Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

CYBER RISK OF COORDINATED ATTACKS IN CRITICAL INFRASTRUCTURES

David M. Nicol

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
1308 W Main St, Urbana IL-61820, USA

ABSTRACT

Critical infrastructures such as the electric grid, oil refineries, telecommunications, transportation, water, emergency services, etc., all depend on some cyber infrastructure. Risk analysis of a critical infrastructure to compromised cyber infrastructure is therefore very important, but has certain challenges. This paper examines the problem of integrating analysis of a cyber network with the impacts possible on the critical infrastructures that are possible through cyber access. We find that adding the infrastructure impact cost considerably increases the complexity of the assessment problem.

1 INTRODUCTION

Most of the critical infrastructures for which the US Departments of Energy and Department of Homeland Security are responsible depend on networked computers for their operation. A comprehensive assessment of the risk to a critical infrastructure must include consideration of the possibility that it is attacked through the underlying cyber infrastructure, and consider the damage that such an attack could inflict on or through the physical system. Examples of such infrastructures include the power grid, dams, water and waste water systems, emergency services, transportation, and manufacturing, to name only a few of the 16 sectors named by Presidential Policy Directive 21 (USGovernment).

Such a risk assessment needs to

- identify the physical system actuators accessible through the cyber infrastructure,
- identify how malicious manipulation or interference with these actuators or data from sensors can harm equipment, the environment, or people and assess the cost of such attacks,
- analyze the possibility of intruders or disgruntled insiders gaining access to those actuators, through the cyber infrastructure.

By way of example, consider the transfer of bulk liquids (like liquified natural gas, or oil) between storage tanks and vessels in a port. The actuators of interest include navigation controls on the ship, controls for connecting tanks on the ship to tanks on shore, and controls for the pumps. Attacks might include moving the ship while it is coupled to the on-shore tanks, inhibiting proper connections between the ship tanks and the shore tanks, inhibiting pumping of the liquid between tanks, altering pressure measurements to induce misinformed control messages. Such attacks could harm the vessel, the pumping infrastructure, the tanks, the personnel, and the environment. For each of these the assessment needs to capture how an outside attacker can gain access to the cyber components implementing these controls, and whether (and how) an insider could likewise circumvent safety controls to induce these same consequences.

While all of these points are important, in this paper we focus on the problem of quantifying the access an attacker or insider potentially has to *multiple* actuators, enabling *coordinated attacks* on the critical infrastructure. In the port example a coordinated attack could involve jamming of a wireless network used to send control messages, following an attack that impacted coupling controls. Jamming of the network

could inhibit situational awareness, and inhibit the ability for operators to remotely respond even if they saw or understood what is happening. The analyses of interest consider how the adversary circumvents security controls by compromising devices and uses them as stepping stones, an activity known as "lateral movement". The capabilities obtained are used to build up access to a set of actuators, from which attacks can be launched. Part of the analysis should include information about known vulnerabilities in those devices, and the difficulty of exploiting them. Another part of the analysis should include information about the privileges the adversary acquires by exploiting those vulnerabilities, and the capabilities that result from those exploits. Furthermore the analysis should include how the adversary builds up a portfolio of actuator accesses, and the attacks and subsequent damage that can result.

For the remainder we assume the presence of an adversary, and consider various quantifications that attempt to describe the inherent potential of that adversary to succeed in getting access to multiple actuators. We identify some existing methodologies for quantifying access to individual actuators, and then consider how these can be extended to quantify access to multiple ones. Throughout we comment on the suitability of existing approaches for expressing risk in terms of impact on the critical infrastructure. One model computes a difficulty metric for every attack path and scores the overall network in terms of the easiest set of attack paths that lead to control of actuators. Another model counts the number of different ways an adversary can reach a given set of actuators. Still another model scores the network in terms of the minimum number of vulnerabilities that if patched would completely inhibit the adversary's ability to reach actuators through known vulnerabilities.

We find that path-based cyber assessments have significantly increased computational complexity when extended from attack paths with single endpoints to attacks that include multiple endpoints. Furthermore, we find that path-based cyber assessments do not obviously mesh well with impact assessment of infrastructures. Turning towards protection based assessments (i.e., how much protection is needed to inhibit an attack on the infrastructure or reduce the costs of potential attacks) we find a better fit for integration with the assessment of attack costs on the infrastructure, but also find combinatorial explosion of means of attacks. Based on past experience with cyber risk assessment we believe the problem needs heuristic solution based on a technique such as Monte Carlo sampling.

The remainder of the paper is organized as follows. Section §2 introduces the central data structure we use in the cyber analysis, the network multi-graph. Section §3 describes our model of attacks on the infrastructure and discusses potential cost functions for those attacks. Section §4 visits prior work on using paths through a graph (in our case the network multi-graph, in related work attack-graphs) and extends the problem to consideration of minimal cost ways of an adversary reaching all of a set of actuators from which attacks are launched. Section §5 views the problem from the point of view of how much it costs to protect the infrastructure from attacks made possible by cyber breaches, shows how computational complexity rises by inclusion of the spectrum of attacks possible on the infrastructure, and points out where Monte Carlo simulation will play a role in estimating curves that describe how much the risk to the infrastructure is reduced as a function of the number of actuators protected from compromise, and the cost of that protection. Section §6 gives our conclusions.

2 NETWORK MULTI-GRAPH

All of the analyses we describe are centered on a *network multi-graph*, defined as follows. A graph node represents a device that has one or more IP interfaces, that potentially is a *stepping-stone*. This means that if an adversary were able to gain execution privileges on the device, it could use that device to find and exploit one or more vulnerabilities in another and gain execution privileges on it. A directed edge is defined from node a to node b if, based on the state of our knowledge of the network configuration, hosts, services running on the hosts, and vulnerabilities of services running on the hosts, there *might* be a vulnerability on b that an adversary on a could use to gain a foothold on b without having to pass first through some other device c. If we have cause to believe b has more than one vulnerability that a might exploit, we introduce an edge for each, making this a multi-graph. This definition allows a multi-graph to have an $a \rightarrow b$ edge

in presence of uncertainty. The more information we have about the network/host/services/vulnerabilities state, the better able we are to determine how many $a \rightarrow b$ edges to place (including, possibly, none.)

We augment a network multi-graph with a node representing an adversary. The placement and connections of this node depend on the particular analysis. If we are considering the threat of an adversary external to the network, the adversary node directs edges to network multi-graph nodes that face the Internet—the first step in the intrusion begins outside. We believe that attacks which begin with a successful phishing campaign can be modeled this way, with an edge from the adversary to the mail server to the host on which malicious email might be opened. An attack that begins with a worker bringing a corrupted USB stick and plugging it into a network host would connect the adversary node to network multi-graph nodes whose role is consistent with a human inserting a USB stick. Likewise an attack launch by a malicious insider is modeled by the introduction of edges from the adversary node to each node consistent with a human presence able to log in.

Some of a network multi-graph's nodes are special in that they represent cyber connection points for one or more critical infrastructure actuators, we call these *actuator nodes*. An *attack path* is a path through the network multi-graph whose origin is the adversary node and which reaches an actuator node. An attack path captures a sequence of exploits the adversary executes in order to reach an actuator.

Edges in the network multi-graph may have labels. The nature of the label depends on the quantification model being employed. For example, in Nicol and Mallapura (2014) we define a network multi-graph assuming that (i) all direct host-to-host connections permitted by firewalled access control are known, (ii) all open ports and services available on hosts are known, and (iii) Common Vulnerability Scoring System (Schiffman et al. 2004) information about the vulnerabilities is acquired from the National Vulnerability Database. Edge labels encode what is defined as the *exploit complexity score*, a number between 0 and 10 with lower scores implying easier exploits to execute. The score of an attack path is the sum of the labels of the edges on the path; the lower the attack path score, the easier this model deems the attack it reflects to be.

Other labels are certainly possible. One might label an edge from $a \rightarrow b$ with a score that captures the damage an adversary might cause at b once compromised. A model like this could weigh the edge into an actuator node to reflect estimates of the damage possible to the physical system by compromising the actuator; potential damage (like wiping file systems) at interior nodes could be scored, to reflect that the damage an adversary could cause extends beyond the access to the actuators. In a model where the larger the edge weights the greater the damage, one might look for paths with heavy sums of edge weights, or possibly paths with largest edge weights.

We'll say more about using individual path costs to score a network, the point to be made right now is that a network multi-graph represents key information needed to quantify the risk contribution of cyber.

3 ATTACK COST MODEL

In this paper we focus on attacks which necessitate the adversary gaining execution privileges on multiple actuators. Different attacks may require the adversary to have different levels of privilege on an actuator. Here we follow the CVSS categorization (NIST) for reported vulnerabilities, an attribute of which is whether the vulnerability once exploited gives the adversary no execution privileges, "low" privileges, or "high" privileges. The CVSS does include an attribute indicating whether one needs physical access or network access to exploit the vulnerability; for actuators we're interested only in those that do not need physical access. The important point is that the CVSS model we adopt does not, for remotely exploitable vulnerabilities, specify the execution privilege level of the adversary on the host from which the exploit is remotely launched. The end state of the adversary's intrusion is a "compromise vector" which describes the state of each actuator. The state of an actuator in this vector is one of four integer codes; we define 0:"uncompromised", 1:"compromised with no execution privileges", 2:"compromised with low execution privileges", or 3:"compromised with high execution privileges". We use integer codes for actuator states to employ vector-based comparisons. Vector $V = (v_1, v_2, \ldots, v_n)$ is said to dominate

vector $U = (u_1, u_2, \ldots, u_n)$, written $U \leq V$, if $u_i \leq v_i$ for all component positions i. We assume that any attack on the critical infrastructure which is possible given compromise vector C_1 is also possible given any compromise vector C_2 with $C_1 \leq C_2$.

A number of different known attacks on the infrastructure may be possible from a compromise vector C. We need to ascribe some potential cost to these attacks. One way is to enumerate the attacks $a_1, a_2, \ldots a_k$ and for each ascribe some (random) attack cost $A(C, a_i)$. We could score the cost of C (say, $S(C)$) by the sum of these random costs, by some function of these random costs, or treat a subsequent attack selection itself as a random action, with attack a_i being selected with probability p_i. For the purposes of this paper any method which makes sense to a practitioner will do. One can include unknown attacks also if one is comfortable with estimating the cost of an unknown attack. Then the attack becomes just one of potentially many and is treated in the same way as the known attacks.

There are potentially a combinatorially large number of unique compromise vectors, and unless scoring costs are automated, it simply isn't feasible to completely define S for all vectors C. We can use ordering relationships among the potential compromise vectors to simplify the task. First, a random variable Y is said to be *stochastically larger* than random variable X, denoted $X \prec Y$, if $\Pr\{Y \leq t\} \leq \Pr\{X \leq t\}$ for all t. It means that the cumulative distribution function of X uniformly dominates that of Y, implying that Y "tends" to be larger than X in a strong sense. Applying this notion to the random attack costs, we assume that if $C_1 \leq C_2$, then $S(C_1) \prec S(C_2)$. Informally this means that if a compromise vector dominates another, then its random cost dominates the random cost of the other, strongly. When S is simply the sum of the costs of attacks made possible by a compromise vector we can *prove* this assumption, because the attacks possible under the dominant compromise vector subsume those possible under the weaker one. We could no doubt with some assumptions on the mixing probabilities prove the assumption in the case when S represents a mixture random variable, but the notational effort really isn't worth the benefit. Intuitively, what we are assuming is that if the attacker has greater execution privileges on the actuators, then the attacker can inflict greater damage on the infrastructure.

4 PATH-BASED QUANTIFICATION

A natural calculation to apply to a weighted graph is to find the path or paths with the most extreme cost (i.e., least cost or greatest cost, depending on the weights.) In Nicol and Mallapura (2014) we consider the k paths with least cost, interpreted as the k easiest sequences leading to compromise of a critical asset. While seemingly intuitive, this quantification has problems. As we point out in Nicol and Mallapura (2014), from the point of view of an adversary, after a vulnerability v has been exploited once, the difficulty of exploiting it again later in the same sequence may not be the same—having learned how to perform the exploit or found the tool to perform the exploit, subsequent appearances of v are likely to be easier. We show that if the cost of an edge e in a path can depend on the identity of edges earlier in the path, then the problem of finding the shortest path is intractable (formally, it is **NP-hard**.) Thus models where edge weights may vary, depending on what the adversary has already seen or done will need to consider approximations. The approach we took in Nicol and Mallapura (2014) used randomly generated paths where choices out of a node a are chosen randomly, with probabilities heuristically skewed towards (hopefully) shorter remaining paths to critical resources.

With modification to the network multi-graph we can adapt the path cost measure to quantify the difficulty of coordinated attacks that require the adversary to have a foothold simultaneously on multiple actuators. Suppose we select a priori the actuators of interest, e.g., those that must be compromised to accomplish a particular attack of interest on the infrastructure. We can ask which subset of edges connect the adversary node to each actuator in this set, such that the sum of the edge weights is minimal. This is illustrated in Figure 1. Here we illustrate a very simple network multi-graph with annotations on the adversary node, and two actuators which need to be compromised for a given attack on the infrastructure.

We illustrate sets of edges that connect the adversary to all the actuators. One of these sets involves five edges, the other involves four. If the edges have unit weight the one with four edges is minimal.

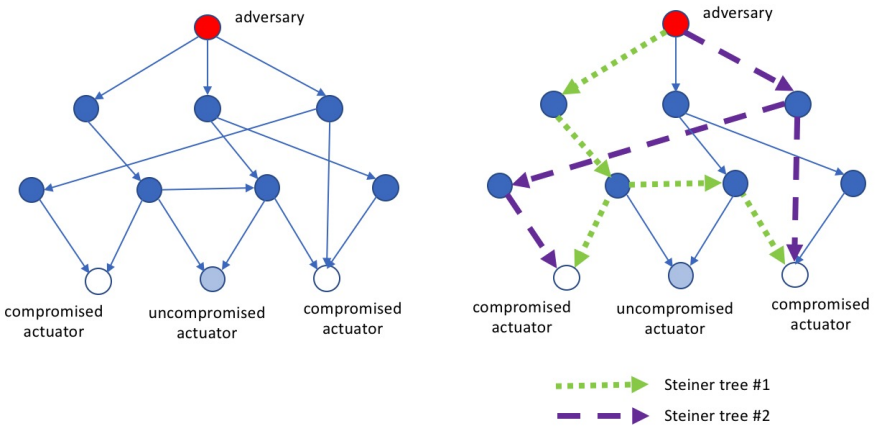

Figure 1: Illustration of Steiner trees in network graph.

This is a generalization of the minimum cost path problem. Correspondingly, when we allow edge costs to be dependent on the identity of other edges, our earlier result shows that finding this subset will be NP-hard. As it happens, even if edge costs are fixed the problem of finding a set with minimum cost is NP-hard, as the problem formulation is that of finding a minimum cost *Steiner Tree* in a directed graph (Du and Hu 2008). Thus heuristics are needed. To follow the direction we took with the shortest-path problem we would want to modify existing heuristics for minimum Steiner tree with some guided Monte Carlo exploration of the potentially huge solution space.

Depending on the vulnerabilities, for a given set A of actuators there may be a large combination of assignments of privilege to the compromised actuators in A. The ability to launch a certain attack may depend on the adversary having high privilege on a particular subset of actuators; if the adversary does not have high privilege on any one of that subset, that attack isn't possible. For a given set of privilege requirements we can modify the multi-graph prior to the Steiner tree optimization by the simple expedient of visiting each actuator and removing from it all inbound edges corresponding to vulnerabilities whose exploit does not grant the adversary sufficient privilege. Following this modification we ought to verify that there is *some* path remaining from the adversary node to the actuator, otherwise we can conclude that no Steiner tree exists which includes it.

An aspect of the Steiner tree approach is that one needs to identify a priori which actuators are to be included in the tree. With n actuators there are $2^n - 1$ possible non-empty subsets, and for each given subset of size k, 3^k combinations of execution privilege. The effort of finding a minimum cost tree for a given set of actuators with given acquired privileges alone is intractable, we see now that there are a huge number of combinations of sets and privileges. In the face of this we need to focus on specific cases motivated by known attacks. For a given attack one would determine which actuators and their privilege levels are needed, create a modified network multi-graph and determine the (approximate) minimal cost to the adversary of acquiring the resources to launch that attack. Note that if we have a compromise vector V with minimal Steiner tree score x, then for any compromise vector U with $U \leq V$ we know that (i) the score of a minimal Steiner tree associated with U is no greater x, and that $S(U) \prec S(V)$.

We could also consider bounding cases, such as

- For each individual actuator and each individual execution privilege, the effort of reaching that actuator and the cost to the infrastructure of the attacks enabled by that access.
- Given the set of all actuators, each with "low" privilege and separately each with "high" privilege, the minimum Steiner tree cost, and the cost to the infrastructure of attacks enabled by that access.

Computational complexity issues aside, it isn't clear how to intuitively connect the costs of Steiner trees to the costs of the resulting potential attacks on the infrastructure. The units aren't the same. Furthermore, identification of the most efficient way an adversary has of compromising actuators is not the same as the effort the adversary will need to expend to compromise those actuators, because the adversary does not have perfect knowledge of the system. These are not the only issues with path-based quantification. While a minimal path or minimal Steiner tree does depend on all graph information in the sense that the cost is least over all possibilities, these measures do not seem to capture diversity of possible ways the adversary has to compromise actuators. Consider for example Figure 2. The two graphs shown have identical min-cost paths, with value 3. However, one graph has only two paths, while the other has thirteen. The minimal path measure has nothing to say about the richness of opportunity an attacker has at achieving an exploit.

One can tractably compute the number of unique paths in a directed acyclic graph. If we use $n(a,b)$ to denote the number of unique paths originating in a and reaching b, we can use a breadth-first traversal to compute

$$n(a,b) = \sum_{p \text{ with } p \to b} n(a,p)$$

with $n(a,a) = 1$ by definition. If $p_1 \neq p_2$, then a path from a to p_1 is distinct from a path from a to p_2, so that the total number of unique paths to b simply adds the number of unique paths from a to each of its predecessors in the graph.

While the number of unique paths to an actuator node can be efficiently computed, it has its own limitations as a sole quantifier of cyber risk. Consider: suppose by patching one vulnerability we can reduce the number of paths from the adversary node to actuator nodes from one million to one hundred thousand. Clearly there is improvement, but are we comfortable inferring that the network is ten times more secure? Even if we were to score by computing the logarithm (base 10) of the number of paths, is that the right measure? Patching multiple vulnerabilities to reduce the path count to one hundred...is that network now (only) three times more secure? Another problem with simply counting paths is that two paths may share sub-sequences. While there are thirteen unique paths in the denser graph of Figure 2, the maximum number of paths between source and sink that do not share any edges is five. The maximum number of edge-independent paths is perhaps a better metric for isolating attack potentials, but isn't easily combined with the costs to the critical infrastructure that result from a successful attack.

We are left with the observation that path based enumerations don't correlate well with an intuitive sense of the difficulty an adversary has in gaining access to actuators. Nor is it clear how to connect the cost model of Section §3 with path counts, path lengths, or Steiner tree costs. Part of the problem is that the minimum costs can give the adversary too much credit for intuiting precisely the easiest traversal, while the number of paths imagines the adversary trying every possible path. Neither of these fits adversary's behavior. We will return to this point later.

5 PROTECTION BASED QUANTIFICATION

Path-based quantification tries to capture how easy or how hard it is for an adversary to reach a set of actuator nodes, and achieve a particular set of execution capabilities on them. A different way to quantify the cyber risk is to base it on the work needed to protect devices, e.g., by patching vulnerabilities. In our network multi-graph every individual instance of a vulnerability is expressed as an edge. To eliminate that vulnerability instance somehow is to remove an edge. There may be a financial cost associated with that removal. So, unlike the path based quantification, we have hope here of being able to align the analysis of the cyber network with the costs to the infrastructure of a cyber-enabled attack—both analyses can have units of monetary value. This kind of analysis offers the hope of asking questions such as "If I have N euros to spend on defense, how do I spend them to as to maximize the reduction in potential costs (in euros) to the infrastructure?"

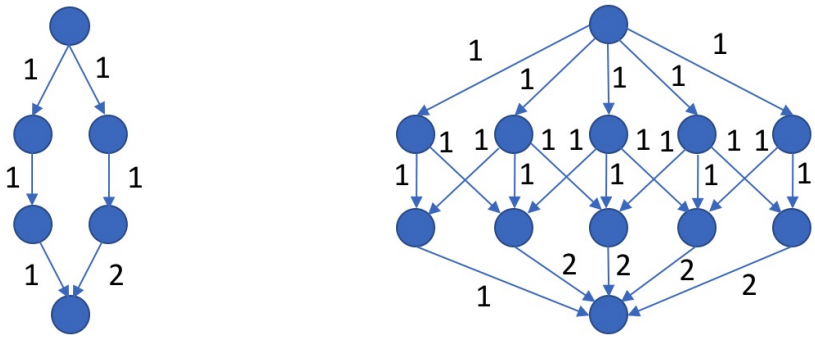

Figure 2: Two network graphs with equal least-cost paths.

There is a significant literature on related problems. The specific problem of optimally choosing instances of vulnerabilities to remove so as to protect all of a set of critical resources (or in a more formal setting, prohibit entry into a particular subset of system states) is sometimes known as the minimum critical attack set (Ammann et al. 2002; Jha et al. 2002; Sheyner et al. 2002; Wang et al. 2006). As we will see, the additional consideration of attacks on the infrastructure that might be launched once a compromise vector is achieved adds another dimension to the problem.

The famous $s-t$ min-cut algorithm (Hao and Orlin 1994) has application here. Given nodes s and t in a directed graph with weighted edges, it asks to find the set of edges with minimum sum of edge weights which when removed eliminate every path from s to t. Efficient solutions are known. This metric can be used as a measure of a given actuator a_i node's exposure to vulnerability exploits that achieve a particular execution privilege p on that actuator. If the execution privilege is higher than "none", we modify the actuator's representation in the network multi-graph by removing all inbound edges associated with vulnerabilities whose exploits lead to privilege strictly less than p. If every edge is scored by value 1, a min-cut computation with that edge weight assignment will compute and identify the minimum number of vulnerability instances which if patched will keep the adversary from achieving execution privilege p on a_i. Notice that if we score edges by the financial cost of applying the patch (a cost which may include estimated costs of dealing with adverse reactions to applying the patch if they should occur) then the min-cut solution identifies the cheapest way of providing protection to that actuator.

The ability to compute the cost of isolating just one actuator can be translated into the ability to efficiently determine the minimum cost of keeping the adversary of achieving a given compromise vector V. While the minimum critical attack set problem formulation would look for a minimal set of vulnerability instances whose removal would deny the adversary *every* actuator in V, to thwart the vector in aggregate all we need do is deny the adversary one actuator at the privilege level specified in V. For every such actuator in V we modify the network multi-graph as described above, and find a cut-set that denies the adversary access to that actuator at the privilege level required. If we do this for every compromised actuator represented in V we can find the actuator-privilege pair whose separation denies V from the adversary at minimum cost.

Now a given attack A on the infrastructure may be enabled from a number of compromise vectors $V_1, V_2, \ldots, V_m$. To completely deny the adversary the ability to mount A we need to deny him every compromise vector V_j. This is the added dimension beyond the minimal critical attack set problem—the attack on the infrastructure can be initiated from multiple sets of cyber targets. However, the machinery we've identified helps us find a feasible—if not minimal—solution. As described above we can find for each V_i a cut-set with minimum cost that denies the adversary V_i. We can then take the union of all

minimum cut sets so identified. The sum of the costs of patching the vulnerability instances in those sets is no greater than the sum of the costs of the individual cut-sets.

This construction makes no attempt to take advantage of using the denial of one actuator at a given privilege level to deny the adversary multiple possible compromise vectors. We describe now an approach that does. We construct a bipartite graph where one set of nodes represent each V_i, one node for each. The second set of nodes are comprised of actuator-privilege pairs extracted from all the V_i. We connect the V_i node to actuator-privilege pair (a_j, p) if that pair appears in V_i. Note that a given pair may, if denied the adversary, deny him more than one compromise vector. Note that the V_i's connected to a given pair (a_j, p) are a subset of the universe of all the V_i, and that the union of all such subsets is the universe of all the V_i. The set-cover problem (Cormen et al. 2009) asks for a selection of these subsets with minimal cardinality whose union is the universe. This is equivalent to looking for the smallest number of pairs of actuator-privilege level nodes such that the union of the compromise vector nodes they touch covers all the compromise vector nodes. The minimum cover set problem is NP-complete, but good heuristics are known for it. Thus we can go from identifying all the minimal set of compromise vectors that enable attack A to identifying a subset of actuator-privilege priority pairs such that if all pairs in the subset are denied the adversary, then A is not possible. With a final application of the min-cut algorithm we can find a minimum cost set of vulnerability instances whose patching denies the adversary from using attack A. Given the actuator-privilege pairs from above, we modify the network multi-graph in two ways. First we visit every actuator node identified in the set of pairs and delete from it every inbound edge corresponding to a vulnerability whose resulting privilege level is less than that specified for the actuator in that set. Second, we introduce a dummy terminal node, direct an edge to it from each of the identified actuator nodes, and weight each edge with a value L equal to the sum of the weights on all other edges in the entire multi-graph. This weighting ensures that none of these edges will appear in the min-cut we next apply to disconnect the adversary node from the dummy terminal node. Whatever set of edges are identified by the min-cut will lay on paths between the adversary and the actuators.

Patching is only one way of increasing a network's protection. For example, one might introduce a firewall with a configuration that eliminates access to multiple vulnerabilities (hence removes multiple numbers of edges) with a single placement, but can introduce its own vulnerabilities (as it will have a management interface for administrative access.) We can approach the problem of efficiently protecting from a given attack A, at least starting the process as before. Starting with A and finding the compromise vectors from which A can be launched we find as before a minimum set of actuator-privilege pairs to be denied. Instead of a final min-cut algorithm though we need to consider how best to place firewalls. Given a budget B is it possible to place firewalls whose total installation cost is less than B which isolates that set of actuator nodes? One can take this a step further and combine the possibility of patching and other protections, using some of the budget B to install firewalls, and some to apply to patching. There, given a placement of firewalls whose total cost is c, does the min-cut solution have cost $B - c$ or less?

However, all of this machinery is seemingly needed to determine how to protect from a single attack. The space of possible attacks is large. Needed is a mechanism that gives one a sense of the cost and associated risk reduction over a spectrum of possibilities. One "knob" in this spectrum could be the number of actuators with exploits yielding a given privilege level. For example, we could find the single actuator whose compromise at privilege level "low" permits the largest attack cost on the infrastructure, simply by going through all the possibilities, and determine the cost of protecting against it. We could do the same at privilege level "high". We could next consider all pairs of actuators at a given privilege level (if the number of actuators is not too large.) As we increase the number of compromised actuators in the compromise vector we run into combinatorial challenges and must resort to Monte Carlo sampling to estimate the largest impact possible on the infrastructure from attacks launched using k compromised actuators at a given privilege level, and the cost of protecting against those attacks. An alternative formulation might use Monte Carlo to estimate, for a given budget B, a given number of compromised actuators and a given privilege level the maximum reduction in attack cost possible by investing B.

6 CONCLUSION

This paper examines the challenge of integrating assessment of how an adversary can attack a cyber network with the cost of the attacks on critical infrastructures that are possible as a result. We find that path-based cyber assessments are problematic in this regard, but that protection based mechanisms have a more natural fit. In both cases we see that the solution complexity rises significantly by including the extra complications of different attacks on the infrastructure. These complexities necessitate the use of Monte Carlo based heuristics to base assessments on simulations rather than complex algorithmic analysis. Future work will address these issues.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number, 2015-ST-061-CIRC01. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

REFERENCES

Ammann, P., D. Wijesekera, and S. Kaushik. 2002. "Scalable, Graph-based Network Vulnerability Analysis". In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, 217–224. New York, NY, USA: ACM.

Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein. 2009. *Introduction to Algorithms, Third Edition*. 3rd ed. The MIT Press.

Du, D., and X. Hu. 2008. *Steiner Tree Problems In Computer Communication Networks*. River Edge, NJ, USA: World Scientific Publishing Co., Inc.

Hao, J., and J. Orlin. 1994. "A Faster Algorithm for Finding the Minimum Cut in a Directed Graph". *Journal of Algorithms* 17(3):424 – 446.

Jha, S., O. Sheyner, and J. Wing. 2002. "Two Formal Analys s of Attack Graphs". In *Proceedings of the 15th IEEE Workshop on Computer Security Foundations*, CSFW '02, 49–. Washington, DC, USA: IEEE Computer Society.

Nicol, D. M., and V. Mallapura. 2014. "Modeling and Analysis of Stepping Stone Attacks". In *Proceedings of the 2014 Winter Simulation Conference*, 3036–3047. edited by A.Tolk et al., Piscataway, New Jersey:IEEE.

NIST. "Common Vulnerability Scoring Calculator, v3". https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator. Accessed May 5, 2018.

Schiffman, M., G. Eschelbeck, D. Ahmad, A. Wright, and S. Romanosky. 2004. "CVSS: A common vulnerability scoring system". *National Infrastructure Advisory Council (NIAC)*.

Sheyner, O., J. Haines, S. Jha, R. Lippmann, and J. M. Wing. 2002. "Automated Generation and Analysis of Attack Graphs". In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, SP '02, 273–. Washington, DC, USA: IEEE Computer Society.

USGovernment. "Presidential Policy Directive 21". https://obamawhitehouse.archives.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil. Accessed July 20, 2018.

Wang, L., S. Noel, and S. Jajodia. 2006, November. "Minimum-cost Network Hardening Using Attack Graphs". *Comput. Commun.* 29(18):3812–3824.

AUTHOR BIOGRAPHIES

DAVID M. NICOL is the Franklin W. Woeltge Professor of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign, and Director of the Information Trust Institute. He is the PI for

two national centers for infrastructure resilience: the DHS-funded Critical Infrastructure Reliance Institute, and the DoE funded Cyber Resilient Energy Delivery Consortium, and is the Director of the Advanced Digital Sciences Center in Singapore, and PI of its Trustworthy and Secure Cyber Plexus program. His research interests include trust analysis of networks and software, analytic modeling, and parallelized discrete-event simulation, research which has lead to the founding of startup company Network Perception, and election as Fellow of the IEEE and Fellow of the ACM. He is the inaugural recipient of the ACM SIGSIM Outstanding Contributions award. He received the M.S. (1983) and Ph.D. (1985) degrees in computer science from the University of Virginia, and the B.A. degree in mathematics (1979) from Carleton College. His email address is dmnicol@illinois.edu.