Attacker-Parametrised Attack Graphs

Alastair Janse van Rensburg*, Jason R.C. Nurse and Michael Goldsmith

Department of Computer Science, University of Oxford, Oxford, UK
Email: alastair.jansevanrensburg@cs.ox.ac.uk*

***Abstract*—Computer network attackers chain system exploits together to achieve their goals, which range from stealing data to corrupting systems. Attack graphs represent these paths through the network, and provide the basis for calculating many security metrics. In this paper, we seek to extend graph-based analysis from the consideration of single graphs to the consideration of multiple. By performing analysis on many graphs at once, we consider the range of threats faced and avoid the downsides of several current techniques, which focus purely on known and expected attackers. In particular, we propose a novel method of generating a set of attack graphs, parametrised by attacker profiles. Our technique would enable security analysts to consider the security of their network from the perspective of many attackers simultaneously. This contrasts with existing techniques, which typically analyse attacker-independent graphs or graphs constructed around predefined attacker profiles. We analyse the resulting set of graphs first through deterministic methods and then using a probability measure.**

Keywords–Attack Graphs; Attacker Profiling; Intrusion Detection.

I. Introduction

Attack graphs are a useful tool for network security analysts, facilitating quantitative study of computer network security. The graph acts as a map of the vulnerabilities in the network, revealing how attackers can combine exploits to achieve their goals. Attack graphs are the basis for calculating many security metrics which provide the analyst with practical information [5].

There are many approaches to attack graphs, and one of the most relevant to our work is presented by Dantu *et al.*, who examine attack paths using three attacker profiles [1]. Their approach examines the risk posed to assets from attackers matching each profile, which represent the network's most significant adversaries. Zhang *et al.* use modelling artefacts to capture relationships between exploits, in a manner similar to our method's *capabilities* [9].

Another area of relevance to our work is metrics on graphs. Wang *et al.* emphasise the importance of careful composition of individual metrics, and demonstrate that poor interpretation of metrics can decrease security [8]. Homer *et al.* present a probabilistic method to quantify risk on attack graphs [3]. They observe that assuming attack paths have independent probabilities will most likely not lead to the correct conclusion. They also claim precise that estimates are not required to take action; it is sufficient to be able to class vulnerabilities as "high risk" or "low risk" if, for example, this facilitates patching of "high risk" vulnerabilities first.

The contribution of this paper is towards expanding profiling techniques: generating complete sets of profiles instead of choosing them individually; defining a probability measure to enable a risk-centric analysis; and making these large sets practical by aggregating metrics across them. This paper represents a work-in-progress, and challenges with the current method and potential future work are discussed at the end.

The remainder of this paper is structured as follows: in Section II, we provide context for our approach. Section III introduces our technique. In Section IV, we discuss applications, in particular looking at ways our method can be applied deterministically and probabilistically to provide actionable information to analysts. Finally, in Section V, we reflect on our method, providing a summary of the benefits and challenges.

II. Context

The precise definition of attack graph varies between authors. With this in mind, our technique aims to avoid being prescriptive – it can be adapted to any typical definition (e.g. [4][7]). For clarity, we choose a definition of attack graph that will be used throughout the paper. Specifically, an *attack graph*, $G = (V, E)$, is a graph consisting of a vertex set V and a (directed) edge multiset E. It may be a multigraph (with multiple edges between the same vertex pair). A simple example graph is presented in Figure 1.

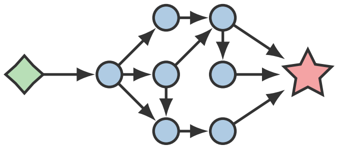

Figure 1. An example attack graph; the initial state is a diamond, the attacker's goal is a red star.

Vertices correspond to states of the network and attacker, and edges correspond to actions that the attacker can take. The attacker seeks to move the network between states, starting from the *initial state* and potentially reaching the *goal*. The presence of an edge indicates a possible action that an attacker could take, but not necessarily an action that they will take. We do not have a strict definition of state – but these typically reflect both the condition of the network and the attacker's access to it. A state might be, for example, that the attacker has root access to a given server, or it might be that a piece of software has been crashed by the attacker.

A. Attacker Profiling

Attacker profiling brings knowledge of the attackers into the modelling process. Network intruders have varying abilities and resources – nation states are likely to have vastly different capabilities to disgruntled ex-employees, who might have subtly different capabilities to those who deface websites. As a result, each attacker has a different set of exploits available to them. Essentially, for each attacker we can derive an individual attack graph, containing only the actions they can perform.

ISBN: 978-1-61208-493-0

It is impractical, however, to predict precisely the attributes of attackers who will attack the network. Instead, attackers are usually categorised into more general profiles. (This is not necessarily a loss of accuracy; if an exploit requires £500 to perform it is not important whether an attacker has £1,000 available to them or £1,000,000. It is sufficient to know that they have *more than £500*, and so our attacker profile need only contain this fact.)

Existing work uses a variety of methods to decide how these attack profiles are created – typically focused around contributions from experts. The aim is to focus analysis: creating profiles to disregard unlikely exploits and give more attention to exploits that attackers are capable of. To facilitate this, a small number of profiles are designed: Grunske and Joyce illustrate their method with two pre-determined profiles [2]; Dantu *et al.* create three profiles based on a survey [1]. These techniques capture the risk posed by these profiles, but with a loss of generality – it may be true that attackers primarily fall into expected categories, but it is not certain that every attacker will. Additionally, if the wrong profiles are chosen then the analysis will be inaccurate as a result.

Our technique seeks to benefit from attacker profiling without neglecting unexpected attackers. To achieve this, we generate a profile for each possible combination of attributes. We restrict ourselves (for the sake of practicality) to binary facts about the attackers. This, we feel, is justified because the impact on the resulting attack graphs is also binary – either an exploit is possible or it is not; either an edge is present or it is not. Using this large and complete family of profiles we aim to have considered the network's security from the perspective of any possible attacker.

III. Attacker Parametrisation

To perform our technique, we require a well-structured set of attacker profiles, and a map from these attacker profiles to attack graphs. With these, we will be able to move from conclusions about individual graphs (corresponding to individual attackers) to conclusions about the set of graphs (corresponding to all attackers). A well-structured set of attackers allows us to translate observations about the resulting graphs to observations about the underlying structure.

To achieve our method, we first define an **attacker profile** as a set of *capabilities*. Each capability is a property which the profile either has or does not have. Such a capability might be "physical access", representing attackers who can gain physical access to the relevant hardware, or "access to X hacking toolkit", representing attackers who can use a specific piece of software. From a set of n capabilities, we generate a total of 2^n attacker profiles, corresponding to each different combination of capabilities. By defining the profiles in this way, we have a set of profiles which is as complete as possible – every combination of capabilities is represented, regardless of how likely or expected they are. Some of these profiles will be those we expect, possessing some (but not all) the capabilities. Others will represent much less plausible attackers; one profile has every capability, one has no capabilities at all.

To utilise the profiles, we use a **base attack graph**. This is the attack graph generated through standard methods, without of consideration of the attackers. We then augment this graph by assigning each edge a condition on profiles. These conditions represent the ability to perform the corresponding attack. A condition could be "does the profile have capability X?", but they do not need to be simple – a complex condition could require multiple capabilities: "does the profile have X, or Y and Z?". For example, pressing a hardware reset button on a piece of equipment would require physical access, so the corresponding edge might have the condition "does the profile have the *physical access* capability?".

We then prune the base attack graph with respect to a profile in order to create an **attacker-profile graph**. This graph corresponds to only the attacks of which the profile is capable. To perform the pruning, we simply remove all edges whose conditions are not satisfied by the profile. We can create such a graph for each possible profile (i.e., each possible combination of capabilities). Consequently, we have converted a base attack graph and a collection of capabilities into a complete set of attacker-profile graphs. Each graph in this set represents how our network appears from the perspective of an attacker with a particular set of capabilities. This is illustrated in Figure 2.

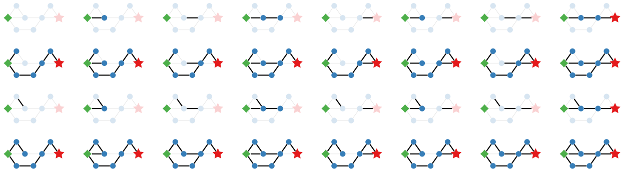

Figure 2. The attacker-profile graphs drawn out for each profile, corresponding to the example in Figure 3.

Formally, we will let C denote the set of all capabilities. An attacker profile is then a subset of C, containing the capabilities of the profile. An attacker-profile graph for the profile $A \subseteq C$ will be denoted G_A. This is the graph that contains only edges whose condition is satisfied by the profile A. We will also use μ to denote a metric on attack graphs, so that $\mu(G) \in \mathbb{R}$, for an attack graph G. For example, μ could be the number of paths from the initial state to a given goal vertex.

IV. Application of the Approach

Using every possible attacker-profile graph ensures the analysis is general. However, considering a large number of profiles requires special techniques, as it is impractical to individually examine each graph. Analysis of a single attacker-profile graph may be beneficial, and standard techniques can be applied to these. However, the key benefits of our method will be gained when analysts are able to consider every graph simultaneously, minimising the additional work while still treating every attack profile separately for as much of the process as possible. To this end, we first explore key properties of the set that may be useful to analysts. We then investigate the effect of defining a probability measure on the set of capabilities, which will enable a risk-centric examination.

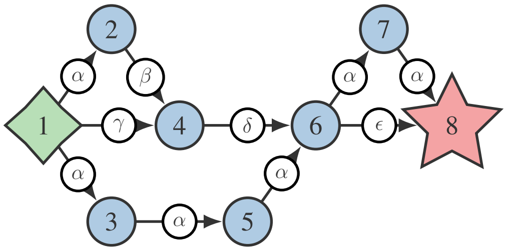

Figure 3. An attack graph, each edge labelled with a single capability that is sufficient and necessary to perform that edge. Vertex v_i is labelled i.

To illustrate the following techniques and to demonstrate them, we will refer to the example attack graph in Figure 3, in which each edge has been labelled with the capability from the set $\{\alpha, \beta, \gamma, \delta, \epsilon\}$ that (alone) enables it (i.e., each edge has the condition "does the profile have capability X?", where X is the label in the corresponding white circle on the graph).

A. Deterministic Analysis

Security analysts may be interested in how well their network is defended against different combinations of capabilities. Which sets of capabilities (there may be multiple) are sufficient to reach the goal? Which capabilities are necessary? By determining these, analysts can examine which capabilities they need to concern themselves with the most. If it is discovered that physical access is necessary to reach the goal, then they may be well-advised to increase their protection of physical systems. Conversely, if physical access is not necessary nor in any of the sufficient sets then it may be that their efforts are better focused elsewhere.

In the example, $\{\alpha\}$ is a sufficient set: an attacker with this capability would be able to reach the goal. From a network-defence standpoint this may be a concern is a sufficient set – attackers need only one capability to reach the goal. This would be of particular concern if α is a simple capability (e.g., access to a common piece of hacking software), but may be of less concern if α is an unusual capability or one that can perhaps be controlled (e.g., physical access to a server room).

This can be generalised to looking at vertices other than the goal node. For a vertex $v \in V(G)$ we can define $n(v)$ to be the set of necessary capabilities that an attacker must have in order to reach the vertex v. If an attacker had been detected (through, for example, an Intrusion Detection System) as having reached a set of states $\{v_1, v_2, \cdots v_k\}$ then it can be inferred that they have, at least, the capability set $\bigcup_{i \in [k]} n(v_i)$. Subsequent decisions on how to react to the attack, or where else the attacker might have reached, can be informed by this. Any state for which this capability set is sufficient to access is a target which the attacker has shown they have the ability to reach.

Relating this again to our example above, we can see that $n(v_2) = \{\alpha\}$ (v_2 is labelled simply as "2" in Figure 3). This, together with the fact that $\{\alpha\}$ is a sufficient set for the goal, implies that an attacker who is detected at v_2 has the capability to reach the goal. Even if the system is only capable of detecting attackers reaching v_2 and not the other states, then a detection at v_2 is a suggestion that the attacker may have also reached the goal, as they have demonstrated the capability to do so. This could act as an early warning when reacting to live events – especially if an attacker demonstrates unusual capabilities early in their intrusion.

B. Probabilistic Analysis

Many existing attack graph techniques use probability to capture the uncertainty in predicting the attributes of attackers. We define a probability measure, $\mathbb{P}$, on the set of attack profiles, weighting the profiles according to the likelihood that an attacker matches them. For a profile $A \subseteq C$, $\mathbb{P}[\{A\}]$ is the probability that an attacker has the capabilities in profile A (and no other capabilities). $\mathbb{P}[\{B \cup \{\alpha\} : B \in \mathcal{P}(C)\}]$ is the probability that an attacker has, at minimum, the capability α.

For the example, we will assume that each capability has a probability of $\frac{1}{2}$, and that each capability is independent of the others. From this, we can derive the probability of each profile – in this example, we see that each profile has the same probability, $2^{-5} = \frac{1}{32}$.

By assigning probabilities to profiles we capture some dependencies between exploits. Compare the two paths to the goal $v_1 \to v_4 \to v_6 \to v_8$ and $v_1 \to v_3 \to v_5 \to v_6 \to v_7 \to v_8$. The first path consists of 3 actions (or edges) and the second path consists of 5. If independent probabilities were assigned directly to edges, this would imply that the probability that an attacker can successfully perform the first path is $\frac{1}{8}$ and the second path is $\frac{1}{32}$ – leading to the conclusion that the first path is the more likely. However, by assigning probabilities to profiles we see that in fact all the actions in the second path are similar; an attacker who can perform one of the attacks can perform the rest. As a result, they all require the same capability and so the probability that an attacker can perform the entire path is the probability that an attacker has that capability: $\frac{1}{2}$. By factoring in this additional information we reach a contradictory (but more accurate) conclusion: the second path is, in fact, more likely despite being longer.

Using the probability measure we are able to generalise any metric on a single attack graph to being a metric on the whole set of attack graphs. Let μ be a metric that maps a graph to a number, representing some property of the graph. Then we can extend μ to the set of graphs by taking its expectation. If the probabilities and capabilities are accurate, this gives a much more representative value than applying the metric to the base attack graph directly. Indeed, the base attack graph assumes the worst-case scenario: it assumes every attacker has every capability. By splitting the attackers into profiles and applying the metric to each profile, we capture interactions between different profiles and exploits. By weighting them with the probability measure and aggregating them, we create a risk-centric summary of the metric.

One metric we can apply to our example is the *number of paths*, that is, we let μ_1 be the number of paths that exist from the initial state v_1 to the goal state v_8. For the base graph, $\mu_1(G) = 6$. In contrast, $\mathbb{E}[\mu_1(G.)] \approx 1.19$, a considerably lower value. This is because the vast majority of attackers do not have every capability. By treating each profile separately and then combining the results of the metric, we get a much more reasonable answer – the expected number of paths an attacker has to the goal is about 1.19. This is a much more meaningful (and potentially reassuring) statement than the conclusion that any attacker can choose from 6 paths to the goal, which the base attack graph might suggest.

We can also define metrics that are not particularly useful on individual graphs but aggregate to give useful results. For instance, we define μ_2 to be 1 if there is any path to the goal and 0 otherwise. Unsurprisingly $\mu_2(G) = 1$, telling us that, for the profile with every capability, there is at least one path to the goal. On the other hand, $\mathbb{E}[\mu_2(G.)] \approx 0.56$, which indicates that about 44% of attackers have no path to the goal. Such a value might be useful when evaluating the security of two possible network configurations.

Modelling each profile with a separate graph enables us to look at properties which depend on the whole graph. It may be the case that the system administrators are able to fix some small set of exploits, but not all of them (it could be too cost prohibitive, or impact usability too much). Attack graphs provide an excellent basis for making this decision; the graph shows the consequences of each exploit and demonstrates the impact of removing it. However, using a base attack graph for this decision is equivalent to making your decision solely around attackers with every capability. Through our method, we can make this decision to target as many attackers (weighted by probability) as possible. As a result, the exploits removed will lead to the greatest reduction in risk.

In the example, removing the edge from $v_4 \rightarrow v_6$ appears to have the most benefit to the base attack graph. It removes the shortest path, and can only be circumvented by the comparatively long route $v_1 \rightarrow v_3 \rightarrow v_5 \rightarrow v_6$. The *number of paths* metric supports this: removing $v_4 \rightarrow v_6$ has the greatest effect, lowering the metric from 6 to 2 – the next-best removal only lowers the metric to 3. If we use the *expected* number of paths instead, we see that removing edge $v_4 \rightarrow v_6$ does lower the metric to 0.75, an improvement on the original 1.19. But this is not the optimal edge removal: removing $v_5 \rightarrow v_6$ lowers the expectation even further to 0.44. This is because attackers only use edge $v_4 \rightarrow v_6$ when they also have other capabilities (a successful attack via this edge requires γ or both α and β to reach the edge, and then α or ϵ to reach the goal). So its removal has significant impact for the few attackers with many capabilities, but little or no impact for the majority. Conversely, removing $v_5 \rightarrow v_6$ prevents the most likely attack path, and has the greatest impact for typical attackers.

V. Conclusion and Future Work

In this paper, we proposed an extension of existing attacker-profiling techniques on attack graphs. Our method improves on existing methods by constructing profiles from a set of capabilities, resulting in a complete collection of profiles for analysis, providing a more rounded consideration of attackers.

A. Contribution and Summary of Benefits

Our method has two main improvements over existing techniques. Firstly, we model each profile with its own attack graph, giving us a conclusion based only on attacks from that profile. These conclusions are then aggregated to give an overall picture of the network. In this way, we avoid conflating attacker profiles and over- or underestimating attackers.

Secondly, we define profiles as collections of capabilities and generate a complete set of profile. This ensures analysis is not only about expected attackers. We use flexible definitions of *capabilities* to allow analysts to model attackers in as much (or as little) detail as required. Considering attackers in terms of their capabilities leads to useful statements about these capabilities. Necessary or sufficient capability sets provide information to analysts which can aid them both when designing and evaluating networks, and when reacting to live incidents.

We avoid complex probability assignments by shifting probabilities from edges to capabilities. In particular, we do not require many large conditional probability tables, as in Bayesian Attack Graphs [6]. The number of capabilities is also much smaller than the number of edges, meaning fewer probability assignments must be made. A typical attack graph could have hundreds or thousands of edges [4], but could be modelled with far fewer capabilities. By assigning probabilities to a smaller set we make it feasible for analysts to spend more time on each value and consider each dependency. Capability probabilities may also be reusable between different networks and potentially even between similar organisations. This would result in significant reductions to the work required.

As demonstrated above, applying metrics to the base attack graph assumes the worst-case scenario. While this evaluation is useful, an expectation-based method gives more realistic results, factoring in knowledge about the attacker to provide a risk-based summary of the network. Our method allows the extension of any metric on individual graphs, enabling use of the substantial existing work on attack graph metrics.

B. Challenges for Future Work

Future work in this area will seek to address several key challenges: Firstly, it is not yet clear how best to build capability sets. These are crucial for successful application, so a framework for finding them would be greatly beneficial. It is expected that this process could be supported by automated techniques, or by shareable and reusable templates.

Secondly, it is not straightforward to assign probabilities. In any probability-based attack graph, considerable effort must be applied to assign probabilities. This is particularly true when dependencies between values are considered. We believe our method alleviates this difficulty by reducing the number of required assignments (other methods require thousands of edges to have assigned probabilities, while our method only requires them for capabilities). However, this is still a challenge which could be improved with further work.

Thirdly, as our method is still at an early point, we have not undertaken testing to validate results. We do not believe our method will have significant computational overhead, but it may require additional effort from experts compared to some existing methods.

References

[1] R. Dantu, P. Kolan, and J. Cangussu, “Network risk management using attacker profiling,” *Security and Communication Networks*, vol. 2, no. 1, pp. 83–96, Jan. 2009. [Online]. Available: http://doi.wiley.com/10.1002/sec.58

[2] L. Grunske and D. Joyce, “Quantitative risk-based security prediction for component-based systems with explicitly modeled attack profiles,” *Journal of Systems and Software*, vol. 81, no. 8, pp. 1327–1345, Aug. 2008. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0164121207003056

[3] J. Homer, X. Ou, and D. Schmidt, “A sound and practical approach to quantifying security risk in enterprise networks,” *Kansas State University Technical Report*, pp. 1–15, 2009.

[4] S. Noel and S. Jajodia, “Managing attack graph complexity through visual hierarchical aggregation,” in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. ACM, 2004, pp. 109–118. [Online]. Available: http://dl.acm.org/citation.cfm?id=1029225

[5] ——, “Metrics suite for network attack graph analytics,” in *Proceedings of the 9th Annual Cyber and Information Security Research Conference*. ACM, 2014, pp. 5–8. [Online]. Available: http://dl.acm.org/citation.cfm?id=2602117

[6] N. Poolsappasit, R. Dewri, and I. Ray, “Dynamic Security Risk Management Using Bayesian Attack Graphs,” *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 61–74, Jan. 2012.

[7] L. Wang, A. Liu, and S. Jajodia, “Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts,” *Computer Communications*, vol. 29, no. 15, pp. 2917–2933, Sep. 2006. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S014036640600137X

[8] L. Wang, A. Singhal, and S. Jajodia, “Toward measuring network security using attack graphs,” in *Proceedings of the 2007 ACM workshop on Quality of protection*. ACM, 2007, pp. 49–54. [Online]. Available: http://dl.acm.org/citation.cfm?id=1314273

[9] S. Zhang, X. Ou, A. Singhal, and J. Homer, “An empirical study of a vulnerability metric aggregation method,” DTIC Document, Tech. Rep., 2011.