1-8-2019

Multi-Criteria Selection of Capability-Based Cybersecurity Solutions

Thomas H. Llansó
Dakota State University

Martha Wagner McNeil
Dakota State University

Cherie Noteboom
Dakota State University

Proceedings of the 52nd Hawaii International Conference on System Sciences | 2019

Multi-Criteria Selection of Capability-Based Cybersecurity Solutions

Thomas Llansó
Johns Hopkins University
Applied Physics Laboratory
thomas.llanso@jhuapl.edu

Martha McNeil
Johns Hopkins University
Applied Physics Laboratory
martha.mcneil@jhuapl.edu

Cherie Noteboom
Dakota State University
cheri.noteboom@dsu.edu

Abstract

Given the increasing frequency and severity of cyber attacks on information systems of all kinds, there is interest in rationalized approaches for selecting the "best" set of cybersecurity mitigations. However, what is best for one target environment is not necessarily best for another. This paper examines an approach to selection that uses a set of weighted criteria, where the security engineer sets the weights based on organizational priorities and constraints. The approach is based on a capability-based representation for cybersecurity mitigations. The paper discusses a group of artifacts that compose the approach through the lens of Design Science research and reports performance results of an instantiation artifact.

1. Introduction

Cyber systems are ubiquitous across society. Breaches to cyber systems continue to be front-page news [1], and, despite more than a decade of heightened focus on cybersecurity, cyber threats continue to evolve and grow [2]–[4]. When threats are insufficiently or incorrectly mitigated based on the anticipated threat, exposure remains. Conversely, over-protection wastes resources and can affect system performance; hence, indiscriminate application of mitigations is ill-advised.

A key method for prioritizing mitigations is to assess the business/mission risks that an organization faces due to the anticipated cyber threat. A number of risk assessment methods are available (e.g., [5]–[12]) that can assist system security engineers (SSEs) in identifying risks in a particular environment. In addition to risk, the SSE must also consider other criteria when making mitigation decisions. Deciding upon, weighting, and quantifying such criteria is a challenge. These decisions are complex, inexact, and involve multiple stakeholders with diverse interests. Moreover, there is no "one size fits all" approach because, for example, information environments, business dependence on those environments, relevant cyber threats, risk tolerance levels, and security budgets vary from one organization to the next [13]. The SSE faces an additional challenge when considering mitigation options: deciding upon those that best balance often-competing criteria (e.g., mitigation cost vs. trustworthiness vs. effectiveness). Such mitigation combinations are often discussed in the context of "Pareto-efficient" solutions. Pareto efficiency is "a state of allocation of resources [e.g., defensive cyber solutions that mitigate threats in this context] from which it is impossible to reallocate so as to make any one individual or preference criterion better off without making at least one individual or preference criterion worse off" [14].

The contribution of this paper is an approach to mitigation selection containing elements of multi-criteria decision-making [15] that recommends a candidate set of defensive solutions using criteria and associated weightings set by the SSE. Primary initial goals are three-fold: (1) identify an approach that we would find useful as practitioners of cybersecurity risk and mitigation analysis, (2) ensure that the approach is compatible with cybersecurity threats and mitigations modeled as capabilities, and (3) identify a practical middle ground between completely ad hoc mitigation selection approaches on the one hand, and approaches whose computational complexity requires the use of sophisticated heuristic algorithms on the other.

The paper is organized as follows: after reviewing related work, we summarize the capability-based representation for cyber threats and defenses against those threats. Next, we use Design Science principles to describe and analyze the artifacts that make up our approach. The description includes a discussion of the underlying object model, associated methods, and an instantiation of the model and methods. Lastly, we evaluate the artifacts with a focus on execution performance for the instantiation, discuss results, and conclude with lessons learned and ideas for future work.

2. Related Work

In practice, it is unrealistic to apply all possible mitigations (also sometimes called security controls) to every threat, due to budget and time pressures, feasi-

URI: https://hdl.handle.net/10125/60169
ISBN: 978-0-9981331-2-6

bility, and other organizational concerns. Several researchers have approached the problem of optimizing mitigation selections, that is, taking a longer list of possible mitigations and down-selecting to a shorter list based on some defined criteria or goals. There are two interesting dimensions to this area, the criteria themselves and the analysis methods.

Dor and Elovici [16] describe a model of information security investment decision-making comprised of concepts that they derive from a grounded theory study. The authors identify great differences in the ways organizations make these decisions influenced by a multitude of criteria, including policy, competitive advantage, financial considerations, quality, compliance, customer expectations, and strategy.

A review of the literature confirms the wide variety of criteria considered when selecting a security control portfolio for a particular situation, including overarching organizational concerns, attributes of specific assets in the environment, anticipated threats, and properties of controls. We summarize these criteria below:

Organizational

- Business impact/disruption, anticipated loss, profit reduction, fines, reputation, decline in stock price, damage [17]–[23]
- Risk tolerance [12], [19], [24]; Budget [19]
- Legal and regulatory [22]
- Self-imposed constraints [22]

Asset

- Importance/value [13], [24]–[27]
- Assessed risk [12], [24]
- Probability of breach, event, or successful attack [13], [24], [26], [28], [29]

Threat

- Anticipated [25], [27], [30], [31]
- Most significant [25]
- Residual risk [23], [32]; Incident data [17]

Control

- Cost, general [12], [13], [30], [32], [18], [20]–[23], [26]–[28]
- Purchase/setup [17], [24], [25], [33]–[35]
- Number of controls as a proxy for cost [36]
- Difficulty of implementation [25]
- Operation, training, and maintenance cost [17], [24], [25], [33], [35]
- Efficiency, effectiveness, performance, degree or number of threats addressed [12], [17], [20], [25], [28], [33], [34]
- Degree of implementation [30]
- Alignment with applicable standards, laws, regulations [33], [34]
- Availability [12]
- Number of benefits accessed [37]
- Controls which, when applied in combination, provide more benefit than the sum of their individual benefits [37]
- Stakeholder preference [31]

Multi-criteria decision-making (MCDM) [15], also known as multiple-criteria decision analysis (MCDA), is widely applied to security portfolio selection [12], [19], [22], [24], [28], [29], [36], [38]. MCDM is a discipline for evaluating multiple conflicting criteria. It is used to analyze problems where there are some measures of costs and benefits that can be traded off to arrive at the best solution under the given constraints. Researchers investigate a number of MCDM techniques for this problem, some of which include or are based on fuzzy set theory [34], multi-attribute utility theory (i.e., value functions, knapsack strategy) [18], [27], [30], [37], evolutionary multi-objective optimization (EMO) also known as genetic algorithms [13], [20], [23], [26], [32], [35], analytic hierarchy process (AHP) [31], grey relational analysis (GRA) [25], simple additive weighting (SAW) [17], the technique for order preference by similarity to ideal solution (TOPSIS) [25], and preference ranking organization method for enrichment evaluation (PROMETHEE) [33].

Several authors apply game theory to security portfolio selection in combination with MCDM techniques. Fielder et al. [30] employs a pure game theoretic approach in a single massive two-person non-cooperative zero-sum static game where the defender (person in charge of choosing defenses) competes against an attacker who chooses among various attack targets. The Nash equilibrium of the game represents the best defensive portfolio. Recognizing that the organization may not have sufficient budget to implement the equilibrium of the pure game, the authors also discuss a hybrid approach combining game theory with a knapsack strategy. Panaousis, et al. [27] model the cybersecurity posture of an organization and then present a series of non-cooperative control-games where each game is between the defender (a single control) and the attacker. The Nash equilibria of the games are derived in consideration of organizational preferences such as costs, anticipated threats, and asset importance. A knapsack approach is subsequently used to optimize investment in security controls within the organization's budget. Finally, Wang and Zhu [21] used evolutionary game theory to investigate long-term cybersecurity investment strategies finding that firms will invest as long as either the cost to invest is low or the cost of a breach is high.

3. Capability-Based Representation

This paper examines defensive solution selection in the context of a capability-based representation for cyber threats and mitigations to those threats [5][39][40]. We define a capability as the ability to contribute in some way to the attack or defense of a target system and a defensive solution as a coordinated set of defensive capabilities. In this approach, the focus is on (1) the underlying offensive capabilities that cyber attackers use to compose attacks and (2) the defensive capabilities composed into defensive solutions that mitigate those offensive capabilities. See the Unified Modeling Language (UML) model [41] in Figure 1 for the basic entities and relationships.

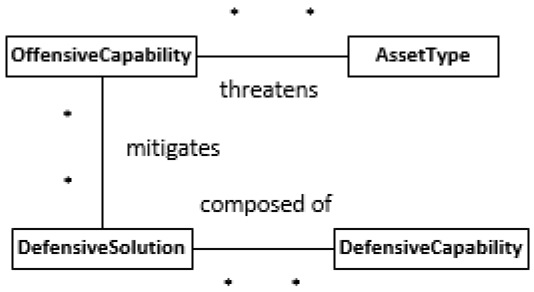


Figure 1: Capability-Based Representation

An example of an offensive capability is "Inject stealthy software implants" and an example of a related defensive capability is "Detect and block most stealthy implants via software whitelisting."

The capability-based approach is in contrast to the historically more common attack-centric approach used in cybersecurity analysis that requires one to enumerate and analyze attack possibilities. We find capability-based analysis more tractable than attack and vulnerability enumeration [42] and justify the approach on the hypothesis that the more one mitigates offensive capabilities possessed by the anticipated adversary, the more difficult it is for the adversary to compose viable attacks from remaining, unmitigated capabilities.

4. Artifacts

This section discusses the artifacts that compose our approach. The artifacts include (1) a model, (2) methods that employ the model to recommend the best potential defensive solutions subject to constraints, and (3) an experimental instantiation (1) and (2).

We examine the artifacts in the context of Design Science (DS) principles as articulated by Peffers, et al., in the paper "A Design Science Research Methodology for Information Systems Research" [43]. Peffers presents a series of steps for artifacts evaluation, specifically: (1) identify the problem and show its importance, (2) define objectives of a solution, (3) design and develop the artifact, (4) demonstrate the artifact in a suitable context, (5) evaluate the effectiveness and efficiency of the artifact, and (6) communicate results. The introduction covered steps 1-2. This section lays artifact design, and we demonstrate and evaluate the artifacts in upcoming sections. Lastly, this paper contributes to the communication requirement.

Model. The UML model in Figure 2 builds on Figure 1 and illustrates an object model used in our experimental prototype implementation.

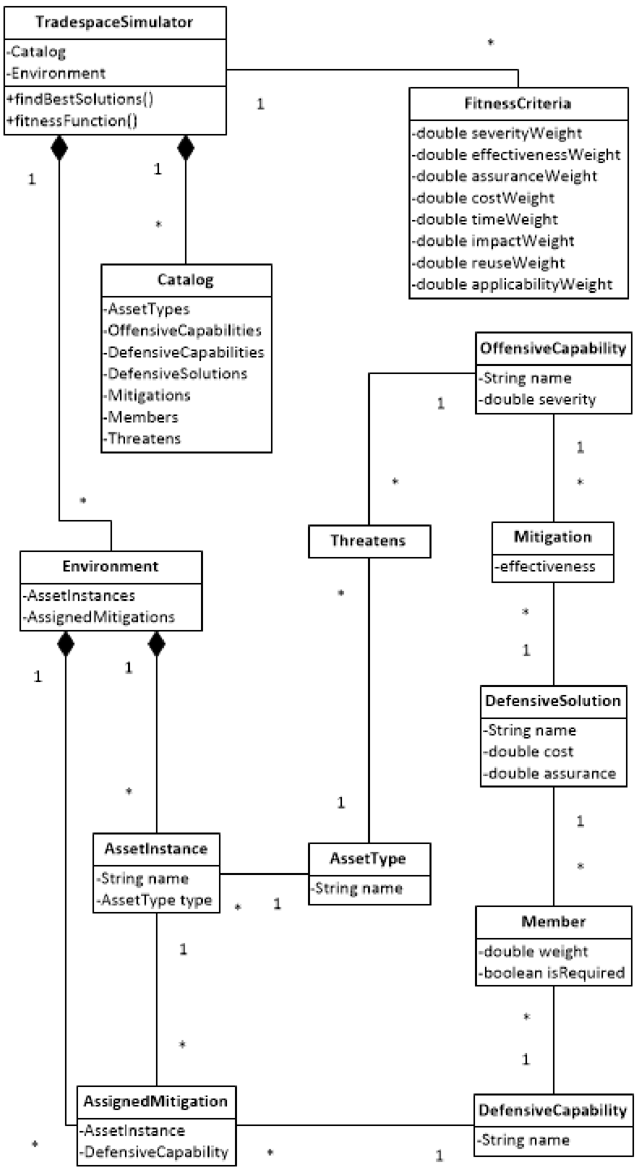


Figure 2: UML Model

Brief descriptions of the classes in Figure 2 appear in Table 1.

Table 1: Class Descriptions

Class	Description
Catalog	A reusable knowledge management repository for capturing the information in Figure 1.
Environment (E)	The asset instances and existing mitigations mapped to them that make up the target environment to be analyzed.
Asset Instance (AI)	An instance of an asset in the target cyber environment. Each asset has a corresponding asset type found in the Catalog.
Assigned Mitigation	A mapping of a defensive capability from the Catalog to an asset instance in the environment.
AssetType (AT)	The type of an asset. Offensive capabilities map to the asset types they threaten.
Defensive Solution (DS)	A collection of defensive capabilities that together can mitigate the effects of one or more offensive capabilities.
Defensive Capability (DC)	The ability to contribute to the mitigation of an offensive capability.
Fitness Criteria	The set of user-selected weights for the fitness criteria.
Member	A mapping of a defensive capability to a given defensive solution.
Mitigation	The mapping of a given defensive solution to an offensive capability that it mitigates.
Offensive Capability (OC)	The ability to contribute to the attack of an asset in some way.
Threatens	A mapping of an offensive capability to a given asset type.
Tradespace Simulator	The main program.

Selection Method. This section uses the abbreviations for classes introduced above to discuss the selection method, called findBestSolutions. The method uses the model given in Figure 2 and an accessory fitness function that scores criteria introduced below.

Evaluation Criteria. To evaluate the fitness of DSs for their role in potentially mitigating OCs, the method employs a set of evaluation criteria shown in Table 2.

Table 2: Defensive Solution Evaluation Criteria

Criterion	Description	Max	Mapping
Severity	The severity of an offensive capability. Severity could be derived, for example, from CVSS scores stored in the National Vulnerability Database [44].	✓	OC
Effectiveness	How effective a DS is believed to be. A value of 0.0 means fully ineffective; 1.0 means fully effective.	✓	DS to OC mapping
Assurance	How trustworthy a given DS is believed to be. A DS might be considered more trustworthy, for example, if it has been rigorously tested by an independent testing laboratory [45].	✓	DS
Cost	An estimate of the total cost of the DS. Cost includes multiple components, such as cost to acquire, integrate, operate, and train users, as applicable.	--	DS
Time	Estimated time to integrate the DS.	--	DS
Impact	Impact to mission/business performance from use of the DS.	--	DS
Reuse	How much of a given DS is already implemented in the environment for a given asset.	✓	DS mapped to AI in E
Applicability	The ratio of the number of offensive capabilities mapped to asset instances in the target environment that the DS can mitigate to the highest number of any DS mapped.	✓	DS mapped to E

Consistent with MCDM, we normalize the range of each criterion to values between 0.0 and 1.0 inclusive. Some criteria, such as cost, time, and reuse, use an ordinal Likert scale mapped to this range. At least initially, subject matter experts (SMEs) set the values for the first six criteria. The findBestSolutions method computes the last two criteria based on E. The long-term goal is to calibrate SME-determined values with empirical reality as such data becomes available.

The Max column in Table 2 indicates whether we wish to maximize (checkmark present) or minimize (checkmark absent) the corresponding criterion. For example, we wish to maximize use of defensive solutions that more effectively address more severe offensive capabilities, whereas we wish to minimize cost, time, and mission impact. The Mapping column in Table 2 indicates the mapping of a criterion into the model in Figure 2. For example, severity is with respect to the effects of an offensive capability (OC) and effectiveness is with respect to the mapping of a defensive solution (DS) to an offensive capability (OC).

We selected the criteria in Table 2 based on a review of the literature and on requirements that stakeholders commonly articulate in our experience. Note that the risk score for each asset in the target environment is not included as a criterion in Table 2 because we use the risk score to filter or down-select the asset instances in the target environment for consideration of mitigations in the first place. That is, the findBestSolutions method only considers the 'riskiest' assets based on a SSE-supplied level of risk tolerance.

Fitness Function. The fitness function, ff, computes a fitness score over the criteria from Table 2 and implements equation (1).

$$ff(ai, oc, ds) = \sum_{i=1}^{n} \left|m_{f_i} - f_i(ai, oc, ds)\right| \cdot fw_i \quad (1)$$

Elements of equation (1) are as follows:

- ff returns a fitness score for a given defensive solution, ds, mapped to a particular offensive capability, oc that, in turn, is mapped to a particular asset instance, ai, in the target environment under consideration. ai maps to a particular AT.
- n is the number of criteria, 8 in this case.
- $f_i(ai, oc, ds)$ is the value of the ith criterion in the context of ai, oc, and ds; 0.0 ≤ value ≤ 1.0.
- fw_i is the weight for the given criterion. The SSE assigns a weight based on relative importance of the criterion in the context of the target environment. Criterion weights are relative to one another and must be non-negative ($fw_i \epsilon \mathbb{Z}^{\geq}$).
- m_{f_i} follows the Max column in Table 2. If the goal is to maximize the given criterion, then m_{f_i} is set to 0.0; otherwise, m_{f_i} is set to 1.0.

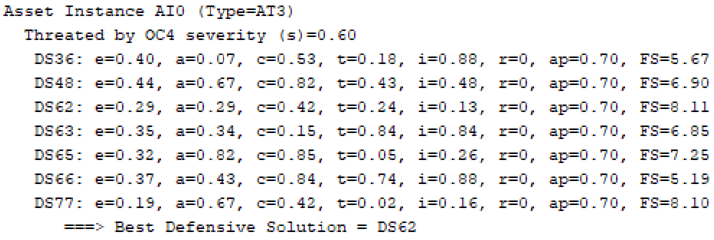
findBestSolutions. findBestSolutions is an implementation of equation (2) and finds the 'best' solutions for the given criteria and associated weightings.

$$findBestSolutions(AI_e, OC, DS) = \underset{ai \in AI_e}{\forall} \; \underset{oc \in OC_{type(ai)}}{\forall} \; \underset{ds \in DS_{oc}}{\forall} ff(ai, oc, ds)$$

(2)

Equation (2) considers each asset instance, ai, in the set of assets instances, AI, in the target environment, e. For each ai, it considers each offensive capability from the set of offensive capabilities, OC, mapped to the asset type, AT, corresponding to the ai. Then for each defensive solution mapped to oc, it applies the fitness function, ff, from equation (1) to ds (in the context of oc and ai).

Instantiation. We call our instantiation of the model and methods described above TradespaceSimulator. To allow us to assess performance, the simulator generates a synthetic sample catalog and a sample target environment using a configurable set of size parameters. Example output from the simulator appears in Figures 3, 4, and 5. Figure 3 is sample output from the findBestSolutions method.

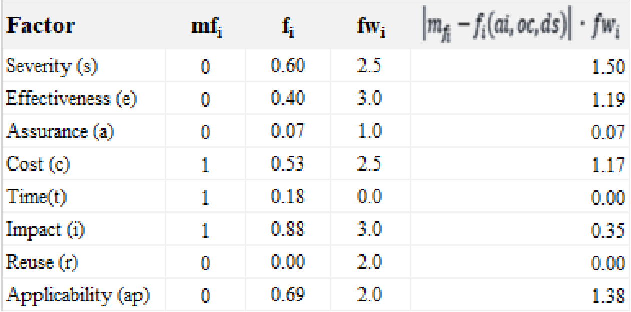


```
Asset Instance AI0 (Type=AT3)
  Threated by OC4 severity (s)=0.60
   DS36: e=0.40, a=0.07, c=0.53, t=0.18, i=0.88, r=0, ap=0.70, FS=5.67
   DS48: e=0.44, a=0.67, c=0.82, t=0.43, i=0.48, r=0, ap=0.70, FS=6.90
   DS62: e=0.29, a=0.29, c=0.42, t=0.24, i=0.13, r=0, ap=0.70, FS=8.11
   DS63: e=0.35, a=0.34, c=0.15, t=0.84, i=0.84, r=0, ap=0.70, FS=6.85
   DS65: e=0.32, a=0.82, c=0.85, t=0.05, i=0.26, r=0, ap=0.70, FS=7.25
   DS66: e=0.37, a=0.43, c=0.84, t=0.74, i=0.88, r=0, ap=0.70, FS=5.19
   DS77: e=0.19, a=0.67, c=0.42, t=0.02, i=0.16, r=0, ap=0.70, FS=8.10
      ===> Best Defensive Solution = DS62
```

Figure 3: Sample output from findBestSolutions

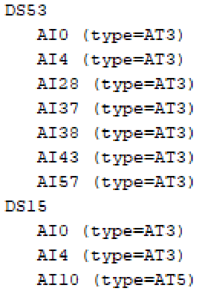
In the Figure 3 sample, the output is for asset instance, AI0, which is of type AT3. The offensive capability threat under consideration is OC4, which has a severity of 0.6. OC4 has seven candidate defensive solutions, each with a fitness score computed by ff. For example, defensive solution DS36 has fitness score of 5.67, which is the sum of the weighted criterion values given in column 5 of Table 3.

Table 3: DS36 score derivation



Factor	**mf$_i$**	**f$_i$**	**fw$_i$**	$\left\|m_{f_i} - f_i(ai, oc, ds)\right\| \cdot fw_i$
Severity (s)	0	0.60	2.5	1.50
Effectiveness (e)	0	0.40	3.0	1.19
Assurance (a)	0	0.07	1.0	0.07
Cost (c)	1	0.53	2.5	1.17
Time(t)	1	0.18	0.0	0.00
Impact (i)	1	0.88	3.0	0.35
Reuse (r)	0	0.00	2.0	0.00
Applicability (ap)	0	0.69	2.0	1.38

Figure 4 shows a small sampling of the best solutions resulting from application of the findBestSolutions method and the asset instances to which they apply.

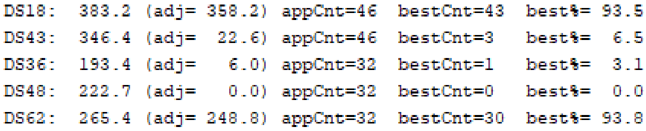


```
DS53
    AI0 (type=AT3)
    AI4 (type=AT3)
    AI28 (type=AT3)
    AI37 (type=AT3)
    AI38 (type=AT3)
    AI43 (type=AT3)
    AI57 (type=AT3)
DS15
    AI0 (type=AT3)
    AI4 (type=AT3)
    AI10 (type=AT5)
```

Figure 4: 'Best' solutions mapped to asset instances

This output provides information for the SSE to consider when making mitigation decisions. To simplify the output, the method does not show the offensive capabilities mitigated by each defensive solution (this information appears elsewhere, such as in Figure 3). Note that the method computes fitness for all defensive solutions mapped to a given threat/asset instance combination, but retains only the highest scoring defensive solution. The collection of highest scoring solutions is then the recommended architecture.

To go along with the output in Figure 4, the simulator performs additional bookkeeping during execution of findBestSolutions to allow it to later provide an 'aggregate' view of each defensive solution, a sampling of which appears in Figure 5.



```
DS18:  383.2 (adj= 358.2) appCnt=46  bestCnt=43  best%= 93.5
DS43:  346.4 (adj=  22.6) appCnt=46  bestCnt=3   best%=  6.5
DS36:  193.4 (adj=   6.0) appCnt=32  bestCnt=1   best%=  3.1
DS48:  222.7 (adj=   0.0) appCnt=32  bestCnt=0   best%=  0.0
DS62:  265.4 (adj= 248.8) appCnt=32  bestCnt=30  best%= 93.8
```

Figure 5: Aggregate Score View

For example, the fifth line of the sample output in Figure 5 shows that defensive solution DS62 has an aggregate score of 265.4, which is the sum of the solution's fitness scores for all the places that it applies in E, which is 32 unique combinations of asset instances and offensive capabilities. The output also shows that DS62 was the 'best' solution in 30 of those cases, giving it an overall 'best' percentage of 93.8%. If we scale the aggregate score of 265.4 by this percentage, the adjusted aggregate score is 248.8.

In combination, figures 4 and 5 provide a local and global view, allowing the SSE to see defensive solutions asset-by-asset, threat-by-threat, but also the overall value of defensive solutions as they pertain to the target environment as a whole.

5. Evaluation and Discussion

This section evaluates the artifacts introduced above.

Model. While the object model is suitable for representing the problem space of interest in this paper, one could enhance the model for broader use, e.g., organizing asset types into a taxonomy to better represent and organize asset type possibilities and expanding the model to include named asset groupings.

Methods. The fitness function, ff, and findBestSolutions method artifacts together select the 'best' solution based on a given set of weighted criteria. The authors chose to relate the criteria in a linear combination instead of arranging criteria into a more general polynomial equation, as a linear combination produced results that we considered to be useful for informing decisions. However, the SSE is free to assign weights along a non-linear scale, if desired.

While the set of criteria chosen in this approach has utility to the authors, we recognize that obtaining values for certain criteria can be a challenge. The use of ordinal scale data that SMEs assign based on their general knowledge partially ameliorates this problem, but ultimately, we would like to introduce, for example, actual cost estimates for the cost attributes associated with defensive capabilities and solutions.

The findBestSolutions method proceeds asset instance by asset instance, considering offensive capabilities that each instance faces. The method includes a more global view as well by computing the applicability criterion value. In addition, and as discussed in the Instantiation section above, the simulator sums up fitness scores for each applicable offensive solution score from the catalog and scales the result based on the percentage of time the solution had the best score.

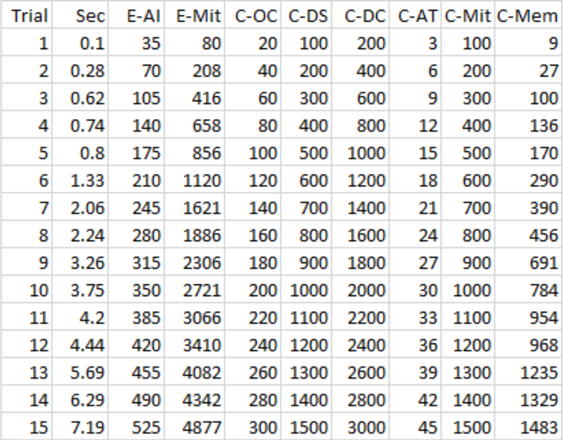
Instantiation. We were interested in performance characteristics of the Java-based instantiation under increasing sizes of catalog and target environment.

Table 4: Variables in Sample Output

Variable	Description
Trial	Trial number (1 to 15)
Sec	Time to generate solutions in seconds
E-AI	Environment: asset instances
E-Mit	Environment: existing mitigations
C-OC	Catalog: offensive capabilities
C-DS	Catalog: defensive solutions
C-DC	Catalog: defensive capabilities
C-AT	Catalog: asset types
C-Mit	Catalog: mapped mitigations
C-Mem	Catalog: solution members (a member is a defensive capability)

With this in mind, we ran 15 trials on a Windows 10 Dell Latitude E5570 laptop with 15 GB of memory and an Intel Core i7-6820 processor. Table 4 describes the variables of interest in a run of the simulator. Table 5 shows the values of each of the variables tracked for each trial execution. Each successive trial used a larger total generated data set for both the Environment (E-prefixed variables) and Catalog (C-prefixed variables). For example, trial 2 had an environment consisting of 70 asset instances and 208 mitigations.

Table 5: Trials and associated data per trial



Trial	Sec	E-AI	E-Mit	C-OC	C-DS	C-DC	C-AT	C-Mit	C-Mem
1	0.1	35	80	20	100	200	3	100	9
2	0.28	70	208	40	200	400	6	200	27
3	0.62	105	416	60	300	600	9	300	100
4	0.74	140	658	80	400	800	12	400	136
5	0.8	175	856	100	500	1000	15	500	170
6	1.33	210	1120	120	600	1200	18	600	290
7	2.06	245	1621	140	700	1400	21	700	390
8	2.24	280	1886	160	800	1600	24	800	456
9	3.26	315	2306	180	900	1800	27	900	691
10	3.75	350	2721	200	1000	2000	30	1000	784
11	4.2	385	3066	220	1100	2200	33	1100	954
12	4.44	420	3410	240	1200	2400	36	1200	968
13	5.69	455	4082	260	1300	2600	39	1300	1235
14	6.29	490	4342	280	1400	2800	42	1400	1329
15	7.19	525	4877	300	1500	3000	45	1500	1483

The second column shows the total time in seconds that the findBestSolutions method took to run, which includes calls to the ff method.

As Figure 6 shows, time increases nearly linearly for the catalog and environment sizes that we sampled. We expect non-linear performance in the long run (for very large catalogs and target environments for analysis) based on the three nested loops of the implementation of equation (2). That said, the catalog and environment sizes in trial #15 are larger than any catalog or target environment we have ever evaluated in our work to date assessing real-world systems.

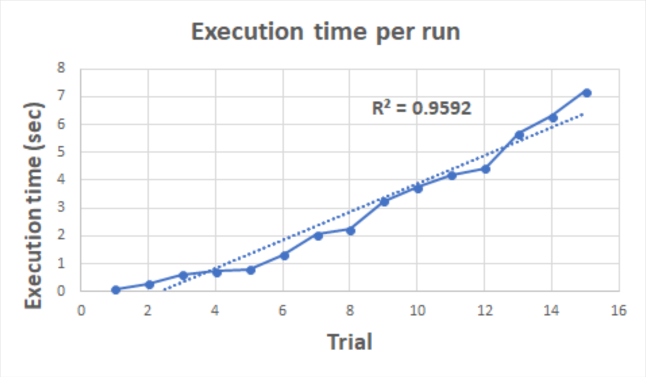


Figure 6: Execution time in seconds per run

The overall approach has certain limitations. For example, one cannot specify “do not exceed” values (e.g., a given budget) or “do not fall below” (e.g., a given level of trustworthiness) values for selected criterion. Other limitations are: the approach does not take into consideration the uncertainty of values for criteria and the approach offers no special assistance for conducting sensitivity analysis beyond manual re-executions that use revised weightings.

6. Conclusions and Future Work

We used Design Science principles in our conception and evaluation of an approach to mitigation selection based on a capability representation for offensive and defensive abilities possessed by attackers and defenders, respectively. The approach uses a set of weighted criteria that are customizable by the SSE based on organizational priorities and constraints. We learned that the approach yielded acceptable performance results for the size of target environments that we commonly see. We also found that we could readily generalize the results to a more global view, specifically a defensive solution’s overall contribution to a target environment.

Future work possibilities include: (1) consider potentially augmenting the approach with more sophisticated methods (e.g., genetic algorithms, linear programming) that can help with the first limitation listed earlier; (2) formally assess the utility of the artifacts to working SSEs, including a survey of SSEs about the solution selection criteria they think are, on average, the most useful and how they would set default weights for those criteria; (3) explore ways to incorporate uncertainty and sensitivity analysis into the approach; (4) apply the artifacts to non-synthetic data sets; (5) compare SSE-selected mitigations to that of the simulator for the same target system and investigate to understand the differences in outcomes; (6) incorporate additional criteria, such as the ability to prefer or avoid certain vendor implementations of given defensive solutions/capabilities and criteria described in the related work section, such as favoring solutions that align to certain standards; and (7) break out the cost criteria into explicit sub-criteria so they can be receive separate weights.

7. Acknowledgements

We wish to thank Michael Smeltzer, Dallas Pearson, and the APL review team for their feedback on early drafts of this paper.

8. References

[1] Equifax, “Equifax Releases Details on Cybersecurity Incident, Announces Personnel Changes,” 2017. .

[2] Verizon, “2017 Data Breach Investigations Report,” *Verizon Bus. J.*, no. 1, pp. 1–48, 2017.

[3] Microsoft, “Microsoft Security Intelligence Report,” 2016.

[4] J. Gosler and L. Von Thaer, “Resilient Military Systems and the Advanced Cyber Threat,” 2013.

[5] T. Llanso, M. McNeil, D. Pearson, and G. Moore, “BluGen: An Analytic Framework for Mission-Cyber Risk Assessment and Mitigation Recommendation,” in *Hawaii International Conference on System Sciences*, 2017, p. 10.

[6] MITRE, “An Overview of MITRE Cyber Situational Awareness Solutions,” 2015.

[7] T. Llanso, P. Hamilton, and M. Silberglitt, “MAAP : Mission Assurance Analytics Platform,” in *IEEE Homeland Security Technologies Conference*, 2012, pp. 549–555.

[8] T. Llanso, G. Tally, M. Silberglitt, and T. Anderson, “Applicability Of Mission-Based Analysis For Assessing Cyber Risk In Critical Infrastructure Systems,” in *International Federation for Information Processing (IFIP) - Critical Infrastructure Protection VII*, 2013th ed., vol. VII, Springer Berlin Heidelberg New York, 2013, pp. 135–148.

[9] R. Caralli, J. Stevens, L. Young, and W. Wilson, “Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process,” 2007. [Online]. Available: http://www.cert.org/octave.

[10] ISACA, “The Risk IT Framework,” 2009.

[11] R. A. M. Schmittling, “Performing a Security Risk Assessment,” *ISACA J.*, vol. 1, 2010.

[12] S. Fenz, A. Ekelhart, and T. Neubauer, “Information security risk management: In which security solutions is it worth investing?,” *Commun. Assoc. Inf. Syst.*, vol. 28, no. 1, pp. 329–356, 2011.

[13] E. Kiesling, A. Ekelhart, B. Grill, C. Strauss, and C. Stummer, “Selecting security control portfolios: a multi-objective simulation-optimization approach,” *EURO J. Decis. Process.*, vol. 4, no. 1–2, pp. 85–117, 2016.

[14] Wikipedia, “Pareto efficiency,” *Wikipedia, the free encyclopedia*, 2017. .

[15] J. Figueira, S. Greco, and M. Ehrogott, *Multiple criteria decision analysis: state of the art surveys*, 78th ed., vol. 78, no. 78. Springer, 2005.

[16] D. Dor and Y. Elovici, “A model of the information security investment decision-making process,” *Comput. Secur.*, vol. 63, pp. 1–13, 2016.

[17] T. Llanso, “CIAM: A Data-driven Approach for Selecting and Prioritizing Security Controls,” in *2012 IEEE International Systems Conference SysCon 2012*, 2012, pp. 1–8.

[18] M. Shapasand, M. Shajari, S. A. H. Golpaygani, and H. Ghavamipoor, “A comprehensive security control selection model for inter-dependent organizational assets structure,” *Inf. Comput. Secur.*, vol. 23, no. 3, pp. 302–316, 2015.

[19] T. Sawik, “Selection of optimal countermeasure portfolio in IT security planning,” *Decis. Support Syst.*, vol. 55, no. 1, pp. 156–164, 2013.

[20] R. Sarala, G. Zayaraz, and V. Vijayalakshmi, “Optimal Selection of Security Countermeasures for Effective Information Security,” *Proc. Int. Conf. Soft Comput. Syst.*, vol. 398, 2016.

[21] Q. Wang and J. Zhu, “Optimal information security investment analyses with the consideration of the benefits of investment and using evolutionary game theory,” *Proc. 2016 Int. Conf. Inf. Manag. ICIM 2016*, pp. 105–109, 2016.

[22] Y. J. Lee, R. J. Kauffman, and R. Sougstad, “Profit-maximizing firm investments in customer information security,” *Decis. Support Syst.*, vol. 51, no. 4, pp. 904–920, 2011.

[23] L. P. Rees, J. K. Deane, T. R. Rakes, and W. H. Baker, “Decision support for Cybersecurity risk planning,” *Decis. Support Syst.*, vol. 51, no. 3, pp. 493–505, 2011.

[24] E. Weishäupl, “Towards a Multi-objective Optimization Model to Support Information Security Investment Decision-making,” *Proc. 4th Work. Secur. Highly Connect. IT Syst. - SHCIS ’17*, pp. 37–42, 2017.

[25] J. Breier and L. Hudec, “On selecting critical security controls,” *Proc. - 2013 Int. Conf. Availability, Reliab. Secur. ARES 2013*, vol. 7799, pp. 582–588, 2013.

[26] E. Kiesling, C. Strauß, and C. Stummer, “A multi-objective decision support framework for simulation-based security control selection,” *Proc. - 2012 7th Int. Conf. Availability, Reliab. Secur. ARES 2012*, pp. 454–462, 2012.

[27] E. Panaousis, A. Fielder, P. Malacaria, C. Hankin, and F. Smeraldi, “Cybersecurity Games and Investments: A Decision Support Approach,” *Lect. Notes Comput. Sci.*, pp. 266–286, 2014.

[28] I. Patterson, J. J. Nutaro, G. Allgood, P. T. Kuruganti, and D. Fugate, “Optimizing investments in cyber-security for critical infrastructure.,” in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, 2013, p. 20.

[29] I. Yevseyeva, V. Basto-Fernandes, M. Emmerich, and A. van Moorsel, “Selecting Optimal Subset of Security Controls,” *Procedia Comput. Sci.*, vol. 64, pp. 1035–1042, 2015.

[30] A. Fielder, E. Panaousis, P. Malacaria, C. Hankin,

and F. Smeraldi, “Decision support approaches for cyber security investment,” *Decis. Support Syst.*, vol. 86, pp. 13–23, 2016.

[31] O. F. El-Gayar and B. D. Fritz, “A web-based multi-perspective decision support system for information security planning,” *Decis. Support Syst.*, vol. 50, no. 1, pp. 43–54, 2010.

[32] M. Gupta, J. Rees, A. Chaturvedi, and J. Chi, “Matching information security vulnerabilities to organizational security profiles: A genetic algorithm approach,” *Decis. Support Syst.*, vol. 41, no. 3, pp. 592–603, 2006.

[33] J.-J. Lv, Y.-S. Zhou, and Y.-Z. Wang, “A Multi-criteria Evaluation Method of Information Security Controls,” *2011 Fourth Int. Jt. Conf. Comput. Sci. Optim.*, pp. 190–194, 2011.

[34] A. R. Otero, “An Information Security Control Assessment Methodology for Organizations,” *Nov. Southeast. Univ. Retrieved from NSUWorks*, no. 266, 2014.

[35] V. Viduto, C. Maple, W. Huang, and D. López-Peréz, “A novel Risk Assessment and Optimisation Model for a multi-objective network security countermeasure selection problem,” *Decis. Support Syst.*, vol. 53, no. 3, pp. 599–610, 2012.

[36] A. Schilling and B. Werners, “Optimal selection of IT security safeguards from an existing knowledge base,” *Eur. J. Oper. Res.*, vol. 248, no. 1, pp. 318–327, 2016.

[37] F. Smeraldi and P. Malacaria, “How to spend it : optimal investment for cyber security Position paper,” *Proc. 1st Int. Work. Agents CyberSecurity*, pp. 1–4, 2014.

[38] D. Buckshaw, G. Parnell, W. Unkenholz, D. Parks, J. Wallner, and S. Saydjari, “Mission Oriented Risk and Design Analysis of Critical Information Systems,” *Mil. Oper. Res.*, vol. 10, no. 2, pp. 19–38, 2005.

[39] P. Dinsmore, “NIPRNet/SIPRNet Cyber Security Architecture Review,” 2016. [Online]. Available: http://www.disa.mil/~/media/Files/DISA/News/Conference/2016/AFCEA-Symposium/3-Dinsmore_NSCSAR.pdf.

[40] NSA, “NSA/CSS Technical Cyber Threat Framework v1,” 2018. [Online]. Available: https://www.iad.gov/iad/library/reports/nsa-css-technical-cyber-threat-framework-v1.cfm.

[41] Object Management Group, “Unified Modeling Language (UML),” 1999. [Online]. Available: http://www.uml.org.

[42] T. Llanso and M. McNeil, “Estimating Software Vulnerability Counts in the Context of Cyber Risk Assessments,” in *Hawaii International Conference on System Sciences*, 2018, p. 7.

[43] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research,” *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, Dec. 2007.

[44] NVD, “National Vulnerability Database.” [Online]. Available: http://nvd.nist.gov.

[45] D. Mellado, E. Fernandez-Medina, and M. Piattini, “A common criteria based security requirements engineering process for the development of secure information systems,” *Comput. Stand. Interfaces*, vol. 29, no. 2, pp. 244–253, 2007.