An empirical test of the accuracy of an attack graph analysis tool

Teodor Sommestad, *Department of Information Security and IT Architecture, Swedish Defence Research Agency (FOI), Linköping, Sweden*

Fredrik Sandström, *Department of Computer Science, Umeå University, Umeå, Sweden*

Purpose: The purpose of this paper is to test the practical utility of attack graph analysis. Attack graphs have been proposed as a viable solution to many problems in computer network security management. After individual vulnerabilities are identified with a vulnerability scanner, an attack graph can relate the individual vulnerabilities to the possibility of an attack and subsequently analyze and predict which privileges attackers could obtain through multi-step attacks (in which multiple vulnerabilities are exploited in sequence).

Design/methodology/approach: The attack graph tool, MulVAL, was fed information from the vulnerability scanner Nexpose and network topology information from 8 fictitious organizations containing 199 machines. Two teams of attackers attempted to infiltrate these networks over the course of two days and reported which machines they compromised and which attack paths they attempted to use. Their reports are compared to the predictions of the attack graph analysis.

Findings: The prediction accuracy of the attack graph analysis was poor. Attackers were more than three times likely to compromise a host predicted as impossible to compromise compared to a host that was predicted as possible to compromise. Furthermore, 29 per cent of the hosts predicted as impossible to compromise were compromised during the two days. The inaccuracy of the vulnerability scanner and MulVAL's interpretation of vulnerability information are primary reasons for the poor prediction accuracy.

Originality/value: Although considerable research contributions have been made to the development of attack graphs, and several analysis methods have been proposed using attack graphs, the extant literature does not describe any tests of their accuracy under realistic conditions.

Keywords: Assessments, Security, Computer security, Computer networks, Attack graphs

1 Introduction

Securing computer networks is a complicated and difficult task. Computer networks in today's enterprises often consist of many hosts. These hosts run many different operating systems and software applications of different versions and configurations. Experience suggests that, in a typical enterprise, a considerable portion of these hosts will contain publicly known vulnerabilities that may be exploited to obtain user and host privileges in the organization. However, not all vulnerabilities are equally important to remediate. Some can be used to provide the attacker with all of the privileges of a host (e.g., root access on a Linux machine), and some can only be used to provide or impact a subset of them (e.g., reading certain parts of the memory in a machine). Furthermore, vulnerabilities may require certain preconditions to be met to be exploitable. For example, it might be required that the attacker be able to interact with the machine physically or that the attacker already possess the credentials of some user in the network. As a result of factors such as these, some vulnerabilities may be difficult for an outsider to exploit on public networks, and some may be easily exploited. Additionally, the successful exploitation of some vulnerabilities may provide access that makes it possible to exploit other vulnerabilities that are not directly exploitable from public networks. Because of these contingencies, analyzing how to prioritize remediation options or determine the present risks can become overwhelmingly complex.

Attack graphs have been designed to assist decision makers in this analysis. According to Heberlein et al. (2004), "*one of the primary focuses of the attack graph efforts is to identify how an adversary can chain together vulnerability exploitation to increase his capability*." An attack graph aims to answer questions such as the following: To which hosts can an attacker on the Internet gain access? In which ways can attackers gain root access to host X, Y, or Z? Which attacks will become impossible if the firewall is set to block port 80? Which attacks will become impossible if all instances of CVE-2014-0497 are removed?

Clearly, a tool that is able to answer questions of this sort will support a decision maker who is considering different remediation options or performing a risk analysis. In addition, this type of tool can also be used in conjunction with other techniques, such as intrusion detection systems (Roschke et al.,

2010) or forensics tools (Liu et al., 2012), to improve their analysis capabilities. Many research papers have been produced on attack graphs, and several software tools have been developed. However, the practical utility and validity of these tools in a realistic setting is unclear, and there are reasons to doubt that attack graphs yield accurate results. For example, the utility of the attack graph approach is dependent on the availability of information about the analyzed computer network and the vulnerabilities associated with it. The designers of tools aiming to support decision makers propose that such information should be gathered with the help of vulnerability and network scanners (see (Ou et al., 2005) (Jajodia, 2007) (Artz, 2002)). However, it is known that vulnerability scanners are limited in their accuracy and only detect approximately half of the vulnerabilities in a network (Holm et al., 2011). Because of such potential issues, this paper attempts to answer the research question: *How well do attack graphs predict the success or failure of attacks under realistic conditions?*

This paper presents a test of MulVAL (Ou et al., 2005), one of the more commonly cited and more mature attack graph tools. In this test, MulVAL is fed network configuration data and vulnerability data collected using the Nexpose vulnerability scanner (from Rapid7) and is used to analyze eight computer networks of different size and complexity. The output of this analysis is compared to observations of successful attacks performed during an offensive cyber security exercise. The accuracy of MulVAL's predictions is reported in terms of how well the predictions correspond to the observations made by attackers on successful and unsuccessful attacks. To our knowledge, this is the first empirical test of an attack graph tool against observed attack attempts. Based on these empirical results, suggestions are made as to how predictions can be improved.

The outline of this paper is as follows. Section 2 gives an overview of attack graphs, with a particular emphasis on the variant used in MulVAL. Section 3 describes the methods used in the test. Section 4 presents the results, which are subsequently discussed in section 5.

2 Attack Graphs

Multiple articles provide reviews and overviews of different attack graph approaches. Lippmann and Ingols (2005) reviewed existing approaches in 2005 and classified them based on how they viewed the goals of the attackers, how they were generated, and how well they scaled. Problems were identified with respect to scalability, obtaining attack (or vulnerability) details, and identifying what attackers could connect to from different locations in the network. Multiple methods of representing attack graphs have evolved (Alhomidi and Reed, 2012). Heberlein et al. (2012) offers a canonical representation of attack graphs, illustrated in Figure 1.

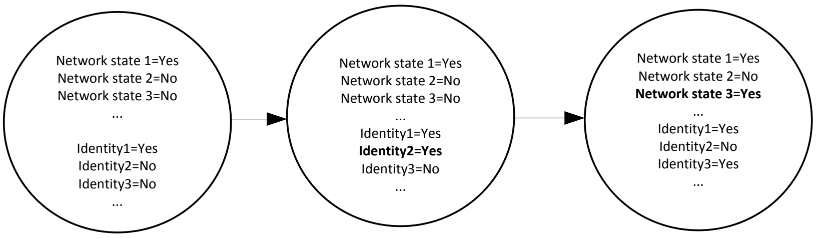

Figure 1. A canonical representation of attack graphs, with transitions between states in the network and identities controlled by the attacker. Adapted from Heberlein et al. (2012).

In this representation, the nodes of the graph represent the state of the entire computer network at a given time. Here, a state characterizes both the state of the computer network itself (including hosts, software, configurations, authorized users and vulnerabilities) and the identities that the attacker owns (i.e., user accounts under the attacker's control). With this state space as a starting point, an attack graph shows how the attacker can move from one state to another using the vulnerabilities and controlled identities present in the network. An attacker can thus either change the state of the network itself (e.g., by reconfiguring a firewall) or obtain a new identify (e.g., by stealing a user's password or session).

Heberlein et al. (2004) also describe some features of the analyses used in attack graph approaches. These features can be summarized as follows:

- **Monotonic or non-monotonic exploits.** To simplify the analysis and offer scalability, it is commonly assumed that exploits are monotonic, i.e., that once the attacker has obtained some identity, that identity is never lost. MulVAL assumes that exploits are monotonic.
- **Single path or all paths**. For simplicity, the analysis is sometimes limited to generating a single attack path, i.e.,

the analysis stops as soon as one possible path to the specified state is found. MulVAL finds all paths.

- **Forward or backward chaining.** In a forward chaining technique, the analysis starts with an assumption of the current state and assesses which states are reachable. In a backward chaining technique, an end-state of interest is defined, and the analysis assesses which states may lead to that end-state. MulVAL uses backward chaining.

The mature attack graph tools from academia, namely MulVAL (Ou et al., 2005) (Homer and Ou, 2009), the TVA tool (Jajodia, 2007) (Noel et al., 2009) (Jajodia and Noel, 2010), and NetSPA (Artz, 2002) (Lippmann, 2002) (Chu et al., 2010) (Ingols et al., 2009), share the same features, except that the TVA tool uses forward chaining. There are, however, other differences between these tools. For example, MulVAL uses Datalog rules to specify its input to Prolog, whereas the TVA tool uses more loosely defined input formats and operates on matrices representing the attack steps. On a conceptual level, it can be argued that the tools share the same problems and weaknesses, and the accuracy of one of these tools ought to reflect the accuracy of the other tools. Thus, while MulVAL is used in this test, the results ought to be generalizable to similar attack graph approaches.

Theoretically, there is little reason to question the validity of the inferences produced by any of the attack graph tools. If the tools' algorithms are provided correct input data, they will most likely produce correct output data and yield accurate results. Unfortunately, fully correct input data are rarely available to the decision makers in enterprises, leading to the question of how accurate results will be under realistic conditions. No reports detailing tests of the accuracy under the conditions proposed in research papers, in which the analyses describe attack paths based on input from vulnerability scanners, can be found for any of these tools. The paper that comes the closest to providing a test of accuracy is the test performed by Zhang et al. (2011). In this test, seven servers were assessed using MulVAL on several occasions. The different results produced by the tool on these different occasions were correlated with, and explained by, changes in the state of the computers (e.g., new vulnerabilities). Consequently, the test assessed the internal validity of the tool, but did not assess its validity in practice or how well it predicted the success of real attacks.

3 Methods and Materials

This section describes the methods and materials used in the test. Section 3.1 describes the criteria used to assess the accuracy and utility of the predictions and the analysis method. Section 3.2 describes the computer networks on which the predictions were made. Section 3.3 describes the attackers in the test, the scenario by which they were guided and the actual attacks they performed. Section 3.4 describes the tool configuration and how the output of the tool was codified.

3.1 Assessment criteria

This test aimed to evaluate the accuracy of the results produced by an attack graph tool. There are different ways to view the output of such tools, with implications for the assessment criteria they ought to be evaluated against. These issues are discussed below.

First, the inclusion or exclusion of the paths in a graph may be performed differently. It might be expected that the attack graph will 1) show all paths that can definitely be used, 2) only exclude paths that definitely cannot be used, or 3) aim to assign all possible attack attempts to the class that fits best. In this test, we assume that the attack graphs adhere to the third option and identify both attacks that are possible and attacks that are impossible. This criterion is well in line with the common view of attack graphs. For example, Alhomidi and Reed (2012) state that attack graphs "show all ways of how an attacker violates a security policy."

Second, the output can be interpreted in a possibilistic or probabilistic fashion. When the output is interpreted possibilistically, all attacks with a likelihood of success above zero are included in the attack graph, even if the likelihood that they can be exploited in practice is infinitely small. In a probabilistic interpretation, it is expected that attempts to perform the attack steps in the graph are likely to succeed, e.g., because they are more likely than some threshold value. In any of these interpretations, it should be expected that the inclusion of an attack path and the possibility of compromising a machine correspond to a higher success rate than the excluded paths and those machines for which there is no possibility of compromise and that all successful attacks are included in the attack graph. Put differently, a possibilistic method should at least be able to predict when an attacker will fail any attempted attack and include all successful attacks. There should therefore be an agreement between the attack graph's predictions and the observed success rates.

Third, one may require correctness at different levels of abstraction. These levels of abstraction may vary from the lowest level, of individual exploits and ports, to higher levels, such as when the output (e.g., the number of paths) is considered to provide only indications or examples of how vulnerable a network is. This assessment focuses on the ability to predict which machines attackers can execute code on and the abstraction level of hosts and their attack paths. This is the attack type and the abstraction level used in the vast majority of all proposals related to attack graphs. The tool is assessed by comparing the attack paths produced by the tool with the attackers' observations and producing a confusion matrix for the predictions (i.e., the possible and impossible attack paths) and the (successful and failed) attacks against designated target machines.

3.2 Attacked computer networks

In this test, over a thousand virtual machines were deployed, together forming computer networks of various sizes and complexity to represent a synthetic threat environment. Of these machines, 199 machines in eight fictitious organizations of different types were monitored and assigned as targets. The 199 machines used 88 different virtual machine templates and were all configured differently in some regard, e.g., with respect to users. Some organizations' networks consisted of only a few computers, representing a small organization or personal network; other organizations' networks consisted of several VLANs with firewalls limiting the access possibilities between them. **Fel! Hittar inte referenskälla.** illustrates one of the monitored computer networks.

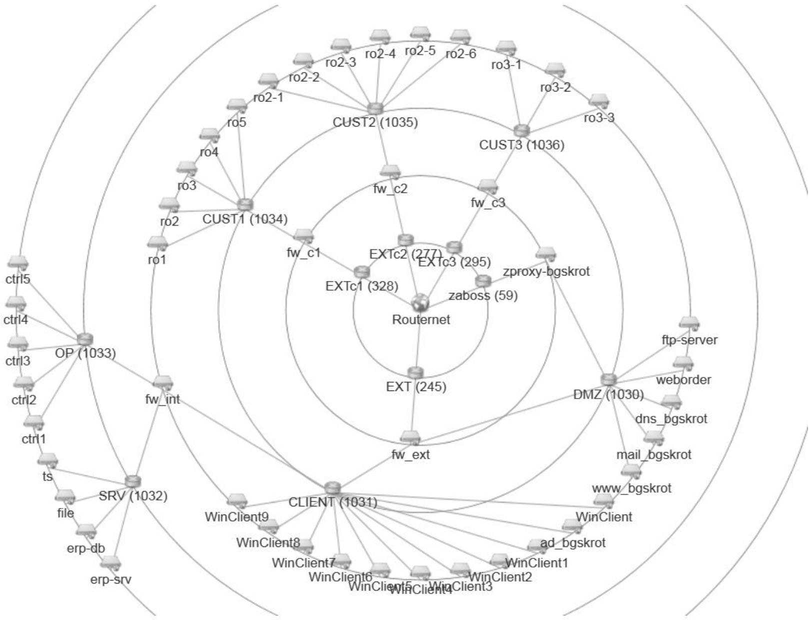

Figure 2. Example of an organization monitored and targeted in the test. Circular nodes are network interfaces, and rectangular nodes are machines. The globe is the link to the "Internet" of the test environment

Multiple operating systems and applications were instantiated in these networks. Different patch levels and versions of Windows (including Windows 2000, XP, and 7 and Windows Server 2003 and 2008) and several versions of Linux-based distributions (including Gentoo, Debian, and Ubuntu) were utilized. These ran multiple desktop and server applications. Among others, the client-side applications included different versions of Adobe Reader, software development tools such as Visual Studio, Internet Explorer, and Firefox, and applications of the Microsoft Office suite or Open Office. Server-side applications included different versions of Wordpress, phpMyAdmin, IIS, domain controllers, network infrastructure services (e.g., DNS and DHCP), and FTP servers. The aim was that the deployed applications should be representative of the standard software found in most enterprises' computer networks. However, to enable a meaningful exercise for the attacking teams and to produce enough data for the test, these applications were more vulnerable than the ones found in the typical enterprise (i.e., they had not been updated and patched recently). Another difference between this environment and the computer networks typical of organizations is the lack of custom-built applications (e.g., interconnected spreadsheet applications) and larger enterprise systems (e.g., ERP systems).

To allow interaction with the users on the machines, scripts implemented in Auto IT (Jonathan Bennett AutoIt Consulting Ltd, 2013) emulated user actions. These scripts used the applications to send emails to each other, surf websites in the environment, open emails and attachments and access files on local machines. To produce a realistic behavioral pattern, the emulated behaviors were performed according to a predefined instruction list created based on the actions of real users in an office environment. The instruction list was produced by collecting the historic usage of web browsers and desktop applications and the outgoing emails of 17 users in three organizations (two research organizations and one game developer). To generate a variety of user behavior, the historic records (covering months to years of computer usage) were split into one-week instruction sets, which allowed thousands of instruction lists (i.e., "users") to be created. The activities performed by the scripted user agents thus followed the same sequence and had the same intensity as real users during a typical workweek. However, the agents were dumb in the sense that they did not engage in dialogs. Their responses to the emails sent to them were to click all links and open all attachments, i.e., they were always cooperative as seen from the attackers' point of view. Furthermore, they only performed standard user behaviors; more sporadic tasks, such as the installation of custom software applications or administrative tasks, were not performed in this test.

3.3 Attackers, scenario and attacks

Two independent teams attacked the computer networks to find secret keys hidden within them. One team consisted of security researchers from the Swedish Defence Research Agency, and the other team consisted of security specialists from the Swedish Armed Forces Network and Telecommunications Unit. Both teams restricted themselves to using only publicly available tools and publicly known exploits, e.g., the tools and exploits packed with the Backtrack 5 operating system.

The attacks started at noon on a Tuesday in November of 2012 and continued until noon on Thursday of the same week, with no scheduled interruptions or pauses. The attackers had no prior knowledge of which machines had secret keys hidden in them but were told that they were in some of the eight computer networks. Their mission was to compromise machines and extracts the keys. As a result, a mix of reconnaissance and penetration activities was conducted. The two teams logged 134 penetration attempts and 31 network scans aimed at the networks monitored in this test. Additionally, over 100 other types of reconnaissance activities were undertaken, e.g., ARP scans. These penetration attempts led to 40 successfully compromised machines. In all of these, privileges were obtained to execute code as root, system administrator, or similar.

The reconnaissance activities and attacks, together with their successes, were recorded by each attacker in a log. The accuracy of these logs was assessed by comparing them to recordings made by screen capture software installed on the machines used by some of the attackers. These comparisons showed very high agreement between what was done and what was logged. All of the deviations identified concerned reconnaissance activities, e.g., network scans.

3.4 Tool configuration and input data

The MulVAL project is open source and available for download (see Ou et al. (2013)). MulVAL requires information on vulnerabilities, subnets, users and host access control lists as input. For each vulnerability, the following fields are specified: the hostname, Common Vulnerabilities Enumeration ID, program (target service or application), range list (if it is remotely or locally exploitable), type of loss (textual), severity (high/medium/low), access control requirement (high/medium/low), service and port. Subnets are described by the hosts they include, and the host access control lists describe by which protocol and on which port entities (i.e., hosts or subnets) they are allowed to communicate. MulVAL also makes it possible to include users and their accounts in the analysis. These users can be labeled as incompetent, meaning they are susceptible to social engineering attacks. In this test, machines with active user agents were labeled as incompetent to reflect that the users opened all email attachments and clicked links indiscriminately.

Descriptions of the systems used in this test were provided to MulVAL by extracting data from a) the system configuration files used to instantiate the computer networks in the cyber range and b) vulnerability scans of the targeted hosts, produced by Nexpose. The system configuration files were validated to provide a reliable source of information and were used for all information except the vulnerabilities and host access control lists. Vulnerability scans were used to collect information about the vulnerabilities in the hosts. These scans were performed repeatedly on the individual vulnerabilities on each of the deployed hosts (88 different variants on 199 machines) in a controlled environment to ensure a reliable output. The host access control lists were created based on the IP table files in the deployed firewalls and the settings in the host firewalls.

To perform these steps and support the analysis, a web-based tool was created that imported the system configuration files and vulnerability scans and performed the analysis of interest. MulVAL was set to analyze all the ways an attacker located on the internet (i.e., outside the computer networks) could execute code on the hosts in the targeted computer networks using different privilege levels. MulVAL operates on the *root*, *user* and less precise *someUser* privileges. Backward chaining was then used to determine all of the ways the machines could be reached. Individual attack paths and attack steps in the resulting attack graph were identified through a breadth-first search on a digraph produced based on the XML file generated by MulVAL. The networks used in this test generated a considerable number of possible paths. With a search depth of 60 steps, 495,360 existing attack paths were found for the eight networks and 199 machines. These steps were compared to the steps attackers used when evaluating the prediction accuracy.

4 Results

The prediction accuracy of MulVAL is presented in section 4.1. As will be shown, the accuracy of MulVAL's prediction poorly fit the attackers' successes and failures. Section 4.2 goes a step further and investigates the reasons for MulVAL's incorrect predictions.

4.1 Prediction accuracy

All attacks performed by the attackers in this test led to the ability to execute code at the highest privilege level (viz. root, system administrator or similar) on the targeted machines. Table 1 summarizes the results and the relationships between attackers' successes in compromising machines and the tool's predictions of the ability to execute code on machines at the highest privilege level (called *root* in MulVAL). In other words, Table 1 describes the relationship between the attackers' ability to obtain root privileges on machines and the tools predictions for obtaining root privileges on the machines.

Table 1. Confusion matrix of MulVAL's predictions for code execution privileges as root and attackers success at obtaining such privileges.

		Code execution by attackers as root	
		Yes	No
Prediction for code execution as root	Possible	6	74
	Impossible	34	85

As Table 1 shows, the attackers successfully compromised 40 hosts in the computer networks as root. Only 6 (15%) of these hosts were compromised using an attack path included in MulVAL's output. The attackers failed to compromise 159 of the hosts they probed, with 85 (53%) of these predicted to be impossible by MulVAL. The attacker could only execute code as root on 6 (8%) of the 80 machines that MulVAL predicted could be compromised to give root-level code execution privileges. Thus, 74 (93%) of the machines that MulVAL predicted were possible to obtain root access on were not compromised. Of the 119 machines MulVAL predicted as impossible to compromise, 34 (29%) were successfully compromised as root. In three of these 34 cases, MulVAL reported the machine as reachable but failed to describe a path that included the exploited vulnerability.

4.2 *Reasons for incorrect predictions*

The successful attacks missed by MulVAL can be explained by a combination of inaccurate vulnerability scans and inaccurate interpretations of vulnerability information. This section provides further details on this.

MulVAL requires information about the vulnerabilities in the computer network. This information is typically, as in this case, collected using a network vulnerability scanner. As could be expected, the imperfect information provided by the vulnerability scanner influences the results negatively. All 34 hosts reported as false negatives in Table 1 were compromised using a vulnerability Nexpose did not report. However, this is only a part of the explanation. If these exploited vulnerabilities that Nexpose missed are manually added to the scan results, only seven of the 34 become true positives. The remaining 27 machines are still assessed as impossible to execute code on with root privileges, but all are possible to execute code on with lower privileges. The reason for this is the way MulVAL interprets and processes vulnerability information.

All the vulnerabilities missed by the scanner (CVE-1999-0504, CVE2003-0352, CVE-2006-3439, CVE-2007-3039, CVE-2007-1748, CVE-2008-4250, and CVE-2010-0478) are known to be able to yield root/administrator/system privileges without subsequent attacks that yield escalate privileges. An indication of this is that all but one (CVE-1999-0504) are marked as having full impact on confidentiality, integrity and availability in the US National Vulnerability Database. However, MulVAL labels these, and all other remote exploits, as yielding the access level of *someUser*. Local exploits are labeled as giving root privileges. In fact, further inspection of the data shows that none of the thirteen true positives obtained with the corrected vulnerability scans were predicted entirely correct. All these attacks involved a privilege escalation attack that increased the *someUser* privileges to *root* privileges or reused *root*-level accounts reached elsewhere in the attack graph.

If vulnerabilities are added manually and MulVAL's interpretation is overridden so that they yield *root*-level privileges, all 40 successful attacks are predicted. All 40 are also predicted as possible to compromise if it is judged as sufficient if MulVAL identifies that code can be executed with some privileges (i.e., *root*-level privileges are not required); however, under such conditions, the vast majority of machines not compromised are also predicted as possible to compromise. Thus, in summary, a combination of missing vulnerability information and poor processing of vulnerability information explains the false negatives.

The 74 false positives are more problematic to find the reason for in this test. When attackers fail to compromise a machine, this provides a clear indication that the machine is more difficult to compromise than the other machines that were compromised. However, their failure does not mean that attacks are impossible to accomplish. Because of this, it is unclear how many of the 74 false positives for root-level access would have been true positives in another test. The 74 false positives include predictions of attacks that were attempted and failed, but it also includes cases where the attackers scanned the host but failed to find an attack worth trying. Furthermore, typical attacks performed against the 74 machines are password-guessing attacks against SSH logins, malware attached to email and exploitation of web application vulnerabilities. Thus, while attackers tried to compromise the machines, they sometimes did not find a path at all, and they rarely tried a stable exploit included in MulVAL's attack paths.

5 Discussion and Future Work

Attack graph tools such as MulVAL are easy to use. Provided that host access control lists can be produced (e.g., from firewall rules) and vulnerability information can be collected (using a vulnerability scanner), the tools can perform their analysis. The output of the analysis contains both a list of the privileges each attack leads to (e.g., on which machines the attack can execute code) and a full graph of the attacks providing those privileges (e.g., the software vulnerabilities exploited). A decision maker may try to work with this information directly or add an analysis technique on top of the attack graph tool's output. For example, critical paths can be identified to produce a prioritized list of mitigation options using ranking algorithms, as described in Sawilla and Ou (2008). Given the sheer number of paths that are produced (almost 500,000 in this test), such post-analysis methods are likely to be needed to make the analysis results comprehensible.

This test did not investigate techniques that could build on attack graphs or how the techniques could influence the analysis results. Instead, the test directly investigated the accuracy of MulVAL when used in combination with a vulnerability scanner. The accuracy of such an analysis would serve as the foundation for analysis methods based on attack graphs. Unfortunately, the results show that MulVAL failed to predict which attacks red teams could accomplish during a two-day exercise:

- The red teams were three times as likely to accomplish what MulVAL predicted as impossible (29% success rate) as to accomplish attacks along paths MulVAL predicted as possible (8%).
- Only 6 (8%) of the 80 machines MulVAL predicted as possible to obtain root level privileges on were comprised during the 48 hours.

This test was performed to answer the research question: *how well do attack graphs predict the success or failure of attacks under realistic conditions?* Given the results of this test, the answer to that question is *very poorly*. The primary reasons for the false negatives are reliance on vulnerability scanners and inaccurate interpretation of vulnerabilities provided by them. Further discussions of the results and conditions that may have skewed the results are discussed in section 5.1 below. In section 5.2, recommendations to researchers are given.

5.1 Possible explanations for poor accuracy

There are several possible explanations for the poor performance of the attack graph tool in the test in addition to the issues with vulnerability information and privilege levels described above. Some of these can be seen as excuses for why this particular test produced an unsatisfying result; others are related to the test criteria and the general properties of attack graphs. In the paragraphs below, some presumed objections to the results are stated (in italics) together with a response to the objection.

The attackers in this test were unrepresentative of the threat scenario in which attack graphs are supposed to be used. It is true that the attackers are of a certain type, and they are not representative of the attackers threatening enterprises in general. For example, there were no malware writers, botnet herds, or security-illiterate disgruntled employees involved in this test. However, the type of threat for which MulVAL produces predictions has not been defined, requiring interpretations of its scope. Based on the reasoning provided in articles on MulVAL and similar tools, it is reasonable to assume that attack graphs (and MulVAL) should work when there is a match between the attacks and vulnerabilities modeled and the attacks and vulnerabilities the threat is capable of finding, i.e., when the vulnerabilities fed into MulVAL are those that the attackers might exploit. In this test, in which the attackers were limited to publicly available tools and MulVAL was fed the output of a vulnerability scan, the definitions of the vulnerabilities were clearly matched. Furthermore, even if there was a mismatch and another type of attacker was imagined for the attack graphs, it is reasonable to expect that the predictions would also be indicative for this type of attacker. In other words, even with the wrong type of attackers, it should still be expected that an accurate prediction would correlate with the observations made for the other attackers.

The attack efforts were not independent and randomly distributed. This claim is true. In this test, the same attack may have been attempted multiple times, and the attacks were selected by human agents who (presumably) reasoned about the best course of action before their attempts. They may have spent hours on some machines and dismissed others as impossible to compromise in seconds. It is reasonable to expect that these humans thought in a similar manner to MulVAL and considered the exploitation of existing vulnerabilities within their reach. If this is the case, they should have been more likely to test the attacks that MulVAL predicted as possible and less likely to test the attacks that MulVAL predicted as impossible. Therefore, it may be the case that a randomized sample of attacks would produce a higher proportion of practically unlikely attacks that would be easy to predict as impossible. However, this is a poor explanation for the low prediction accuracy for successful attacks or for attacks predicted as possible: the attackers were more likely to succeed if MulVAL said that the attack was impossible than if MulVAL said it was possible. Additionally, it can be argued that a test with randomly selected attack paths would lack the ecological validity needed to say whether MulVAL works in an operational context, in which attacks are not random.

The attackers' logs may have been erroneous and introduced biased measurement errors. Half of the attackers had their screens recorded. No issues were found with the logs' accuracy, except for some cases in which the attacker did not report a failed attempt, typically when the attacker tried multiple ways to compromise a machine. The omission of failed attempts does not influence the results in this test, in which all of the probed machines were considered interesting targets for the attackers. The logs produced by the attackers probably contained some flaws and missed other additional information. However, based on the comparisons with the screen recordings, it is safe to say that no errors were of a sufficient magnitude to threaten the overall conclusions.

Unrealistically vulnerable networks were used, and this influenced the results. The computer networks used in this test were certainly more vulnerable than the average enterprise’s computer network, with some of the machines not having been updated in a decade. The high number of exploitable vulnerabilities led to an unusually large number of possible attack paths through the computer network. Although this is unrealistic, it is difficult to see why this would cause the poor accuracy of the attack graph tool. On the contrary, the use of well-known vulnerabilities implies that the tool had accurate information and therefore should have produced accurate results.

This was really a test of the vulnerability scanner providing vulnerability information to MulVAL. To some extent, this is a valid objection to the results. The analysis was not made under the premise that perfect information was available. Incomplete scan results contributed to a large portion of false negatives. For example, the scans did not report the CVE-2008-4250 vulnerability in Windows machines, a vulnerability used by attackers in 38 attack paths to compromise 24 machines in this test. However, the result would have been poor even if these vulnerabilities had been reported by the scanner. When the scan results were complemented with all vulnerabilities used by the attackers, MulVAL still missed 27 of 40 successful root-level code execution attacks. In fact, closer inspection shows that, because of problems associated with interpreting the effect of vulnerabilities, none of the predicted attacks correspond perfectly to the attacks performed by the attacker. To predict all successful attacks it is not sufficient to complement the vulnerability scan with missed vulnerabilities; it is also required that an analyst interpret the results and manually set the privileges that can be obtained if the vulnerability is exploited. Such manual adjustments may be possible to do before an analysis, and MulVAL can be improved to guess privilege levels better. However, even if the problem with the analysis had been the input from the vulnerability scanner, it has been repeatedly argued that attack graphs are practically useful because they can be fed input from such vulnerability scans and then make predictions in an automated fashion. Thus, although the vulnerability scanner is important to the results, it makes sense to see it as part of the solution. As a side note, the makers of MulVAL advocate the use of Nessus, another vulnerability scanner, by providing scripts for parsing its output files. In this test, these files were adapted to use Nexpose's output instead. According to the test performed in 2011 by Holm et al. (2011), no significant difference should be expected in accuracy if Nessus were used instead.

Attack graphs are possibilistic, and treating the output as a probabilistic indicator is unfair. As noted above, it is unclear how the results of attack graphs such as MulVAL are supposed to be interpreted. In this test, it was expected that MulVAL's classification could be used as an indicator of the attackers' capabilities. If a truly possibilistic interpretation were to be used instead, an attack would be labeled as possible even if it were highly unlikely that it could be accomplished and impossible only if it were affirmed that it was impossible under the conditions given. In the confusion matrix in Table 1, this would mean that failed attacks could not be used to evaluate the tool because they could be the result of, for example, bad luck or an incompetent attacker. It would also imply that the probability of success of an attack could be any value, including extremely low values such as one percent. First, not even an extreme possibilistic interpretation can explain why 34 of the 119 machines that were considered impossible to compromise as root were actually compromised. To come to a possibilistic result, the input from the vulnerability scanner is corrected and MulVAL's interpretation of it is manually overridden or improved. Second, the designers of MulVAL do not seem to have implemented a possibilistic (worst case) solution. Because it is, of course, possible that a remote exploit yields root-level access, a possibilistic solution ought to have indicated that this is possible rather than indicating that the privileges of some user are obtained.

5.2 Recommendations to researchers

Given these results, four recommendations are provided to researchers interested in attack graphs.

First, we recommend that researchers interpret the results of this study with caution and a positive spirit. Although the results suggest that attack graphs, when used together with a normal vulnerability scanner, fail to predict what a security professional can accomplish, there are several nuisance variables that may have distorted the results. These nuisance variables include the attackers themselves, the attack graph tool used, the vulnerabilities in the computer networks and the vulnerability scanner used. Although it may be hard to see how any realistic configuration of these variables would result in accurate predictions, further tests should be performed. Furthermore, the labeling of privilege levels in MulVAL could be improved to produce better predictions of privilege levels, e.g., by guessing based on the impact vector in the Common Vulnerability Scoring System (CVSS) (Mell et al., 2007), using complementary vulnerability information from other sources or making a possibilistic (worst-case scenario) guess.

Second, there may be techniques and algorithms that could improve the accuracy of attack graphs that should be tested empirically. MulVAL produced approximately 500,000 un-prioritized attack paths for the eight networks included in this test. A visual graph would be impossible for the human eye to comprehend, and it is unclear how it would support decision-making. If attack graphs were prioritized and ranked relative to each other, the output of the analysis might become more accurate and more useful. Several suggestions have been made in this direction, including assessments of critical assets in the graph (Sawilla and Ou, 2008) and probabilistic rankings of the paths (Homer et al., 2010) (Singhal and Ou, 2009). Alternatively, conditions not related to individual vulnerabilities could be used to improve the accuracy, for example, by using quantitative estimates of remote code execution attacks, such as those provided in studies such as (Sommestad et al., 2012). These alternatives ought to be considered after techniques for interpreting the privilege levels attained by exploitations of vulnerabilities are improved.

Third, the accuracy of vulnerability scanners is a serious practical obstacle for attack graphs today. Provided that vulnerability information is interpreted correctly, significant improvements in their accuracy would lead to significant improvements in the accuracy of attack graphs. Research could be directed toward improving vulnerability information and vulnerability scanners to accelerate their progress. Additionally, improvements in the accuracy of attack graphs could be gained through improvements in threat intelligence, e.g., by improving the understanding of the modus operandi of presumed threat agents or which vulnerabilities they are capable of exploiting. With such information, it would be possible to provide the attack graph tools with better data and increase their accuracy, e.g., through combination with probabilistic approaches or other means of ranking attack steps and paths to support decision-making.

A fourth recommendation is to lower the expectations for attack graphs as a practically useful vulnerability prediction or analysis tool. Although no direct claims have been made about the accuracy of attack graph tools in general, or MulVAL in particular, it is easy to get the impression that a vulnerability scan and attack graph analysis will serve as an effective vulnerability prediction and analysis method for decision makers. For example, Ou et al. (2006) start their abstract with, "*[a]ttack graphs are important tools for analyzing security vulnerabilities in enterprise networks*." Roschke et al. (2009) start their abstract with, "*[a]ttack graph is used as an effective method to model, analyze, and evaluate the security of complicated computer systems or networks*." Williams (2008) starts with the declaration, "*[a]ttack graphs are valuable tools in the assessment of network security, revealing potential attack paths an adversary could use to gain control of network assets*." Jajodia (2007) concludes that the TVA tool is "*a powerful approach to global network vulnerability analysis."* Such statements are clearly questionable in light of the results of this test, where attacks are more likely to succeed if they are not predicted by the tool. But, as noted above, attack graphs may become practically useful in the future. Furthermore, despite the difficulty in providing accurate input data, attack graphs could function as a framework on which security theories could be attached, related to each other and synthesized.

6 Conclusions

This test determines that the attack graph tool MulVAL predicts human attackers' successes poorly when used together with the vulnerability scanner Nexpose. Only 8 percent of the machines predicted as possible to compromise were compromised; 29 percent of the machines predicted as impossible to compromise were compromised. This inaccuracy is due to the combination of inaccurate vulnerability scans and improper interpretation of the privileges that vulnerabilities grant. If vulnerabilities are manually added and manually corrected to provide the right privilege level, all but one compromised machine are predicted as possible to compromise.

7 References

Alhomidi, M. a. and Reed, M.J. (2012), "Attack graphs representations", *2012 4th Computer Science and Electronic Engineering Conference (CEEC)*, Ieee, pp. 83–88.

Artz, M.L. (2002), *Netspa: A network security planning architecture*, Massachusetts Institute of Technology, available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q

=intitle:NetSPA+:+A+Network+Security+Planning+Architecture+by#0.

Chu, M., Ingols, K., Lippmann, R., Webster, S. and Boyer, S. (2010), "Visualizing attack graphs, reachability, and trust relationships with NAVIGATOR", *Proceedings of the Seventh International Symposium on Visualization for Cyber Security*, ACM, pp. 22–33.

Heberlein, T., Bishop, M., Ceesay, E., Danforth, M., Senthilkumar, C. and Stallard, T. (2004), "A Taxonomy for Comparing Attack-Graph Approaches", *netsq.com*.

Holm, H., Sommestad, T., Almroth, J. and Persson, M. (2011), "A quantitative evaluation of vulnerability scanning", *Information Management & Computer Security*, Vol. 19 No. 4, pp. 231–247.

Homer, J., Manhattan, K., Ou, X. (Simon) and Schmidt, D. (2010), *A Sound and Practical Approach to Quantifying Security Risk in Enterprise Networks*, *people.cis.ksu.edu*, Kansas, available at: http://people.cis.ksu.edu/~xou/publications/tr_homer_0809.pdf (accessed 24 June 2010).

Homer, J. and Ou, X. (Simon). (2009), "SAT-solving approaches to context-aware enterprise network security management", *IEEE Journal on Selected Areas in Communications*, Citeseer, Vol. 27 No. 3, pp. 315–322.

Ingols, K., Chu, M., Lippmann, R., Webster, S. and Boyer, S. (2009), "Modeling Modern Network Attacks and Countermeasures Using Attack Graphs", *Annual Computer Security Applications Conference*, IEEE, pp. 117–126.

Jajodia, S. (2007), "Topological analysis of network attack vulnerability", *Proceedings of the 2nd ACM symposium on Information, computer and communications security - ASIACCS '07*, ACM Press, New York, New York, USA, p. 2.

Jajodia, S. and Noel, S. (2010), *Advanced cyber attack modeling analysis and visualization*, Rome, NY, available at: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA516716 (accessed 1 April 2014).

Jonathan Bennett AutoIt Consulting Ltd. (2013), "AutoIt Script Editor", available at: http://www.autoitscript.com/site/autoit/ (accessed 13 January 2014).

Lippmann, R. (2002), *Netspa: A network security planning architecture*, *Network Security*, Massachusetts Institute of Technology.

Lippmann, R. and Ingols, K. (2005), *An annotated review of past papers on attack graphs*, Lexington, Massachusetts, available at: http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA431826&amp;Location=U2&amp;doc=GetTRDoc.pdf (accessed 14 September 2010).

Liu, C., Singhal, A. and Wijesekera, D. (2012), "Using Attack Graphs in Forensic Examinations", *2012 Seventh International Conference on Availability, Reliability and Security*, Ieee, pp. 596–603.

Mell, P., Scarfone, K. and Romanosky, S. (2007), *A Complete Guide to the Common Vulnerability Scoring System (CVSS), Version 2.0*.

Noel, S., Elder, M., Jajodia, S., Kalapa, P., O'Hare, S. and Prole, K. (2009), *Advances in Topological Vulnerability Analysis*, *2009 Cybersecurity Applications & Technology Conference for Homeland Security*, IEEE, Washington, DC, pp. 124–129.

Ou, X. (Simon), Boyer, W.F. and Zhang, S. (2013), "MulVAL: A logic-based enterprise network security analyzer", available at: http://www.arguslab.org/mulval.html (accessed 4 March 2014).

Ou, X. (Simon), Boyer, W.W.F. and McQueen, M.A. (2006), "A scalable approach to attack graph generation", *Proceedings of the 13th ACM conference on Computer and communications security*, ACM, Alexandria, Virginia, USA, pp. 336–345.

Ou, X. (Simon), Govindavajhala, S. and Appel, A.W. (2005), "MulVAL: A logic-based network security analyzer", *Proceedings of the 14th conference on USENIX Security Symposium-Volume 14*, USENIX Association, p. 8.

Roschke, S., Cheng, F. and Meinel, C. (2010), "Using vulnerability information and attack graphs for intrusion detection", *2010 Sixth International Conference on Information Assurance and Security*, IEEE, pp. 68–73.

Roschke, S., Cheng, F., Schuppenies, R. and Meinel, C. (2009), "Towards unifying vulnerability information for attack graph construction", *Information Security*, pp. 218–233.

Sawilla, R. and Ou, X. (Simon). (2008), “Identifying critical attack assets in dependency attack graphs”, *13th European Symposium on Research in Computer Security (ESORICS)*, Springer, pp. 18–34.

Singhal, A. and Ou, X. (Simon). (2009), “Techniques for enterprise network security metrics”, *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*, ACM, p. 25.

Sommestad, T., Holm, H. and Ekstedt, M. (2012), “Estimates of success rates of remote arbitrary code execution attacks”, *Information Management & Computer Security*, Vol. 20 No. 2, pp. 107 – 122.

Williams, L. (2008), *GARNET: A graphical attack graph and reachability network evaluation tool*, (Goodall, J.R., Conti, G. and Ma, K.-L.,Eds.)*5th International Workshop, VizSec 2008,* Springer Berlin Heidelberg, Cambridge, MA, USA.

Zhang, S., Ou, X. (Simon), Singhal, A. and Homer, J. (2011), *An empirical study of a vulnerability metric aggregation method*, *csrc.nist.gov*, available at: http://csrc.nist.gov/staff/Singhal/xou-anoop-workshop2011-paper.pdf (accessed 27 July 2011).