },
              "longitude": {
                "type": "float"
              }
            }
          }
        }
      },
      "wineventlog": {
        "properties": {
          "@timestamp": {
            "type": "date"
          },
          "@version": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "MD5": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "SHA256": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "beat": {
            "properties": {
              "hostname": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "name": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "version": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              }
            }
          },
          "computer_name": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "duration_seconds": {
            "type": "float"
          },
          "event_data": {
            "properties": {
              "CommandLine": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "CreationUtcTime": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "CurrentDirectory": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "DestinationHostname": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "DestinationIp": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "DestinationIsIpv6": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "DestinationPort": {
```

```
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "DestinationPortName": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "Details": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "EventType": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "Hashes": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "Image": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "ImageLoaded": {
              "type": "text",
              "fields": {
                "keyword": {
```

```
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"Initiated": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"IntegrityLevel": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"LogonGuid": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"LogonId": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"NewThreadId": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"ParentCommandLine": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
```

```
                  }
                },
                "ParentImage": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "ParentProcessGuid": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "ParentProcessId": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "PreviousCreationUtcTime": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "ProcessGuid": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "ProcessId": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "Protocol": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "SchemaVersion": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "Signature": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "SignatureStatus": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "Signed": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "SourceHostname": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "SourceImage": {
                  "type": "text",
                  "fields": {
                    "keyword": {
```

```
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"SourceIp": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"SourceIsIpv6": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"SourcePort": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"SourceProcessGuid": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"SourceProcessId": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"StartAddress": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
```

```
          }
        },
        "StartFunction": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "StartModule": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "State": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "TargetFilename": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "TargetImage": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "TargetObject": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        },
        "TargetProcessGuid": {
```

```
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "TargetProcessId": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "TerminalSessionId": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "User": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "UtcTime": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "Version": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                }
              }
            },
            "event_id": {
              "type": "long"
            },
            "event_id_text": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "fields": {
              "properties": {
                "env": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                }
              }
            },
            "file_extension": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "file_extension_ransomware": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "geoip": {
              "dynamic": "true",
              "properties": {
                "city_name": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "continent_code": {
                  "type": "text",
                  "fields": {
                    "keyword": {
```

```
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "country_code2": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "country_code3": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "country_name": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "dma_code": {
                "type": "long"
              },
              "ip": {
                "type": "ip"
              },
              "latitude": {
                "type": "float"
              },
              "location": {
                "type": "geo_point"
              },
              "longitude": {
                "type": "float"
              },
              "postal_code": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "region_code": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "region_name": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "timezone": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              }
            }
          },
          "host": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "level": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "log_name": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "message": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "opcode": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "process_id": {
            "type": "long"
          },
          "provider_guid": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "record_number": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "source_name": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "started": {
            "type": "date"
          },
          "tags": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
```

```
          },
          "task": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "thread_id": {
            "type": "long"
          },
          "type": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "user": {
            "properties": {
              "domain": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "identifier": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "name": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "type": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
```

```
        }
      }
    }
  }
},
"version": {
  "type": "long"
},
"virustotal": {
  "properties": {
    "md5": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "permalink": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "positives": {
      "type": "long"
    },
    "resource": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "response_code": {
      "type": "long"
    },
    "scan_date": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "scan_id": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "scans": {
      "properties": {
        "ALYac": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "AVG": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "AVware": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "Ad-Aware": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "AegisLab": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
              }
            }
          }
        }
      },
      "AhnLab-V3": {
        "properties": {
          "detected": {
            "type": "boolean"
          },
          "result": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "update": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "version": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          }
        }
      },
      "Antiy-AVL": {
        "properties": {
          "detected": {
            "type": "boolean"
          },
          "result": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "update": {
            "type": "text",
            "fields": {
              "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "Arcabit": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "Avast": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "Avast-Mobile": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "Avira": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "Baidu": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "BitDefender": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "Bkav": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
```

```
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"CAT-QuickHeal": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"CMC": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "update": {
```

```
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "version": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      }
                    }
                  },
                  "ClamAV": {
                    "properties": {
                      "detected": {
                        "type": "boolean"
                      },
                      "result": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "update": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "version": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      }
                    }
                  },
                  "Comodo": {
                    "properties": {
                      "detected": {
                        "type": "boolean"
```

```
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"CrowdStrike": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Cybereason": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Cylance": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Cyren": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"DrWeb": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"ESET-NOD32": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Emsisoft": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Endgame": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"F-Prot": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"F-Secure": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
```

```
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Fortinet": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"GData": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Ikarus": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Invincea": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Jiangmin": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "update": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "version": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      }
                    }
                  },
                  "K7AntiVirus": {
                    "properties": {
                      "detected": {
                        "type": "boolean"
                      },
                      "result": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "update": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "version": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      }
                    }
                  },
                  "K7GW": {
                    "properties": {
                      "detected": {
                        "type": "boolean"
                      },
                      "result": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "update": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "version": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      }
                    }
                  },
                  "Kaspersky": {
                    "properties": {
                      "detected": {
                        "type": "boolean"
                      },
                      "result": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "update": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Kingsoft": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"MAX": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Malwarebytes": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"McAfee": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"McAfee-GW-Edition": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"MicroWorld-eScan": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Microsoft": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"NANO-Antivirus": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Paloalto": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Panda": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
```

```
      }
    }
  }
},
"Qihoo-360": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Rising": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"SUPERAntiSpyware": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"SentinelOne": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
```

```
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "version": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      }
                    }
                  },
                  "Sophos": {
                    "properties": {
                      "detected": {
                        "type": "boolean"
                      },
                      "result": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "update": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "version": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      }
                    }
                  },
                  "Symantec": {
                    "properties": {
                      "detected": {
                        "type": "boolean"
                      },
                      "result": {
```

```
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "update": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "version": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                }
              }
            },
            "Tencent": {
              "properties": {
                "detected": {
                  "type": "boolean"
                },
                "result": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "update": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
                    }
                  }
                },
                "version": {
                  "type": "text",
                  "fields": {
                    "keyword": {
                      "type": "keyword",
                      "ignore_above": 256
```

```
                          }
                        }
                      }
                    }
                  },
                  "TheHacker": {
                    "properties": {
                      "detected": {
                        "type": "boolean"
                      },
                      "update": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "version": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      }
                    }
                  },
                  "TotalDefense": {
                    "properties": {
                      "detected": {
                        "type": "boolean"
                      },
                      "update": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "version": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      }
                    }
                  },
                  "TrendMicro": {
                    "properties": {
```

```
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "TrendMicro-HouseCall": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "VBA32": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "update": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
            "version": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            }
          }
        },
        "VIPRE": {
          "properties": {
            "detected": {
              "type": "boolean"
            },
            "result": {
              "type": "text",
              "fields": {
                "keyword": {
                  "type": "keyword",
                  "ignore_above": 256
                }
              }
            },
                              "update": {
                                "type": "text",
                                "fields": {
                                  "keyword": {
                                    "type": "keyword",
                                    "ignore_above": 256
                                  }
                                }
                              },
                              "version": {
                                "type": "text",
                                "fields": {
                                  "keyword": {
                                    "type": "keyword",
                                    "ignore_above": 256
                                  }
                                }
                              }
                            }
                          },
                          "ViRobot": {
                            "properties": {
                              "detected": {
                                "type": "boolean"
                              },
                              "result": {
                                "type": "text",
                                "fields": {
                                  "keyword": {
                                    "type": "keyword",
                                    "ignore_above": 256
                                  }
                                }
                              },
                              "update": {
                                "type": "text",
                                "fields": {
                                  "keyword": {
                                    "type": "keyword",
                                    "ignore_above": 256
                                  }
                                }
                              },
                              "version": {
                                "type": "text",
                                "fields": {
                                  "keyword": {
                                    "type": "keyword",
                                    "ignore_above": 256
                                  }
                                }
                              }
                            }
                          },
                          "Webroot": {
                            "properties": {
                              "detected": {
                                "type": "boolean"
                              },
                              "result": {
                                "type": "text",
                                "fields": {
                                  "keyword": {
                                    "type": "keyword",
                                    "ignore_above": 256
                                  }
                                }
                              },
                              "update": {
                                "type": "text",
                                "fields": {
                                  "keyword": {
                                    "type": "keyword",
                                    "ignore_above": 256
                                  }
                                }
                              },
                              "version": {
                                "type": "text",
                                "fields": {
                                  "keyword": {
                                    "type": "keyword",
                                    "ignore_above": 256
                                  }
                                }
                              }
                            }
                          },
                          "WhiteArmor": {
                            "properties": {
                              "detected": {
                                "type": "boolean"
                              },
                              "result": {
                                "type": "text",
                                "fields": {
                                  "keyword": {
                                    "type": "keyword",
                                    "ignore_above": 256
                                  }
                                }
                              },
                              "update": {
                                "type": "text",
                                "fields": {
                                  "keyword": {
                                    "type": "keyword",
                                    "ignore_above": 256
                                  }
                                }
                              }
                            }
                          },
                          "Yandex": {
```

```
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Zillya": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"ZoneAlarm": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"Zoner": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"eGambit": {
  "properties": {
    "detected": {
      "type": "boolean"
    },
    "result": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "update": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    },
    "version": {
      "type": "text",
      "fields": {
        "keyword": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
},
"nProtect": {
  "properties": {
```

```
                      "detected": {
                        "type": "boolean"
                      },
                      "result": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "update": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      },
                      "version": {
                        "type": "text",
                        "fields": {
                          "keyword": {
                            "type": "keyword",
                            "ignore_above": 256
                          }
                        }
                      }
                    }
                  }
                }
              },
              "sha1": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "sha256": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              },
              "total": {
                "type": "long"
              },
              "verbose_msg": {
                "type": "text",
                "fields": {
                  "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                  }
                }
              }
            }
          },
          "whitelisted": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          }
        }
      }
    },
    "settings": {
      "index": {
        "creation_date": "1516665604694",
        "number_of_shards": "5",
        "number_of_replicas": "1",
        "uuid": "54kZ3HndTgSds8Ve7ng0ew",
        "version": {
          "created": "5060599"
        },
        "provided_name": "winlogbeat-2018.01.23"
      }
    }
  }
}
```

APPENDIX N: WATCHER ALERT FOR VIRUSTOTAL HITS

This research created a Watcher alert for reporting malicious processes according to VirusTotal results. The watcher code can be found online at `https://github.com/dsugraduate/dsu2018/`. The code is included below for convenience, with line numbers displayed for reference.

```
{
  "trigger": {
    "schedule": {
      "interval": "1m"
    }
  },
  "input": {
    "search": {
      "request": {
        "search_type": "query_then_fetch",
        "indices": [
          "<winlogbeat-{now-2m}>",
          "<winlogbeat-{now}>"
        ],
        "types": [],
        "body": {
          "query": {
            "bool": {
              "filter": {
                "range": {
                  "@timestamp": {
                    "gt": "now-1m"
                  }
                }
              },
              "must": {
                "range": {
                  "virustotal.positives": {
                    "gt": 0
                  }
                }
              }
            }
          }
        }
      }
    }
  },
  "condition": {
    "compare": {
      "ctx.payload.hits.total": {
        "gt": 0
      }
    }
  },
  "actions": {
    "send_email": {
      "email": {
        "profile": "gmail",
        "attachments": {
          "dashboard.pdf": {
            "reporting": {
              "url": "http://127.0.0.1:5601/api/reporting/generate/dashboard/AWDIhv9bq-FH7NNrhWya"
            }
          },
          "data.yml": {
            "data": {
              "format": "yaml"
            }
          }
        },
        "to": [
          "<insert_email_address>"
        ],
        "subject": "[WATCHER] Malicious Virustotal File Executed",
        "body": {
          "html": "<h1>{{ctx.payload.hits.total}} Malicious Process Creation Events In Past 1 Minute</h1>  <h3>Hashes:</h3>  <ol> {{#ctx.payload.hits.hits}} <li>{{_source.virustotal.scans.Malwarebytes.result}} <a href='https://www.virustotal.com/#/file/{{_source.SHA256}}'> {{_source.SHA256}}</a></li> {{/ctx.payload.hits.hits}}</ol>"
        }
      }
    }
  }
}
```

APPENDIX O: WATCHER ALERT FOR SUSPICIOUS FILE EXTENSIONS

This research created a Watcher alert for reporting malicious processes according to VirusTotal results. The watcher code can be found online at `https://github.com/dsugraduate/dsu2018/`. The code is included below, with line numbers displayed for reference.

```
{
  "trigger": {
    "schedule": {
      "interval": "1m"
    }
  },
  "input": {
    "search": {
      "request": {
        "search_type": "query_then_fetch",
        "indices": [
          "<winlogbeat-{now-2m}>",
          "<winlogbeat-{now}>"
        ],
        "types": [],
        "body": {
          "from": 0,
          "size": 10000,
          "query": {
            "bool": {
              "filter": {
                "range": {
                  "@timestamp": {
                    "gte": "now-1m"
                  }
                }
              },
              "must": {
                "terms": {
                  "file_extension.keyword": [
                    ".402",
                    ".4035",
                    ".4090",
                    ".4091",
                    ".452",
                    ".707",
                    ".725",
                    ".726",
                    ".8899",
                    ".911",
                    ".f41o1",
                    ".2cXpCihgsVxB3",
                    ".3ncrypt3d",
                    ".au1crypt",
                    ".BONUM",
                    ".BRT92",
                    ".BUSH",
                    ".C8B089F",
                    ".CHAK",
                    ".clinTON",
                    ".crypt",
                    ".FIX",
                    ".fuck",
                    ".goro",
                    ".gotham",
                    ".granny",
                    ".happ",
                    ".lpcrestore",
                    ".keepcalm",
                    ".LIN",
                    ".MAKB",
                    ".medal",
                    ".mtk118",
                    ".needdecrypt",
                    ".needkeys",
                    ".NIGGA",
                    ".nWcrypt",
                    ".paycyka",
                    ".pizdec",
                    ".pscrypte",
                    ".ReaGAN",
                    ".rubmblegoodboy",
                    ".s1crypt",
                    ".scorp",
                    ".sea",
                    ".skunk",
                    ".Trump",
                    ".UNLIS",
                    ".vdul",
                    ".wallet",
                    ".write_",
                    ".YAYA",
                    ".zuzya",
                    "..doc",
                    ".xlsm",
                    ".AK47",
                    ".STN",
                    ".coded",
                    ".cerber3",
                    ".asasin",
                    ".ykcol"
                  ]
                }
              }
            }
          },
          "sort": [
            {
```

```
              "event_data.UtcTime.keyword": "asc"
            }
          ]
        }
      }
    }
  },
  "condition": {
    "compare": {
      "ctx.payload.hits.total": {
        "gt": 0
      }
    }
  },
  "actions": {
    "send_email": {
      "email": {
        "profile": "gmail",
        "attachments": {
          "dashboard.pdf": {
            "reporting": {
              "url":
"http://127.0.0.1:5601/api/reporting/generate/dashboard/AWDPFPvTvSdcMfsiRhD
e"
            }
          },
          "data.yml": {
            "data": {
              "format": "yaml"
            }
          }
        },
        "to": [
          "<insert_email_address>"
        ],
        "subject": "[WATCHER] Potentially Malicious File Extension
                   Found",
        "body": {
          "html": "<h1>{{ctx.payload.hits.total}} Potentially Malicious
      File Extensions Seen In Past 1 Minute</h1>  <h3>File
      Extensions (timestamp --- extension --- file):</h3>  <ol>
      {{#ctx.payload.hits.hits}} <li>{{_source.event_data.UtcTime}} ---
      {{_source.file_extension}}  ---
      {{_source.event_data.TargetFilename}}</li>
      {{/ctx.payload.hits.hits}}</ol> <h3>Elastic Discover Query:</h3>
<a
href='http://127.0.0.1:5601/app/kibana#/discover/AWDPG23_vSdcMfsiRhqc'>http
://127.0.0.1:5601/app/kibana#/discover/AWDPG23_vSdcMfsiRhqc</a>"
        }
      }
    }
  }
}
```

APPENDIX P: WATCHER ALERT FOR CHANGES TO FILE CREATION TIMES

This research created a Watcher alert for reporting when a file creation time was changed by a process. The watcher code can be found online at `https://github.com/dsugraduate/dsu2018/`. The code is included below, with line numbers displayed for reference.

```
{
  "trigger": {
    "schedule": {
      "interval": "1m"
    }
  },
  "input": {
    "search": {
      "request": {
        "search_type": "query_then_fetch",
        "indices": [
          "<winlogbeat-{now-2m}>",
          "<winlogbeat-{now}>"
        ],
        "types": [],
        "body": {
          "from": 0,
          "size": 10000,
          "query": {
            "bool": {
              "filter": [
                {
                  "range": {
                    "@timestamp": {
                      "gte": "now-1m"
                    }
                  }
                },
                {
                  "term": {
                    "event_id": 2
                  }
                }
              ]
            }
          },
          "sort": [
            {
              "event_data.UtcTime.keyword": "asc"
            }
          ]
        }
      }
    }
  },
  "condition": {
    "compare": {
      "ctx.payload.hits.total": {
        "gt": 0
      }
    }
  },
  "actions": {
    "send_email": {
      "email": {
        "profile": "gmail",
        "attachments": {
          "dashboard.pdf": {
            "reporting": {
              "url": "http://127.0.0.1:5601/api/reporting/generate/dashboard/AWDIhWBRq-FH7NNrhWqk"
            }
          },
          "data.yml": {
            "data": {
              "format": "yaml"
            }
          }
        },
        "to": [
          "<insert_email_address>”
        ],
        "subject": "[WATCHER] Process Changed File Creation Time",
        "body": {
          "html": "<h1>{{ctx.payload.hits.total}} File Creation Times Changed In Past 1 Minute</h1>  <h3>File Creation Times Changed (timestamp --- file --- process):</h3>  <ol> {{#ctx.payload.hits.hits}} <li>{{_source.event_data.UtcTime}} --- {{_source.event_data.TargetFilename}} --- {{_source.event_data.Image}}</li> {{/ctx.payload.hits.hits}}</ol><h3>Elastic Discover Query:</h3><a href='http://127.0.0.1:5601/app/kibana#/discover/AWDImY92q-FH7NNrhYSC'>http://127.0.0.1:5601/app/kibana#/discover/AWDImY92q-FH7NNrhYSC</a>"
        }
      }
    }
  }
}
```

APPENDIX Q: WATCHER ALERT FOR FILES CREATED WITHIN SHORT TIMEFRAME

This research created a Watcher alert for reporting when more than 240 files are created on a system in less than 1 minute. The watcher code can be found online at `https://github.com/dsugraduate/dsu2018/`. The code is included below, with line numbers displayed for reference.

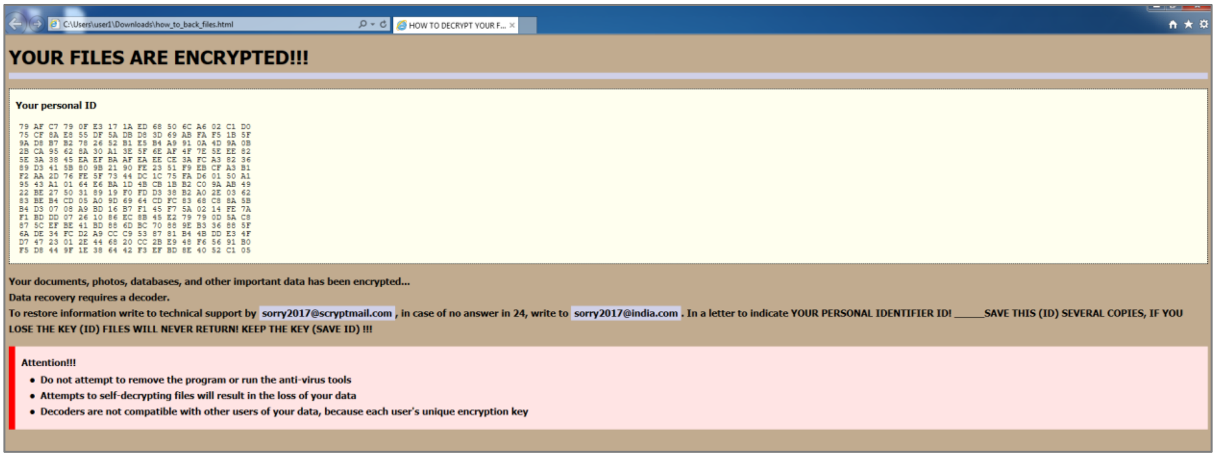
```
1   {
2     "trigger": {
3       "schedule": {
4         "interval": "1m"
5       }
6     },
7     "input": {
8       "search": {
9         "request": {
10          "search_type": "query_then_fetch",
11          "indices": [
12            "<winlogbeat-{now-2m}>",
13            "<winlogbeat-{now}>"
14          ],
15          "types": [],
16          "body": {
17            "from": 0,
18            "size": 10000,
19            "query": {
20              "bool": {
21                "filter": [
22                  {
23                    "range": {
24                      "@timestamp": {
25                        "gte": "now-1m"
26                      }
27                    }
28                  },
29                  {
30                    "term": {
31                      "event_id": 11
32                    }
33                  }
34                ]
35              }
36            },
37            "sort": [
38              {
39                "event_data.UtcTime.keyword": "asc"
40              }
```

```
          ]
        }
      }
    }
  },
  "condition": {
    "compare": {
      "ctx.payload.hits.total": {
        "gt": 240
      }
    }
  },
  "actions": {
    "send_email": {
      "email": {
        "profile": "gmail",
        "attachments": {
          "dashboard.pdf": {
            "reporting": {
              "url":
"http://127.0.0.1:5601/api/reporting/generate/dashboard/AWDIhv9bq-FH7NNrhWya"
            }
          },
          "data.yml": {
            "data": {
              "format": "yaml"
            }
          }
        },
        "to": [
          "<insert_email_address>"
        ],
        "subject": "[WATCHER] File Creation Threshold Exceeded",
        "body": {
          "html": "<h1>{{ctx.payload.hits.total}} Files Created In Past
1 Minute</h1>  <h3>Files Created (timestamp --- file):</h3>  <ol>
{{#ctx.payload.hits.hits}} <li>{{_source.event_data.UtcTime}} ---
{{_source.event_data.TargetFilename}}</li>
{{/ctx.payload.hits.hits}}</ol><h3>Elastic Discover Query:</h3><a
href='http://127.0.0.1:5601/app/kibana#/discover/AWDPHZZyvSdcMfsiRh6s'>http
://127.0.0.1:5601/app/kibana#/discover/AWDPHZZyvSdcMfsiRh6s</a>"
        }
      }
    }
  }
}
```

APPENDIX R. SCREENSHOTS OF RANSOMWARE NOTIFICATION MESSAGES

This appendix contains screenshots of the ransomware notification messages observed after executing each ransomware sample. The screenshots are included below so that other researchers know the effects of each sample without having to execute the samples in their own environments.

GlobeImposter sample G1 used file extension "`.crypt`" and created notification file "`how_to_back_files.html`". A screenshot of the notification file is included in Figure 20.



YOUR FILES ARE ENCRYPTED!!!

Your personal ID

```
79 AF C7 79 0F E3 17 1A ED 68 50 6C A6 02 C1 D0
75 CF 8A E8 55 DF 5A DB D8 3D 69 AB FA F5 1B 5F
9A D8 B7 B2 78 26 52 B1 E5 B4 A9 91 0A 4D 9A 0B
2B CA 95 62 8A 30 A1 3E 5F 6E AF 4F 7E 5E EE 82
5E 3A 38 45 EA EF BA AF EA EE CE 3A FC A3 82 36
89 D3 41 5B 80 9B 21 90 FE 23 51 F9 EB CF A3 B1
F2 AA 2D 76 FE 5F 73 44 DC 1C 75 FA D6 01 50 A1
95 43 A1 01 64 E6 BA 1D 4B CB 1B B2 C0 9A AB 49
22 BE 27 50 31 89 19 F0 FD D3 38 B2 A0 2E 03 62
83 BE B4 CD 05 A0 9D 69 64 CD FC 83 68 C8 8A 5B
B4 D3 07 08 A9 BD 16 B7 F1 45 F7 5A 02 14 FE 7A
F1 BD DD 07 26 10 86 EC 8B 45 E2 79 79 0D 5A C8
87 5C EF BE 41 BD 88 6D BC 70 88 9E B3 36 88 5F
6A DE 34 FC D2 A9 CC C9 53 87 81 B4 4B DD E3 4F
D7 47 23 01 2E 44 68 20 CC 2B E9 48 F6 56 91 B0
F5 D8 44 9F 1E 38 64 42 F3 EF BD 8E 40 52 C1 05
```

Your documents, photos, databases, and other important data has been encrypted...

Data recovery requires a decoder.

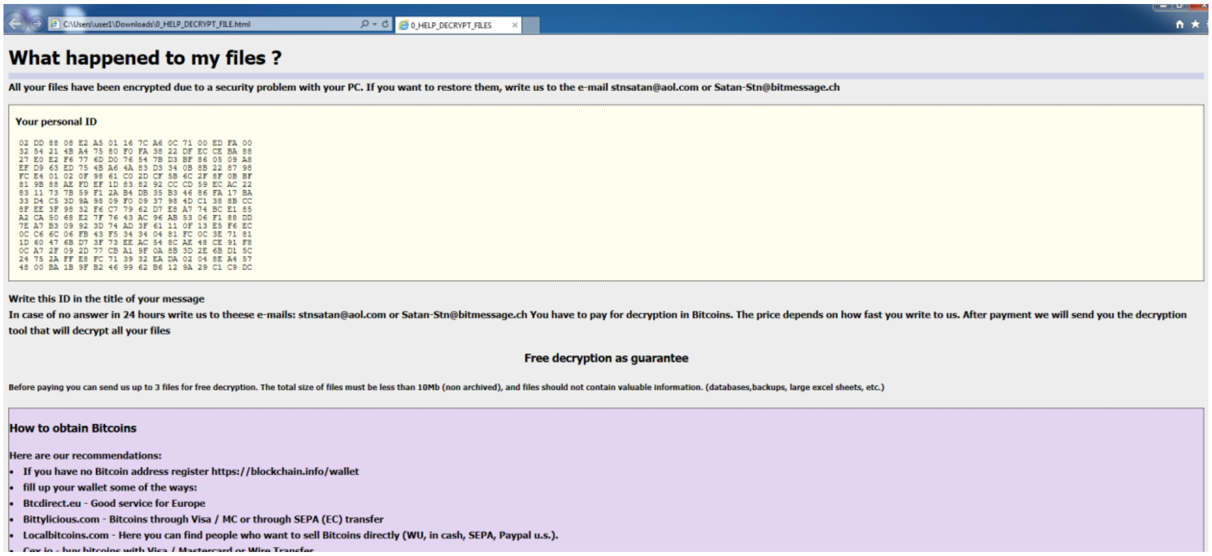
To restore information write to technical support by sorry2017@scryptmail.com , in case of no answer in 24, write to sorry2017@india.com . In a letter to indicate YOUR PERSONAL IDENTIFIER ID! _____SAVE THIS (ID) SEVERAL COPIES, IF YOU LOSE THE KEY (ID) FILES WILL NEVER RETURN! KEEP THE KEY (SAVE ID) !!!

Attention!!!

- Do not attempt to remove the program or run the anti-virus tools
- Attempts to self-decrypting files will result in the loss of your data
- Decoders are not compatible with other users of your data, because each user's unique encryption key

Figure 20. Screenshot of Ransomware Sample G1

GlobeImposter sample G2 used file extension "`.STN`" and created notification file "`0_HELP_DECYRPT_FILE.html`". A screenshot of the notification file is included in Figure 21.



What happened to my files ?

All your files have been encrypted due to a security problem with your PC. If you want to restore them, write us to the e-mail stnsatan@aol.com or Satan-Stn@bitmessage.ch

Your personal ID

```
02 DD 88 08 E2 A5 01 16 7C A6 0C 71 00 ED FA 00
32 54 21 4B A4 75 80 F0 FA 38 22 DF EC CE BA 88
27 E0 E2 F6 77 6D D0 76 54 7B D3 BF 86 05 09 A8
EF D9 63 ED 75 4B A6 4A 83 D3 34 0B 8B 22 87 98
FC E4 01 02 0F 98 61 C0 2D CF 5B 6C 2F 8F 0B BF
81 9B 88 AE FD EF 1D 83 82 92 CC CD 59 EC AC 22
83 11 73 7B 59 F1 2A B4 DB 35 B3 46 86 FA 17 BA
33 D4 C5 3D 9A 98 09 F0 09 37 98 4D C1 38 8B CC
8F EE 3F 98 32 F6 C7 79 62 D7 E8 A7 74 BC E1 85
A2 CA 50 68 E2 7F 76 43 AC 96 AB 53 06 F1 88 DD
7E A7 B3 09 92 3D 74 AD 3F 61 11 0F 13 E5 F6 EC
0C C6 6C 06 FB 43 F5 34 34 04 81 FC 0C 3E 71 81
1D 60 47 6B D7 3F 73 EE AC 54 8C AE 48 CE 91 F8
0C A7 2F 09 2D 77 CB A1 9F 0A 8B 3D 2E 6B D1 5C
24 75 2A FF E8 FC 71 39 32 EA DA 02 04 8E A4 57
48 00 BA 1B 9F B2 46 99 62 B6 12 9A 29 C1 C9 DC
```

Write this ID in the title of your message
In case of no answer in 24 hours write us to theese e-mails: stnsatan@aol.com or Satan-Stn@bitmessage.ch You have to pay for decryption in Bitcoins. The price depends on how fast you write to us. After payment we will send you the decryption tool that will decrypt all your files

Free decryption as guarantee

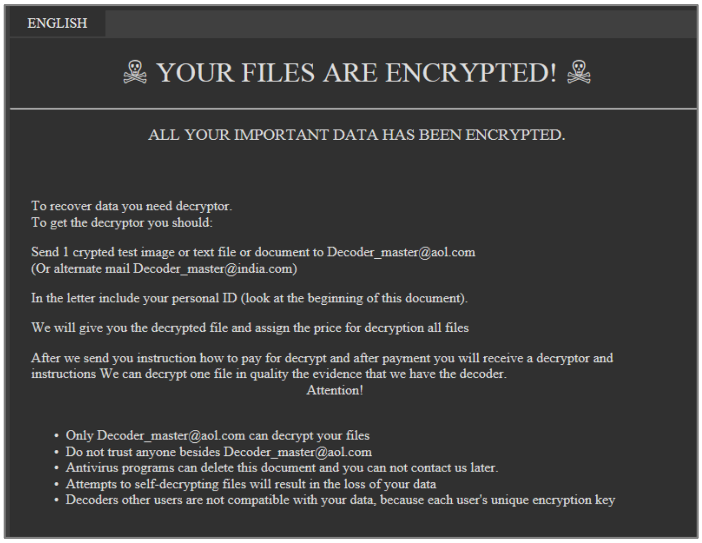
Before paying you can send us up to 3 files for free decryption. The total size of files must be less than 10Mb (non archived), and files should not contain valuable information. (databases,backups, large excel sheets, etc.)

How to obtain Bitcoins

Here are our recommendations:
- If you have no Bitcoin address register https://blockchain.info/wallet
- fill up your wallet some of the ways:
- Btcdirect.eu - Good service for Europe
- Bittylicious.com - Bitcoins through Visa / MC or through SEPA (EC) transfer
- Localbitcoins.com - Here you can find people who want to sell Bitcoins directly (WU, in cash, SEPA, Paypal u.s.).
- Cex.io - buy bitcoins with Visa / Mastercard or Wire Transfer

Figure 21. Screenshot of Ransomware Sample G2

GlobeImposter sample G3 used file extension "`.coded`" and created notification file "`how_to_back_files.html`". A screenshot of the notification file is included in Figure 21.



☠ YOUR FILES ARE ENCRYPTED! ☠

ALL YOUR IMPORTANT DATA HAS BEEN ENCRYPTED.

To recover data you need decryptor.
To get the decryptor you should:

Send 1 crypted test image or text file or document to Decoder_master@aol.com
(Or alternate mail Decoder_master@india.com)

In the letter include your personal ID (look at the beginning of this document).

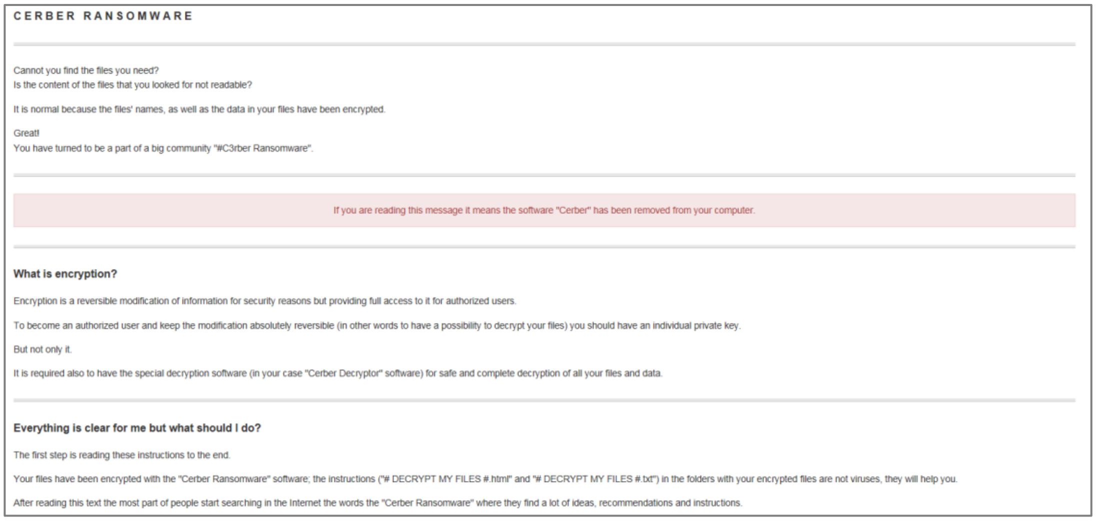
We will give you the decrypted file and assign the price for decryption all files

After we send you instruction how to pay for decrypt and after payment you will receive a decryptor and instructions We can decrypt one file in quality the evidence that we have the decoder.
Attention!

- Only Decoder_master@aol.com can decrypt your files
- Do not trust anyone besides Decoder_master@aol.com
- Antivirus programs can delete this document and you can not contact us later.
- Attempts to self-decrypting files will result in the loss of your data
- Decoders other users are not compatible with your data, because each user's unique encryption key

Figure 22. Screenshot of Ransomware Sample G3

Cerber sample C1 used file extension "`.cerber3`" and created notification file "`@__README__@.txt`" and "`@__README__@.html`". A screenshot of the notification file is included in .



CERBER RANSOMWARE

Cannot you find the files you need?
Is the content of the files that you looked for not readable?

It is normal because the files' names, as well as the data in your files have been encrypted.

Great!
You have turned to be a part of a big community "#C3rber Ransomware".

If you are reading this message it means the software "Cerber" has been removed from your computer.

What is encryption?

Encryption is a reversible modification of information for security reasons but providing full access to it for authorized users.

To become an authorized user and keep the modification absolutely reversible (in other words to have a possibility to decrypt your files) you should have an individual private key.

But not only it.

It is required also to have the special decryption software (in your case "Cerber Decryptor" software) for safe and complete decryption of all your files and data.

Everything is clear for me but what should I do?

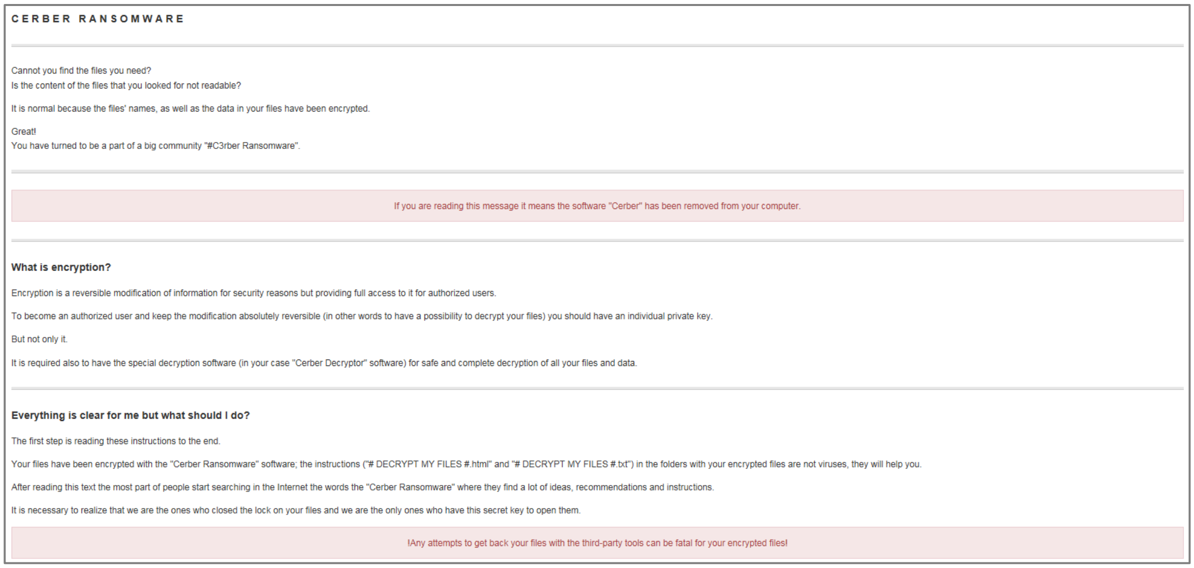
The first step is reading these instructions to the end.

Your files have been encrypted with the "Cerber Ransomware" software; the instructions ("# DECRYPT MY FILES #.html" and "# DECRYPT MY FILES #.txt") in the folders with your encrypted files are not viruses, they will help you.

After reading this text the most part of people start searching in the Internet the words the "Cerber Ransomware" where they find a lot of ideas, recommendations and instructions.

Figure 23. Screenshot of Ransomware Sample C1

Cerber sample C2 used file extension "`.cerber3`" and created notification file "`@__README__@.txt`" and "`@__README__@.html`". A screenshot of the notification file is included in .



CERBER RANSOMWARE

Cannot you find the files you need?
Is the content of the files that you looked for not readable?

It is normal because the files' names, as well as the data in your files have been encrypted.

Great!
You have turned to be a part of a big community "#C3rber Ransomware".

If you are reading this message it means the software "Cerber" has been removed from your computer.

What is encryption?

Encryption is a reversible modification of information for security reasons but providing full access to it for authorized users.

To become an authorized user and keep the modification absolutely reversible (in other words to have a possibility to decrypt your files) you should have an individual private key.

But not only it.

It is required also to have the special decryption software (in your case "Cerber Decryptor" software) for safe and complete decryption of all your files and data.

Everything is clear for me but what should I do?

The first step is reading these instructions to the end.

Your files have been encrypted with the "Cerber Ransomware" software; the instructions ("# DECRYPT MY FILES #.html" and "# DECRYPT MY FILES #.txt") in the folders with your encrypted files are not viruses, they will help you.

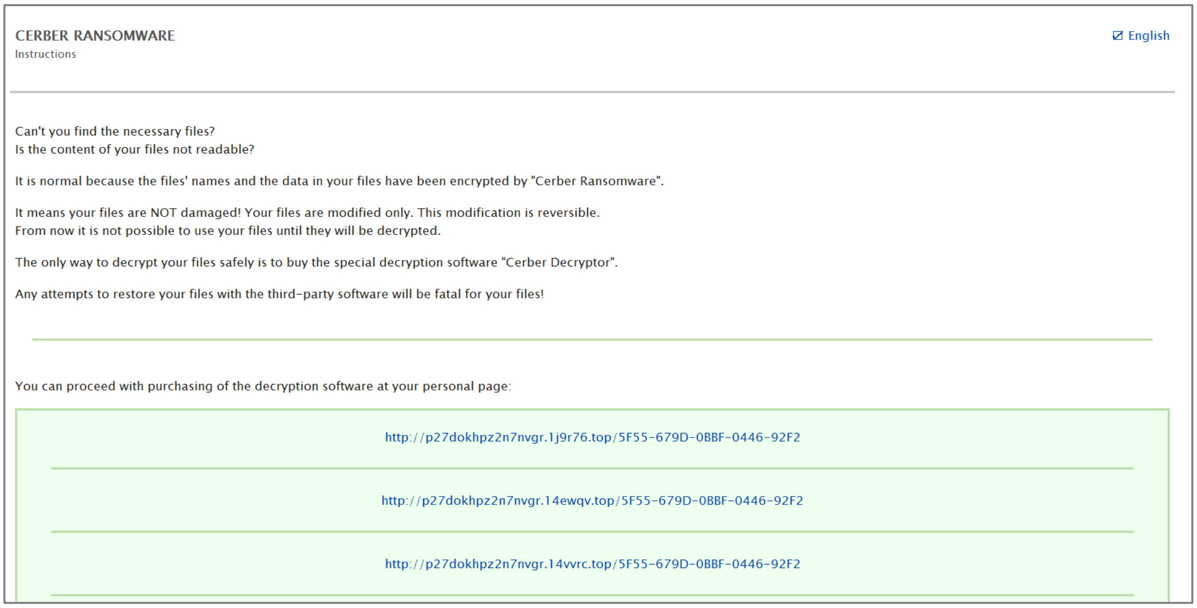
After reading this text the most part of people start searching in the Internet the words the "Cerber Ransomware" where they find a lot of ideas, recommendations and instructions.

It is necessary to realize that we are the ones who closed the lock on your files and we are the only ones who have this secret key to open them.

!Any attempts to get back your files with the third-party tools can be fatal for your encrypted files!

Figure 24. Screenshot of Ransomware Sample C2

Cerber sample C3 used file extension ".8899" and created notification file "R_E_A_D__T_H_I_S_######.txt" and "R_E_A_D__T_H_I_S_######.hta", where ###### is a random number. A screenshot of the notification file is included in Figure 25.



CERBER RANSOMWARE
Instructions

Can't you find the necessary files?
Is the content of your files not readable?

It is normal because the files' names and the data in your files have been encrypted by "Cerber Ransomware".

It means your files are NOT damaged! Your files are modified only. This modification is reversible.
From now it is not possible to use your files until they will be decrypted.

The only way to decrypt your files safely is to buy the special decryption software "Cerber Decryptor".

Any attempts to restore your files with the third-party software will be fatal for your files!

You can proceed with purchasing of the decryption software at your personal page:

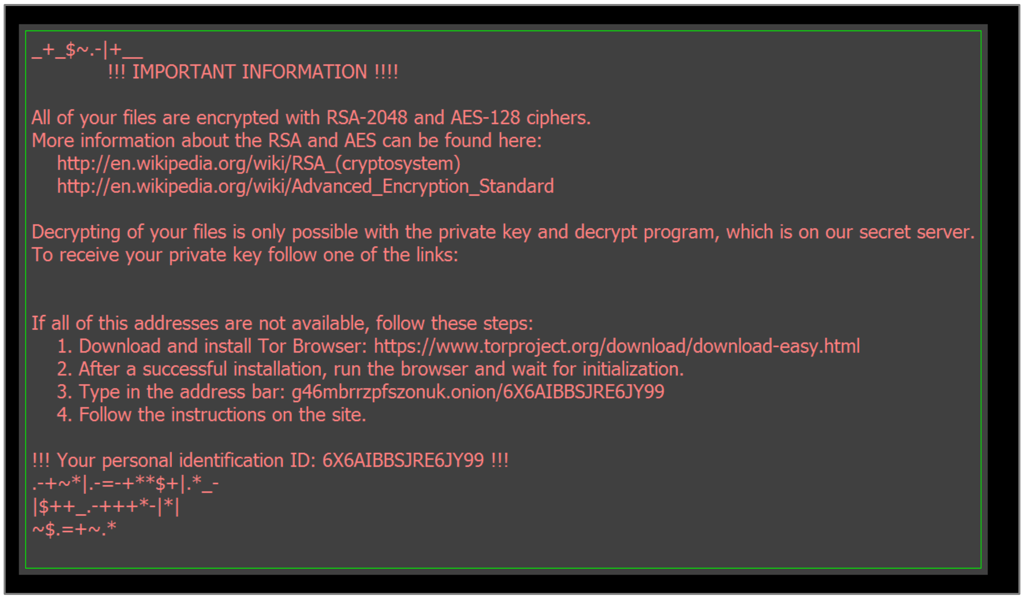
http://p27dokhpz2n7nvgr.1j9r76.top/5F55-679D-0BBF-0446-92F2

http://p27dokhpz2n7nvgr.14ewqv.top/5F55-679D-0BBF-0446-92F2

http://p27dokhpz2n7nvgr.14vvrc.top/5F55-679D-0BBF-0446-92F2

Figure 25. Screenshot of Ransomware Sample C3

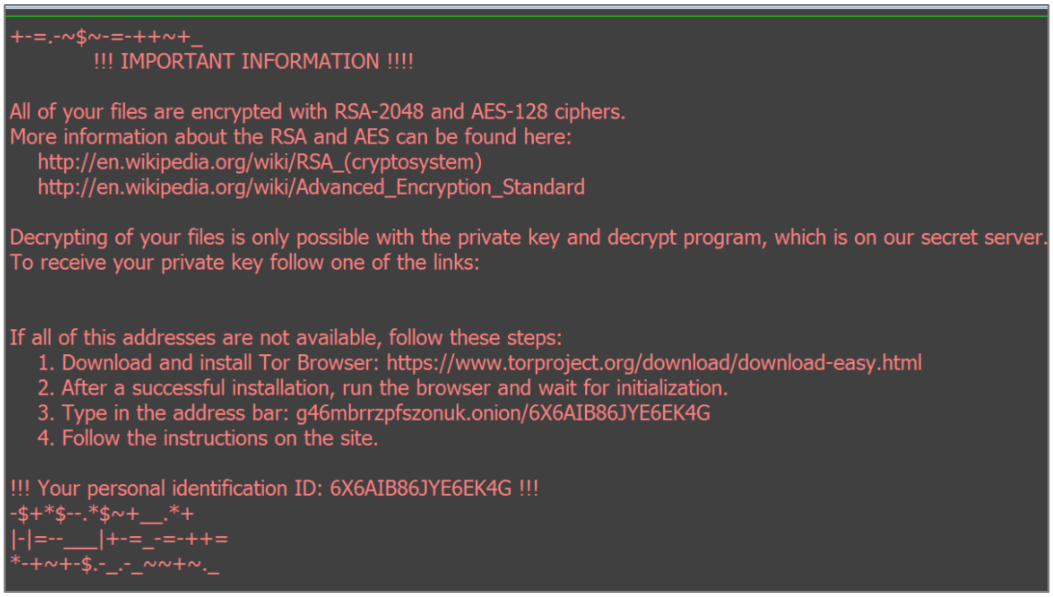
Locky sample L1 used file extension ".asasin" and created notification files "asasin.html" and "asasin.bmp". A screenshot of the notification file is included in Figure 26.



```
_+_$~.-|+__
          !!! IMPORTANT INFORMATION !!!!

All of your files are encrypted with RSA-2048 and AES-128 ciphers.
More information about the RSA and AES can be found here:
    http://en.wikipedia.org/wiki/RSA_(cryptosystem)
    http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Decrypting of your files is only possible with the private key and decrypt program, which is on our secret server.
To receive your private key follow one of the links:


If all of this addresses are not available, follow these steps:
    1. Download and install Tor Browser: https://www.torproject.org/download/download-easy.html
    2. After a successful installation, run the browser and wait for initialization.
    3. Type in the address bar: g46mbrrzpfszonuk.onion/6X6AIBBSJRE6JY99
    4. Follow the instructions on the site.

!!! Your personal identification ID: 6X6AIBBSJRE6JY99 !!!
.-+~*|.-=-+**$+|.*_-
|$++_.-+++*-|*|
~$.=+~.*
```

Figure 26. Screenshot of Ransomware Sample L1

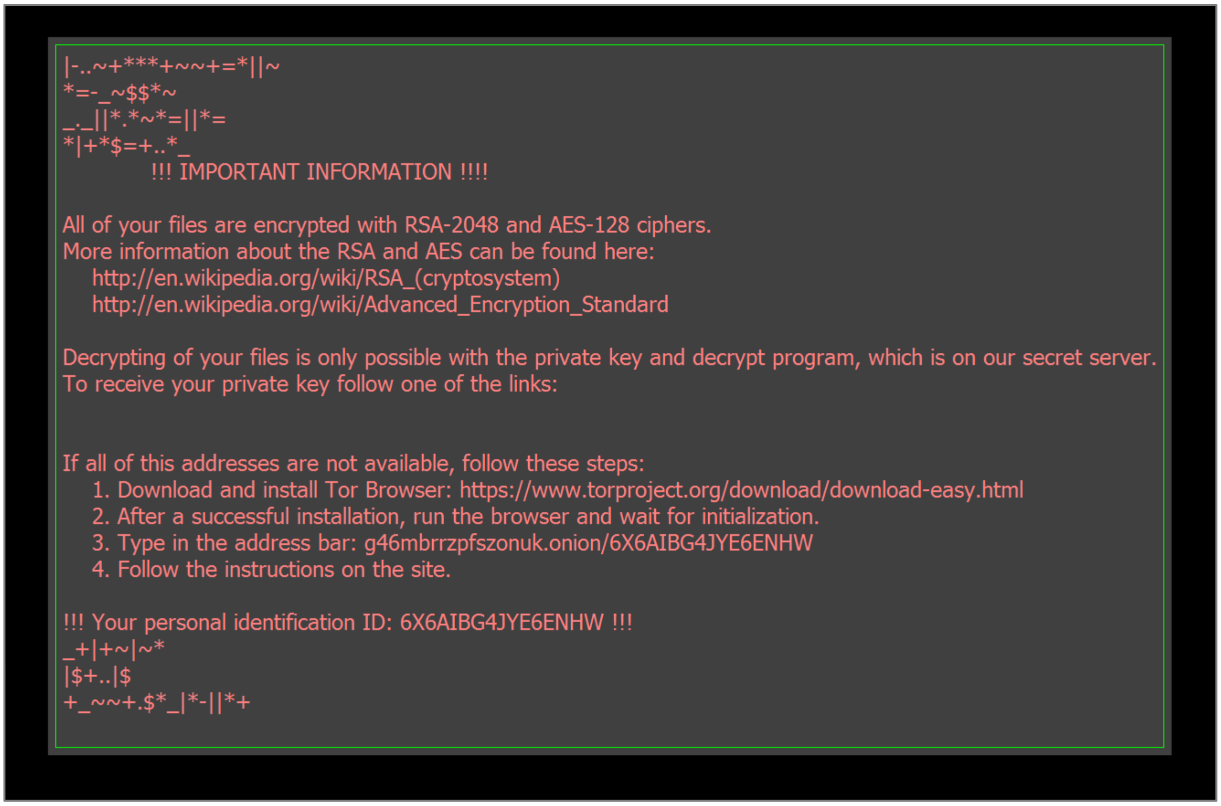
Locky sample L2 used file extension "`.asasin`" and created notification files "`asasin.html`" and "`asasin.bmp`". A screenshot of the notification file is included in Figure 27.



```
+-=.-~$~-=-++~+_
          !!! IMPORTANT INFORMATION !!!!

All of your files are encrypted with RSA-2048 and AES-128 ciphers.
More information about the RSA and AES can be found here:
    http://en.wikipedia.org/wiki/RSA_(cryptosystem)
    http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Decrypting of your files is only possible with the private key and decrypt program, which is on our secret server.
To receive your private key follow one of the links:


If all of this addresses are not available, follow these steps:
    1. Download and install Tor Browser: https://www.torproject.org/download/download-easy.html
    2. After a successful installation, run the browser and wait for initialization.
    3. Type in the address bar: g46mbrrzpfszonuk.onion/6X6AIB86JYE6EK4G
    4. Follow the instructions on the site.

!!! Your personal identification ID: 6X6AIB86JYE6EK4G !!!
-$+*$--.*$~+__.*+
|-|=--___|+-=_-=-++=
*-+~+-$.-_.-_~~+~._
```

Figure 27. Screenshot of Ransomware Sample L2

Locky sample L3 used file extension "`.asasin`" and created notification files "`asasin.html`" and "`asasin.bmp`". A screenshot of the notification file is included in Figure 28.



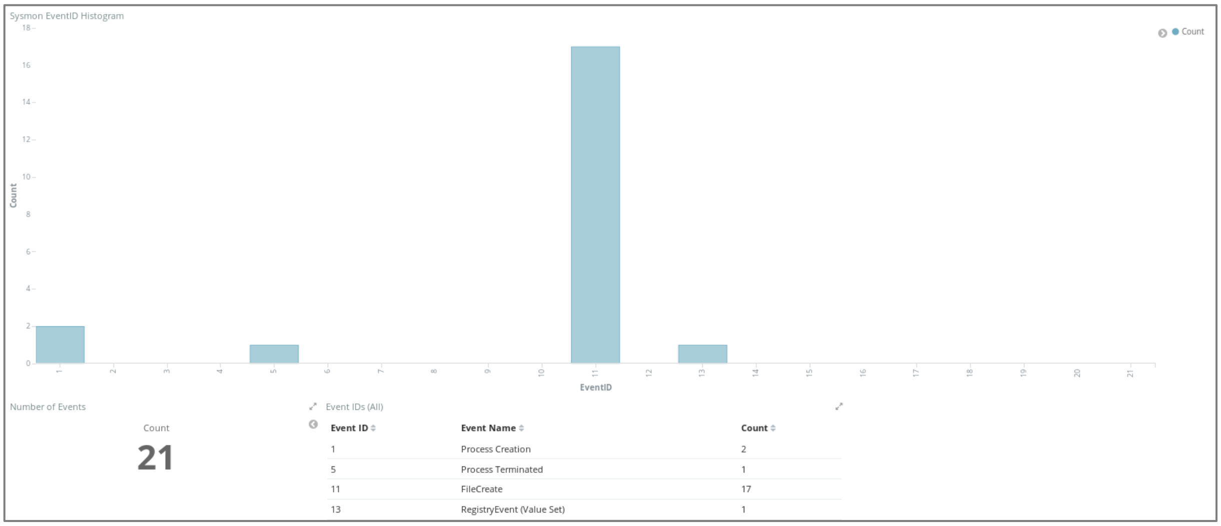
```
|-..~+***+~~+=*||~
*=-_~$$*~
_._||*.*~*=||*=
*|+*$=+..*_
         !!! IMPORTANT INFORMATION !!!!

All of your files are encrypted with RSA-2048 and AES-128 ciphers.
More information about the RSA and AES can be found here:
   http://en.wikipedia.org/wiki/RSA_(cryptosystem)
   http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Decrypting of your files is only possible with the private key and decrypt program, which is on our secret server.
To receive your private key follow one of the links:


If all of this addresses are not available, follow these steps:
   1. Download and install Tor Browser: https://www.torproject.org/download/download-easy.html
   2. After a successful installation, run the browser and wait for initialization.
   3. Type in the address bar: g46mbrrzpfszonuk.onion/6X6AIBG4JYE6ENHW
   4. Follow the instructions on the site.

!!! Your personal identification ID: 6X6AIBG4JYE6ENHW !!!
_+|+~|~*
|$+..|$
+_~~+.$*_|*-||*+
```

Figure 28. Screenshot of Ransomware Sample L3

APPENDIX S. HISTOGRAMS OF RANSOMWARE SAMPLES

This appendix documents the histograms of the Sysmon events triggered by each ransomware sample used in this research study. A description of the ransomware dataset is found in the Chapter 4 on page 20.

The GlobeImposter sample labeled G1 was identified by the SHA-256 hash 7d49a2a9d788fc8dbaa6331c8b740f689e20600ff7e8d3692b1a9c6d37a37bd6 (Hybrid Analysis, 2017g). Sample G1 triggered 21 Sysmon events. The sample created two processes, terminated one process, created 17 files, and set one registry value. The histogram of the Sysmon events are found in Figure 29.



Number of Events

Count

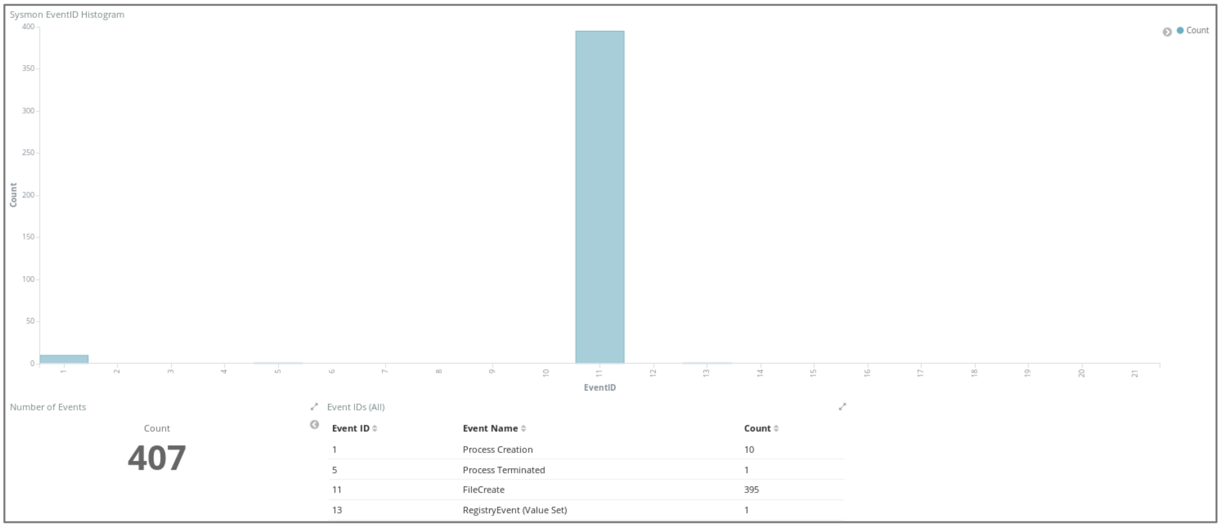
21

Event IDs (All)

Event ID	Event Name	Count
1	Process Creation	2
5	Process Terminated	1
11	FileCreate	17
13	RegistryEvent (Value Set)	1

Figure 29. Histogram of Sysmon Events for Sample G1

The GlobeImposter sample labeled G2 was identified by the SHA-256 hash edf67ba035e52cd903017a24271544caba57dace039be51b1e867fdfd5252744 (Hybrid Analysis, 2017b). Sample G2 triggered 407 Sysmon events. The sample created 10 processes, terminated one process, created 395 files, and set one registry value. The histogram of the Sysmon events generated by sample G2 is found in Figure 30.



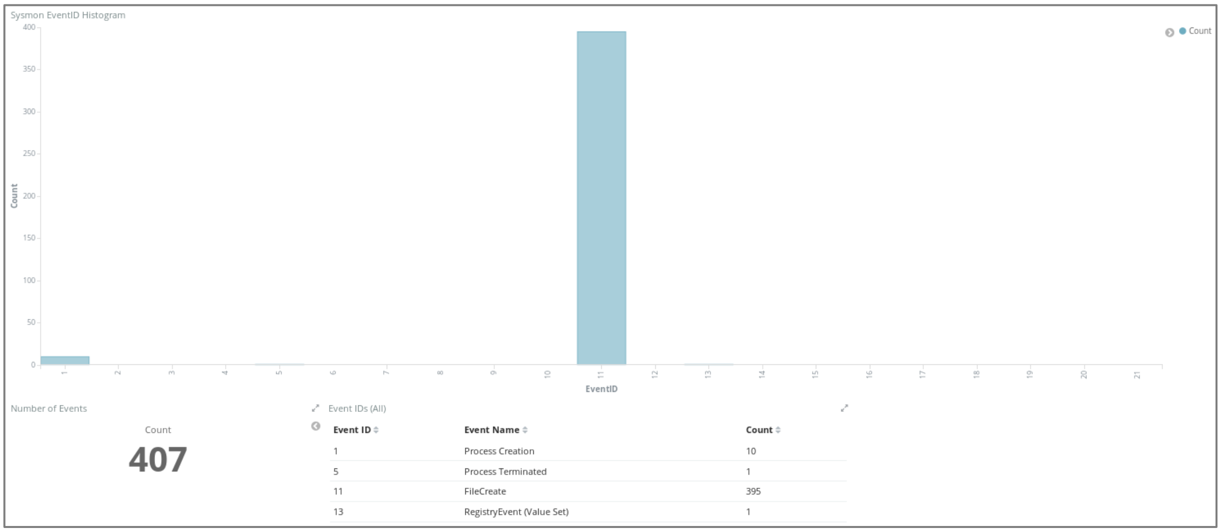
Count

407

Event ID	Event Name	Count
1	Process Creation	10
5	Process Terminated	1
11	FileCreate	395
13	RegistryEvent (Value Set)	1

Figure 30. Histogram of Sysmon Events for Sample G2

The GlobeImposter sample labeled G3 was identified by the SHA-256 hash b2282de3df95c6a9d0151ad61d2ab4e99400ca3104ce9003a0b13290260a7a55 (Hybrid Analysis, 2017c). Sample G3 triggered 407 Sysmon events. The sample created 10 processes, terminated one process, created 395 files, and set one registry value. The histogram of the Sysmon events generated by sample G3 is found in Figure 31.



Number of Events

Count

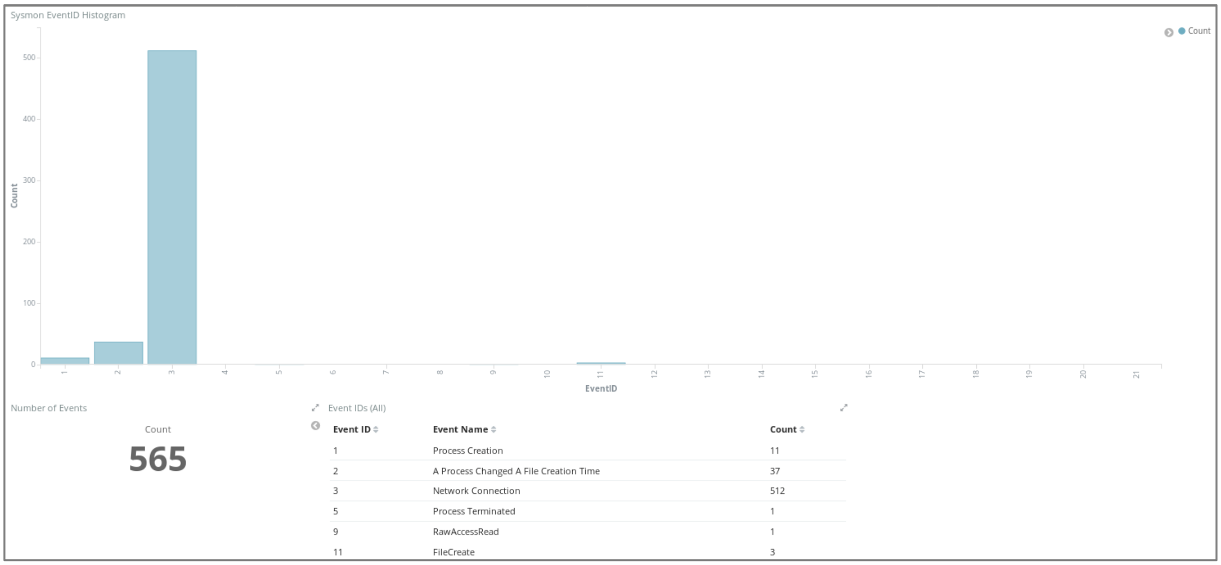
407

Event IDs (All)

Event ID	Event Name	Count
1	Process Creation	10
5	Process Terminated	1
11	FileCreate	395
13	RegistryEvent (Value Set)	1

Figure 31. Histogram of Sysmon Events for Sample G3

The Cerber sample labeled C1 was identified by the SHA-256 hash 6563059c4e556e2bc1589b9711a328c4499baed3b0a14b533a467ce65ee37af6 (Hybrid Analysis, 2017i). Sample C1 triggered 565 Sysmon events. The sample created 11 processes, changed the creation time of 37 files, created 512 network connections, terminated one process, accessed one raw disk, and created three files. The histogram of the Sysmon events generated by sample C1 is found in Figure 32.



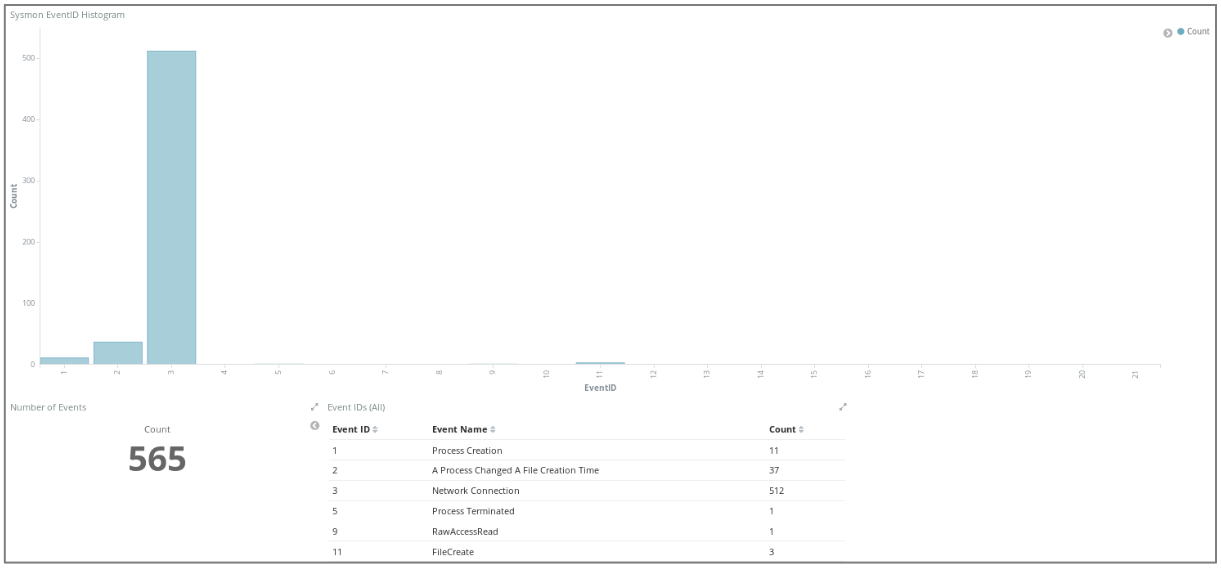
Count

565

Event ID	Event Name	Count
1	Process Creation	11
2	A Process Changed A File Creation Time	37
3	Network Connection	512
5	Process Terminated	1
9	RawAccessRead	1
11	FileCreate	3

Figure 32. Histogram of Sysmon Events for Sample C1

The Cerber sample labeled C2 was identified by the SHA-256 hash 403577074344d4832649881daf8885fed4d9afc3e7a4b02247ceb9b51d858794 (Hybrid Analysis, 2017h). Sample C2 triggered 565 Sysmon events. The sample created 11 processes, changed the creation time of 37 files, created 512 network connections, terminated one process, accessed one raw disk, and created three files. The histogram of the Sysmon events generated by sample C2 is found in Figure 33. A keen observer will notice that this is the exact same profile as sample C1.



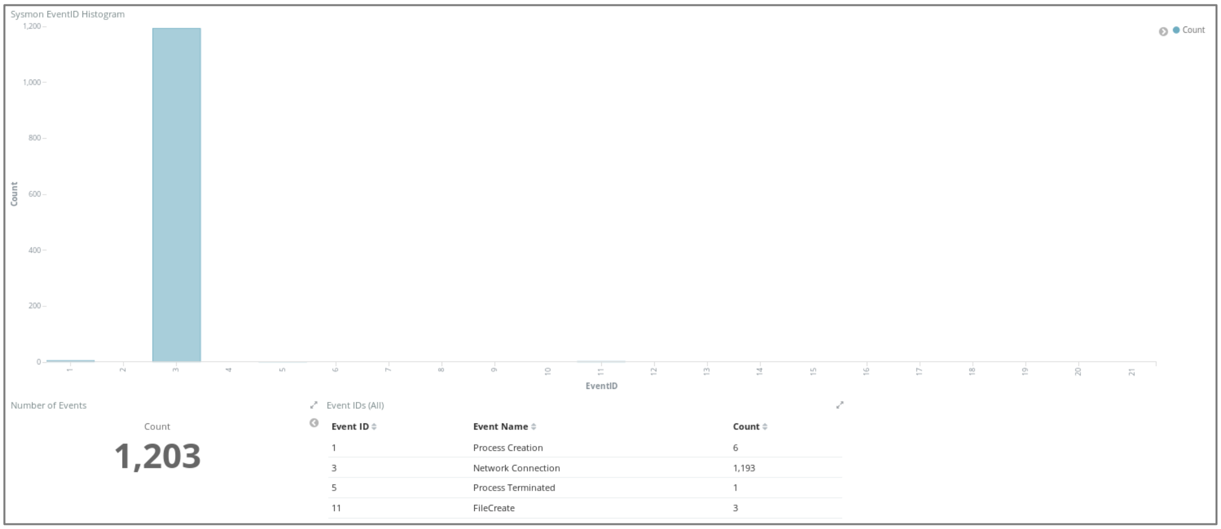
Count

565

Event ID	Event Name	Count
1	Process Creation	11
2	A Process Changed A File Creation Time	37
3	Network Connection	512
5	Process Terminated	1
9	RawAccessRead	1
11	FileCreate	3

Figure 33. Histogram of Sysmon Events for Sample C2

The Cerber sample labeled C3 was identified by the SHA-256 hash e67834d1e8b38ec5864cfa101b140aeaba8f1900a6e269e6a94c90fcbfe56678 (Hybrid Analysis, 2017a). Sample C3 triggered 1203 Sysmon events. The sample created six processes, created 1,193 network connections, terminated one process, and created three files. The histogram of the Sysmon events generated by sample C3 is found in Figure 34.



Number of Events

Count

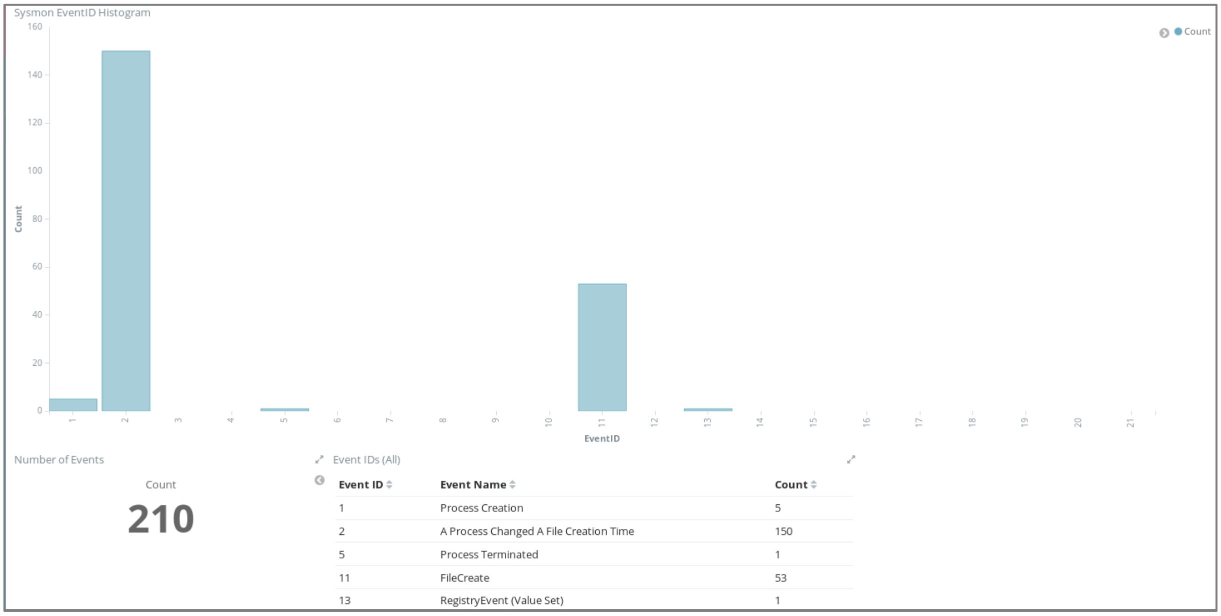
1,203

Event IDs (All)

Event ID	Event Name	Count
1	Process Creation	6
3	Network Connection	1,193
5	Process Terminated	1
11	FileCreate	3

Figure 34. Histogram of Sysmon Events for Sample C3

The Locky sample labeled L1 was identified by the SHA-256 hash c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f (Hybrid-Analysis, 2017l). Sample L1 triggered 210 Sysmon events. The sample created five processes, changed the creation time on 150 files, terminated one process, and created 53 files, and set one registry value. The histogram of the Sysmon events generated by sample L1 is found in Figure 35.



Number of Events

Count

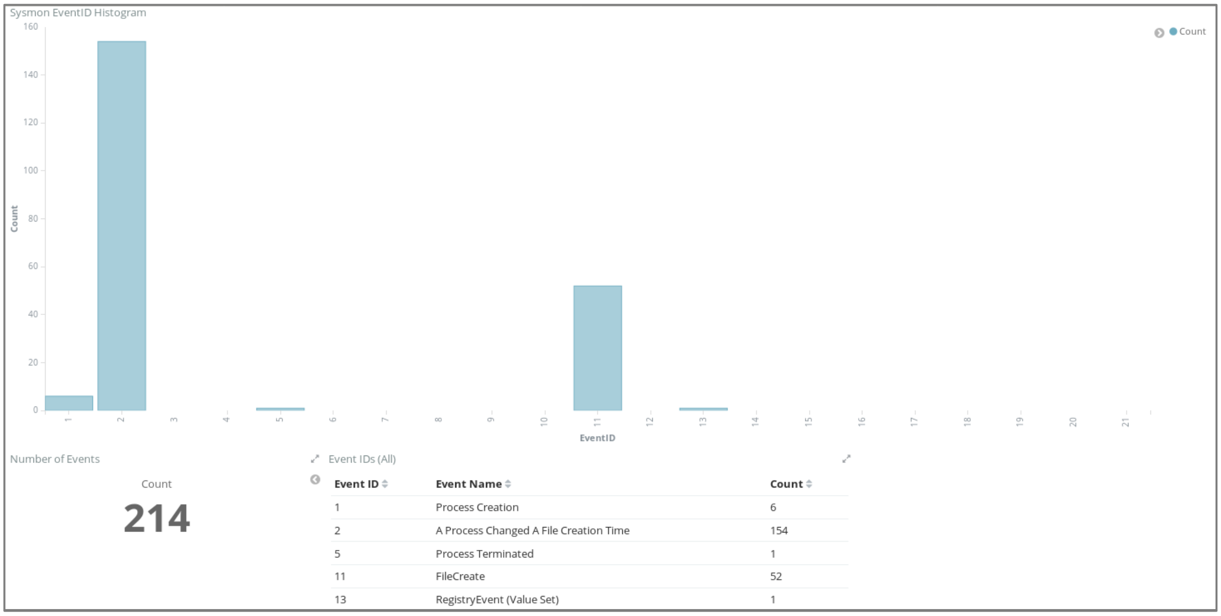
210

Event IDs (All)

Event ID	Event Name	Count
1	Process Creation	5
2	A Process Changed A File Creation Time	150
5	Process Terminated	1
11	FileCreate	53
13	RegistryEvent (Value Set)	1

Figure 35. Histogram of Sysmon Events for Sample L1

The Locky sample labeled L2 was identified by the SHA-256 hash 294f55a28930c8afed9b95d2af108a6916eeb2c79967e91f4dde48026bab15ce (Hybrid-Analysis, 2017e). Sample L2 triggered 214 Sysmon events. The sample created six processes, changed the creation time on 154 files, terminated one process, and created 52 files, and set one registry value. The histogram of the Sysmon events generated by sample L2 is found in Figure 36.



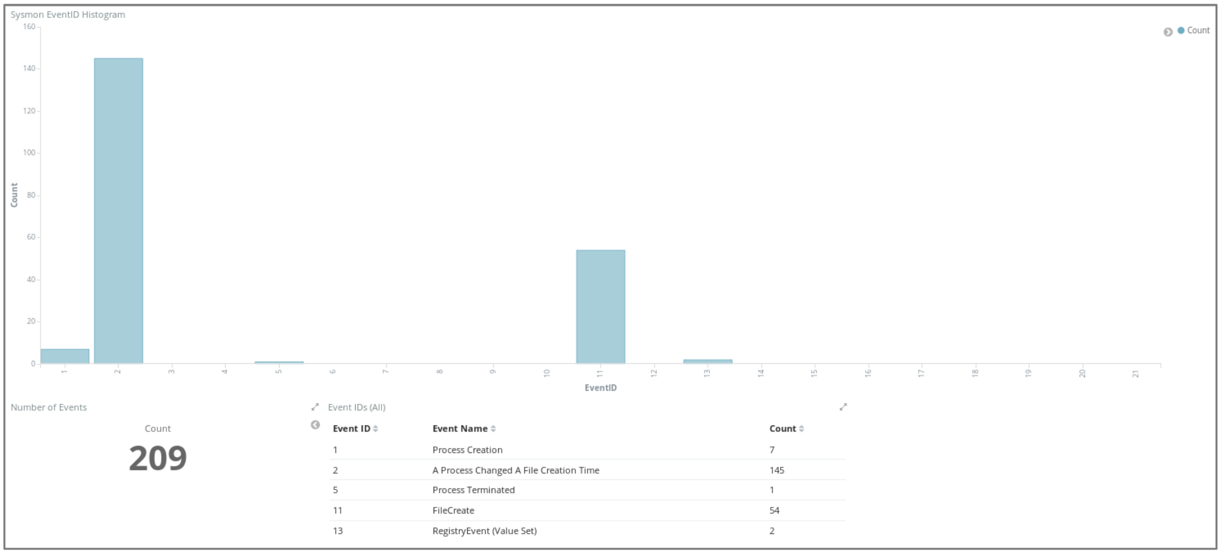
Count

214

Event ID	Event Name	Count
1	Process Creation	6
2	A Process Changed A File Creation Time	154
5	Process Terminated	1
11	FileCreate	52
13	RegistryEvent (Value Set)	1

Figure 36. Histogram of Sysmon Events for Sample L2

The Locky sample labeled L3 was identified by the SHA-256 hash 4c054127056fb400acbab7825aa2754942121e6c49b0f82ae20e65422abdee4f (Hybrid-Analysis, 2017d). Sample L3 triggered 209 Sysmon events. The sample created 7 processes, changed the creation time on 150 files, terminated one process, and created 54 files, and set two registry values. The histogram of the Sysmon events generated by sample L3 is found in Figure 37.



Number of Events

Count

209

Event IDs (All)

Event ID	Event Name	Count
1	Process Creation	7
2	A Process Changed A File Creation Time	145
5	Process Terminated	1
11	FileCreate	54
13	RegistryEvent (Value Set)	2

Figure 37. Histogram of Sysmon Events for Sample L3