Vulnerability management tools for COTS software - A comparison

S.M. Welberg
Student MBI at University of Twente
postbus 217
7500 AE Enschede

s.m.welberg@student.utwente.nl

ABSTRACT

In this paper, we compare vulnerability management tools in two stages. In the first stage, we perform a global comparison involving thirty tools available in the market. A framework composed of several criteria based on scope and analysis is used for this comparison. From this global view of the tools, we detected that only three tools perform correlated analysis. Correlated analysis can be done in two ways: (i) correlation of scanning results with the output from other security devices such as firewall and intrusion detection systems, or (ii) correlation between vulnerabilities composing attack scenarios. Although both correlations add value to vulnerability management, the latter is especially important to unveil stepping stones which could be exploited by attackers. The comparison shows that two out of three tools perform correlation of the second type but scalability and the amount of manual input required seems to be their biggest problems.

Keywords

Vulnerability management, network architecture, tools, Commercial-Off-The-Shelf (COTS), Vulnerability correlation, Attack tree, Attack graph, correlated analysis

1. INTRODUCTION

A lot of organizations use Commercial-Off-The-Shelf (COTS) products nowadays. There are several reasons why COTS software is attractive. First, the time to market is much shorter for COTS software than for in-house developed projects. Second, the costs for in-house developed projects are usually much higher. Third, organizations not only depend on organization-specific information systems, but also on generic infrastructures. The software for these infrastructures is often provided by COTS vendors [7, 10], for example operating systems provide an infrastructure for an organization. Thus, often organizations depend on COTS software.

Vulnerability management is becoming very important in organizations; the exploitation of vulnerabilities is costing organizations money every year. According to a report [44] by the CSI, compared to 2006, the average annual loss due to security problems has doubled in 2007. In 2007 alone, 6704 vulnerabilities have been added to the NVD database. This is 3% more than the total of 2006 [41]. Additionally there are other factors why organizations must perform vulnerability management:

- Not even COTS vendors know how secure their products are [5].
- The consequences of vulnerabilities are left with product buyers because vulnerabilities are externalities [5].

This means that COTS products are likely to contain vulnerabilities and organizations (product buyers) are the ones who need to manage them.

Although vulnerability management is necessary, it is not a straight forward task and requires balance. Security is always a trade-off between the costs of protection and the benefits from preventing successful exploitation of vulnerabilities by attackers. Thus, vulnerability management is not just a matter of applying patches straightaway, because they can introduce other vulnerabilities additionally the amount of patches is so huge that it is often too costly to apply all of them. As a consequence an organization needs to choose between fixes based on cost/benefit and need to prioritize the vulnerabilities they want to mitigate.

Vulnerability management is not easy to do, because (i) there are so many vulnerabilities, (ii) low-risk vulnerabilities can be stepping-stones that lead to high-risk attack scenarios that can affect valuable assets [25, 62]. (iii) Vendors use vendor-specific nomenclature and scores for vulnerabilities, (iv) often companies have large infrastructures with a high amount of network devices all being subject to vulnerabilities, and (v) a single vulnerable point can compromise the security of the whole network.

According to all the above arguments, it is very difficult for organizations to manage vulnerabilities by hand, so tools and standards are needed to help them. Furthermore, it is important that vulnerability management tools provide correlated information. This correlation can occur in the form of attack scenarios or in the form of checking information from different security devices, such as Intrusion Detection Systems (IDS's) or firewalls.

There are many standards rising in the COTS software vulnerability sector. It is now possible to look up the vulnerabilities of given COTS systems from comprehensive vulnerability libraries [29]. A standard for providing uniform names across vulnerability reporting sources is the Common Vulnerabilities and Exposures (CVE) [28, 34]. The Common Weakness Enumeration specification [35], currently maintained by the MITRE Organization, provides a common language of discourse for discussing, finding and dealing with causes of software security vulnerabilities as they are found in code, design, or system architecture. There are a lot more standards for vulnerability management, which are elaborated in more detail in subsection 2.1.3.

Many tools have been developed for managing vulnerabilities within an organization, or on one single host. Some of these tools report in terms of standards. Since there are a lot of tools, it is not clear which of the tools use which standards and how do these tools manage vulnerabilities. A lot of questions arise here, e.g. do these tools manage vulnerabilities for an entire architecture consisting of multiple COTS software packages or just on a single host? There is no good overview of different vulnerability management tools.

In this paper, we aim to compare vulnerability management tools and our final objective is to understand how such tools address the challenge of correlation among information about vulnerabilities. This paper is structured as follows. First, in Section 2, a framework for the global comparison is described. Thirty tools are selected and the results are presented and discussed in Section 3. In Section 5 we select from the thirty tools, the ones which provide any sort of correlated analysis for a focused comparison (Section 6). Related work is reviewed in Section 7 and final conclusions are drawn in Section 8.

2. FRAMEWORK FOR GLOBAL COMPARISON

The criteria for the global comparison are divided into scope and analysis.

2.1 Scope

The scope of a tool is the range of a certain tool, and it is divided into multiple sub criteria elaborated in subsections 2.1.1 to 2.1.4.

2.1.1 Software Platforms

Vulnerabilities can and do reside in all platforms such as Windows, Unix, Linux, HP/UX, Solaris, and Mac OS.

2.1.2 Magnitude

The magnitude is the scale of the vulnerabilities managed by a tool. It can either be that a tool is only capable of identifying the vulnerabilities in a single host, or the tool is able to identify vulnerabilities in an entire architecture within an organization.

2.1.3 Standards

The following standards are the most common standards for vulnerability management and we will explain them below:

- XCCDF
  The Extensible Configuration Checklist Description Format is a specification language for writing security checklists, benchmarks, and related kinds of documents [40].
- CVE
  The Common Vulnerabilities and Exposures is a standard for providing uniform names across vulnerability reporting sources [34]. E.g. CVE-2007-3168 Summary: A certain ActiveX control in the EDraw Office Viewer Component (edrawofficeviewer.ocx) 4.0.5.20, and other versions before 5.0, allows remote attackers to delete arbitrary files via the DeleteLocalFile method.
- CPE
  The Common Platform enumeration is a structured naming scheme for information technology systems, platforms, and packages, providing common names for all software systems. This makes it a lot easier to link the different vulnerabilities to systems, since everybody speaks about the same system when CPE is used. The CPE structure: cpe:/ {part} : {vendor} : {product} : {version} : {update} : {edition} : {language} [12]. An example of a CPE: cpe:/o:microsoft:windows-nt:2000:sp4:pro
- CCE
  The Common Configuration enumeration provides unique identifiers to system configuration issues in order to facilitate fast and accurate correlation of configuration data across multiple information sources and tools. A CCE list item containts: the CCE identifier number, a description, conceptual parameters, associated technical mechanisms and citations [33]. In Table 1 an example of a CCE is shown.:

Table 1 – CCE Example

CCE NR	CCE Definition	CCE Platform	CCE Parameters	CCE Technical measures
CCE-871	The "maximum password age" policy should meet minimum requirements.	Microsoft Windows Vista	(1) number of days	(1) defined by Local or Group Policy

- CVSS
  The Common Vulnerability Scoring System provides a standardized rating for vulnerabilities present in the NIST (NVD) database. CVSS scores can range from 0 (low severity) to 10 (high severity) [20]. For an example CVE-2007-3168 has a CVSS score of 7.8 (High).

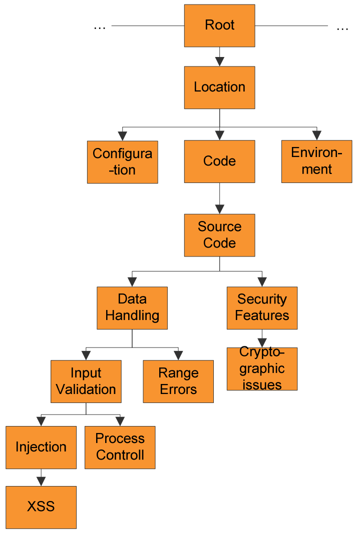

Figure 1 – Portion of CWE Structure from NIST [39].

2.1.4 Type of vulnerabilities

Vulnerabilities can reside in three different areas: configuration, the source code and the environment (see Figure 1**)** of software systems.

- Configuration

  Vulnerabilities in the configuration are caused by configuration errors in the software which can be exploited by attackers, for example, for remote code execution on the attacked machine.

- Source Code

  Vulnerabilities in the source code are caused by errors in the source code of software. These errors could, for example, make it possible to cause a buffer overflow error which can harm the attacked system.

- Environment

  Environmental vulnerabilities are caused by direct or indirect modification of environment variables by attackers to exploit software. For example an attacker which changes an environment variable of the system in order to gain access to that system. The variable is changed by executing code on that software system.

2.2 Analysis

The criteria in this subsection elaborate the kind of analysis a tool provides for finding and assessing vulnerabilities in COTS Based Systems (CBSs).

2.2.1 Kind of analysis

We distinguish four types of analysis performed by vulnerability tools: compliance checking, patch management, vulnerability scanning and correlated analysis, as explained next. It is worthy to mention that when a tool performs one type of analysis, this does not mean that another kind of analysis is excluded.

- Compliance checking

  Nowadays there are several security compliance frameworks and laws, such as HIPAA, SOX, ISO17799, GBLA, FDCC, FISMA, PCI. Tools can automate the process of checking compliance with these frameworks.

- Patch Management

  Tools falling under this only give a list of e.g. patches installed, software that is installed, missing patches, etc. So only an inventory of a host is created, but no additional assessment is done with this information.

- Vulnerability scanning

  Tools which comply to the vulnerability scanning category assess the vulnerabilities in a system or architecture and report vulnerabilities found. These vulnerabilities can be reported in terms of CVE's or not.

- Correlated analysis

  It is also possible that a tool correlates or aggregates vulnerabilities with other information (or vulnerabilities) in order to perform better vulnerability analysis. There are two types of vulnerability correlation currently discussed in the literature. First, correlation between different types of security devices like IDSs and firewalls. Second, correlation between several vulnerabilities which can be used as stepping-stones for an attack scenario.

2.2.2 Types of results

Many tools create an overview of the combined vulnerability score of a system in terms of how vulnerable a system or architecture is. This can be either done qualitatively, e.g. with a color ranging from green to red, or quantitatively with a number which can be vendor-specific and/or based on a CVSS score.

2.2.3 Information available

This criterion indicates whether enough information was available for conducting the comparison.

3. GLOBAL COMPARISON

The tools which are included in the comparison are listed in Table 2. The main selection criterion for the tools is whether they use some kind of standard and preferably CVE's. The reason for this is the fact that the vendor-specific nomenclature is one of the reasons why vulnerability management is not easy to do. These standards allow sharing of vulnerability information among the community that needs to manage them.

A cross in a box means that a tool qualifies to a certain criterion. An empty box represents a tool does not comply with a certain criterion. A question mark shows that not enough information was available to exclude or approve a certain criterion. Per criterion remarks on the comparison are given below.

Almost all tools support multiple platforms. Apart from the standard platforms, often also HP/UX, Solaris and Mac OS are supported. This relates to the impact criterion, which shows that almost all tools support multiple systems and thus are capable to manage vulnerabilities in an entire architecture. Just three tools (Threatguard's Secutor Prime Free [57], Belarc's Advisor [8], and the CIS-CAT tool by CIS [14]) are only capable of scanning a single host. The other tools are capable of scanning multiple hosts. A remark here is that some of the tools do have scanners which can only scan a single host but the information is then collected by a central unit which analyses this information in an enterprise-wide manner to draw conclusions.

The comparison gives some information about standards. A lot of the tools use or comply to some kind of standard, but the information regarding to what standards they comply to is often hard to find.

Usually it is very clear which types of configuration and source code vulnerabilities the tools can uncover. However, it is difficult to find out whether the tools can detect environmental vulnerabilities. Only for the Security center by Nessus [54] it was clear that their tool is able to find environmental vulnerabilities. This is due to the fact that they also use information by Intrusion Detection Systems in order to create reports on vulnerabilities.

For the analysis criteria there were also some remarks. First compliance to laws or quality frameworks is very important in business, there is much regulatory pressure; this is reflected by the number of tools which support compliance checking to the known compliance frameworks or information security laws, such as SOX, HIPAA, ISO 17799, and PCI.

Table 2: Tool comparison

		Platform			Magni-tude		Standards					Type of vul-nerabilities			Analysis				Types of results		info
Vendor	*Name*	*Windows*	*Linux*	*Other*	*single*	*enterprise*	*XCCDF*	*CVE*	*CPE*	*CCE*	*CVSS*	*configuration*	*source code*	*environment*	*Compliance checking*	*Patch Management*	*Vulnerability scanning*	*Correlated analysis*	*Quantitative*	*Qualitative*	*Information available*
Tenable	Nessus [54]	X	X	X		X	X	X			X	X	X	?	X	X	X		X	X	X
	Security Center 3 [55]	X	X	X		X	X	X			X	X	X	X	X	X	X	X	X	X	X
GFI	LANguard Vulnerability manager 8 [21]	X	X	X		X		X				X	X	?	X	X	X			X	X
Secure elements	C5 Platform [47]	X	X	X		X	X	X	X	X	X	X	X	?	X	X	X		X	X	X
Threatguard	Secutor Prime Free [57]	X			X		X	X	X	X	X	X		?	X	X	X		X	X	
	Secutor prime Magnus [58]	X				X	X	X	X	X	X	X		?	X	X	X		X	X	
	Vulnerability Management System [59]	X	X	X		X	X	X	X	X	X	X	X	?	X	X	X		X	?	
Belarc	Belarc NIST Advisor [8]	X			X			X		X	X	X		?	X	X			X		
IBM	Internet Scanner [24]	X	X	X		X		X				X	X	?		X	X			X	
	Tivoli Security compliance manager [23]	X	X	X		X	?	?	?	?	?	X	?	?	X	X	?	?		X	
CA	Vulnerability manager r8.3 [13]	X	X	X		X	?	?	?	?	?	X	X	?	X	X	X		?	?	
Qualys	QualysGuard Enterprise [42]	X	X	X		X		X			X	X	X	?	X	X	X		X	X	X
Skybox	Secure [52]	X	X	X		X		X			X	X	X	?	X	X	X	X	X	X	
Amenaza	Secure*IT*ree [3]	X	X	X		X												X	X		X
Gideon Technologies	SecureFusion Portal [22]	X	X	X		X	X	X	X	X		X	X	?	X	X	X			X	
CIS	CIS-CAT [14]	X	X	X	X		X	?	?	?	?	X	?	?	X	X	?	?	?	?	
Configuresoft	Enterprise Configuration Manager (ECM) [16]	X	X	X		X	?	?	?	?	?	X		?	X	X			X		
Rapid7	NeXpose Unified vulnerability management (UVM) [43]	X	X	X		X		X				X	X	?	X	X	X		X	X	X
Core Security	Core Impact 6.0 [17]	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	
eEye	Retina Network Security Scanner [18]	X	X	X		X		X				X	X	?	X	X	X			X	X
McAfee	Policy Auditor and Remediation Manager (PARM) [31]	X	X	X		X		X		X		X		?	X	X			?	?	
	Foundstone [30]	X	X	X		X	?	?	?	?	?	X	X	?	X	X	X			X	
NetIQ	Risk and Compliance Center [38]	X	X	X		X		X			X	X	X	?	X	X	X		X	X	X
nCircle	Configuration Compliance Manager [37]	X	X	X		X		X				X		?	X					X	X
Shavlik	NetChk Compliance [49]	?	?	?		X	?	?	?	?	?	X		?	X				?	?	
	ARM [48]	X	?	?		X	?	?	?	?	?	X	X	?		X			?	?	
Microsoft	Microsoft baseline security analyser [32]	X				X						X		?		X			X		
Cisco	Cisco IntelliShield Alert manager [15]	X				X		X			X	X		?	X		X		X		
Lumension Security	Patchlink Scan [26]	X	X	X		X		X				X	X	?	X	X	X			X	X
BgFix	Discovery 7 [9]	X	X	X		X	?	?	?	?	?	X	X	?	X	X			?	?	

nCircle's Configuration Compliance Manager [37], NetIQ's Risk and Compliance Center [38], and NetChk Compliance [49] by Shavlik even only support compliance checking. Second, almost all the tools provide patch management, mostly in combination with compliance checking. Third, vulnerability scanning is in all but one case combined with patch management. The only case in which it is not combined is in the Intellishield alert manager by Cisco [15]. And at last the correlation criterion is one to which only three of the thirty tools comply to.

With respect to the reporting criterion, we found that although more tools report qualitatively, the difference is not significantly. Also some tools support both. The tools which do support quantitative reporting often fulfill this criterion by using the CVSS standard for scoring vulnerabilities. Only Amenaza's Secur*IT*ree [2] and Rapid7's NeXpose Unified vulnerability management (UVM) [43] have their own way of creating numerical scores for vulnerabilities.

A few remarks on the global comparison must be made. First, there are three tools in the comparison which are freeware these are Tenable's Nessus [54], Threatguard's Secutor Prime free [57] and Belarc's NIST Advisor [8]. Second, the amount of information was often limited so it was hard to find whether the tools did or did not comply with the comparison criteria.

4. CRITERION FOR FOCUSED COMPARISON

The comparison is structured as follows. First a description of the tools is given (section 5). Then a detailed description of the way the tools correlate vulnerabilities is given (sections 6.1 to 6.3) and finally conclusions on the comparison are drawn (section 6.4).

5. TOOLS FOR FOCUSED COMPARISON

In this section we present the tools selected for the focused comparison. These tools are the only three tools in the global comparison (see Table 2) that support correlated analysis.

5.1 Tenable Security Center 3.0

The Tenable Security Center is a security suite which provides continuous, asset-based security and compliance monitoring. Tenable has two types of scanners, their active and passive scanner. The Nessus Vulnerability Scanner is an active scanner that provides a snapshot of network assets, their vulnerability exposure, their configuration, and if they contain sensitive data [55]. The Passive Vulnerability Scanner behaves like a security motion detector on the network. It maps new hosts and services as they appear on the network and monitors for vulnerabilities 24/7.

All the information gathered by the scanners can be used to create reports. There is a possibility to extensively drill down content making it possible for several audiences to get to the right information.

5.2 Skybox Secure

The secure suite by Skybox Security is based on the Skybox view platform. Skybox Secure automates and manages the risk lifecycle by identifying threat exposures, quantifying risk, and managing countermeasures. It collects and normalizes security information while producing key performance indicator and actionable intelligence. The Skybox secure package consists of three modules: Threat Alert Manager, Risk Exposure Analyzer, and Security Profile Advisor.

The Threat Alert Manager normalizes and correlates threat and alert feeds, patch information, vulnerability data, and business information A ticketing system enables ticket creation and assignment that can be integrated with existing systems. Threat Alert Manager can also create and assign remediation tickets automatically to the appropriate personnel [51].

The Risk Exposure Analyzer continuously quantifies risk and prioritizes vulnerabilities by taking into account business logic, threats, security controls, and network policies. Security professionals can understand and analyze the business impact of threats, and simulate the most cost-effective remediation alternatives [51].

Finally, the Security Profile Advisor automates the collection of security and compliance data from multiple systems, calculates Key Performance Indicators (KPI) and presents security advisories through actionable dashboards and reports, by providing threat and remediation metrics [51].

5.3 Amenaza Secur*IT*ree

The Secur*IT*ree application by Amenaza provides a tool to assess risks in an organization. It uses an attack-tree method for assessing how an asset in an organization can be attacked by an attacker, what harm he does with his attack, and what measures need to be taken to become immune to that attack [3]. The Secur*IT*ree application is not able to detect vulnerabilities itself, it is only able to correlate them based on the attack-tree which has to be created.

6. FOCUSED COMPARISON

In this section the comparison of the tools selected in the previous section (section 0) is conducted. The selected tools are compared in terms of the correlated analysis they provide one by one and put in separate sub sections. At last we provide a conclusion based on the comparison.

6.1 Tenable Security Center 3

The tenable security center uses a log correlation engine for the correlation of vulnerability information. This correlation engine correlates vulnerability information with Intrusion Detection System (IDS) information. The tool is capable of using information from different kinds of IDSs, like the commercially available snort IDS .This correlation is done automatically.

The real-time correlation is performed by the Security Center, which has knowledge of the state of each server's vulnerabilities and automatically correlates known attacks against known vulnerabilities. This reduces the number of IDS alerts by exclusion of the false positive attacks [56].

The vulnerability information used by the Nessus Security Center is gathered from two sources, the active and passive vulnerability scanner. The advantage of the passive scanner is that it monitors the network 24-7, therefore data is always up-to-date. This is crucial for the entire process. Assuming for example, that the vulnerability information is not up-to-date, an IDS alert could then be discarded as a false-alarm but, in fact, it was not. Such a situation must, at all times, be avoided.

6.2 Skybox Secure

Skybox Secure is based on attack graph analysis. It has a four step sequence for determining how vulnerable an organization's network is to attacks. However, only the first two steps are related to correlation of vulnerabilities and will therefore be described here [50].

In the first step Skybox Secure creates an integrated security model in order to capture the business and IT environment context. The model consists of threats, network information, vulnerabilities data, and business assets. The threats need to be defined by the users and are, for example possible starting points, attacker skills, and likelihood of attack. There is however an initial set of threats defined after installation. Network information is automatically collected and consists of network topology, routers, firewalls, servers, and other hosts. For each gateway, namely routers and firewalls, routing information and access filtering rules are also collected. Business assets can be mapped in groups which can consist of a group of servers in several segments. And finally, the business impact rules need to be defined and linked to an asset in order to specify the potential damage to a certain asset. Different types of damage values can be modeled: confidentiality, integrity or availability, or regulatory compliance [50].

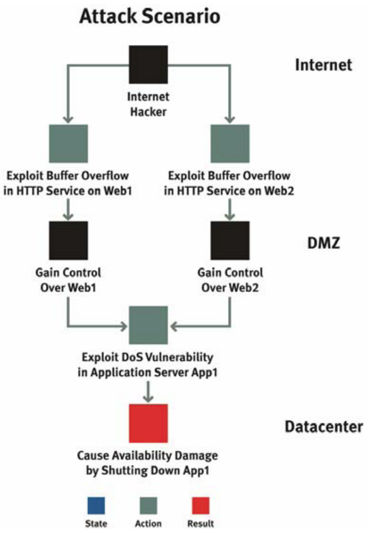
Attack Scenario
Internet
Internet Hacker
Exploit Buffer Overflow in HTTP Service on Web1
Exploit Buffer Overflow in HTTP Service on Web2
DMZ
Gain Control Over Web1
Gain Control Over Web2
Exploit DoS Vulnerability in Application Server App1
Datacenter
Cause Availability Damage by Shutting Down App1
State
Action
Result

Figure 2 – Example of attack Scenario Skybox Secure [50]

In the second step, attack scenarios are simulated in order to identify exposures. This is done by the Attack Simulation Engine developed by Skybox Security. The engine simulates all possible attack scenarios for all threats. An attack scenario represents a set of actions that can be performed by an attacker given the specific context of an organization's network. **Figure 2** represents an example of such an attack scenario. This scenario represents two web servers in the DMZ segment and one application server in the datacenter segment.

Based on the attack scenario, vulnerabilities are classified in the following three categories: Directly exposed, indirectly exposed or mitigated. Directly exposed vulnerabilities pose an immediate risk to security and can be exploited directly by a threat. Indirect vulnerabilities, on the other hand, can only be exploited after a direct vulnerability has been exploited. Mitigated vulnerabilities cannot be exploited due to existing security measures [50].

Attack graphs are known to be badly scalable [25] since the amount of attack paths increases exponentially when the number of nodes in a network increases. According to Lippmann et al. [25], Skybox patent suggests that the complexity of their algorithm to find attack scenarios from a host representing the attacker to all other hosts, scales to N^4 where N is the number of nodes in a network. This is significant for networks consisting of more than a few hundred nodes, and thus scalability for the solution provided by Skybox Security is not very high. A way to improve this would be to combine the nodes with the same configuration which reside in the same subnet and therefore share the same vulnerabilities [25]. This way the amount of nodes could be kept reasonably small, but it is not clear whether this is possible with the Skybox Secure tool.

6.3 Amenaza Secur*IT*ree

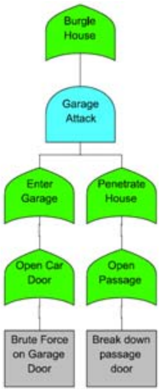

Figure 3 – Attack scenario example Amenaza's Secur*IT*ree

The tool by Amenaza is based on attack trees [45] which use a graphical, mathematical construction to model attacks. The model includes estimates of the resources needed to carry out specific attacks, the impact of those attacks on the victim and the benefits realized by the attacker [1]. The following information is required by the model before it can be analyzed: attacker behavior (scenarios), behavioral indicators, impact indicators, capabilities of attackers, propensity of attack [2].

An examle attack scenario (attack tree) is shown in Figure 3. At the top of this scenario the goal of the attacker is stated. An attack tree shows a logical breakdown of the various options available to an attacker. By performing the exploits associated with one or more leaf level events the attacker can achieve the root level goal. The leaf with the round top (*Garage Attack*) represents an AND leaf. So both the garage must be entered and the house must be penetrated in order to successfully attack the garage. The OR nodes (e.g. *Open passage)* represent that only one of the exploits is necessary for reaching a higher level goal [1].

With this information in place the behavioral indicators are determined. These indicators show how likely certain leafs are to happen. This is done by selecting classes of resources, assets or traits that are required to carry out the specific exploit. These values must be estimated by the analyst since they are often not at hand. It is necessary that all resource types and leafs have behavioral indicators [1].

Then impact indicators which model the consequences of an attack need to be defined. Damage from an exploit may be not too high but whenever another exploit happens the damage is accumulated towards the root node. These impact indicators can reflect loss of money, damage to reputation or even casualties.

Finally the capabilities and the motivation of the attackers need to be modeled. The capabilities are for example the money or knowledge which an attacker has at his disposal. The motivation can be important for a more accurate estimation of the risk an organization is exposed to [1].

After the final model is completed the attack tree is analyzed, this is done automatically based on all the information which is put into the model. The outcome is the risk that a certain attack is carried out by an attacker. Adding up all the risks of the different scenarios gives the total risk figure for an organization [2].

6.4 Conclusion from focused comparison

All the tools have a different way of correlating vulnerabilities. Skybox's Secure and Amenaza's SecurITree both correlate vulnerabilities with each other by creating attack paths or scenarios as they call them. SecurITree's scenarios are from the viewpoint of the attackers whereas the scenarios from Secure are from the organization's viewpoint. The fact that SecurITree needs a lot of manual work and estimation to complete the model can make this tool error prone and time consuming. The total accuracy of this tool is purely based on the accuracy of the estimations. The problem with Skybox Secure and also with Secur*IT*ree is that scalability is quite low (few hundred nodes for Secure, Amenaza possibly even less because of the effort needed for development of an attack scenario) so it cannot be used in very large networks, which could benefit the most from such tools. The Tenable tool correlates vulnerabilities in a completely different way. It shows correlated vulnerabilities with IDS alerts in order to see which vulnerabilities an organization is exposed to and which are actually being exploited at a certain moment in time.

All in all also low-risk vulnerabilities can be very important to mitigate, there seems to be a gap between what is needed by organizations to better manage vulnerabilities, i.e. a high level picture of attack scenarios generated automatically done in a scalable way, and what tools available in the market can deliver. However, further work to analyze even a more extensive set of tools is needed to confirm this claim.

7. RELATED WORK

We divided the relevant related work for this paper into two parts, which we will discuss below. First, surveys and comparisons of tools. Second, correlation of vulnerabilities and correlation between vulnerability information and of output from several devices.

There are some surveys and comparisons of vulnerability management tools. However, they do not cover both aspects of correlation mentioned here, or are too superficial. An academic survey on buffer overflow written by Wilander and Kamkar [61] compares different tools for the prevention of dynamic buffer overflows. Another survey, written by Brackin [11], explained the steps of vulnerability management and discusses different tools for all the distinguished steps, however, the author looks at totally different criteria compared to this paper. Finally a book chapter [27] written by Manzuik et al. evaluate a selection of vulnerability management tools based on a framework they constructed, they do not take the correlation of vulnerabilities (attack scenarios) into account. A non-academic source of information is the SC magazine [46] which provides comparisons for commercial vulnerability management tools.

Correlation of vulnerabilities is not new. The method which Skybox and Amenaza use is based on the methodology of attack trees and attack graphs. In 1991 Weiss published a paper [60] describing threat logic trees, this is the basis of the attack trees as we know them now. In 1994 Amoroso [4] detailed a modeling concept he called threat trees. More recently, Bruce Schneier [45] popularized the idea, and renamed it to attack trees. Other researchers have continued to develop the idea of tree based, threat analysis models [19, 36]. Lippmann and Ingols have summarized all work done on attack graphs until 2005 in a paper [25]. Another paper which gives a good overview of attack graphs and tools is written by Swiler [53].

Correlation of IDS alerts with other kinds of information to reduce the amount of false positives is often referred to, in the literature, as "log Correlation". A good overview can be found in the book Intrusion Detection by Bace [6]. Gula, the CTO of Tenable Network security has written a report on vulnerability correlation with IDS event, he discusses nine possible cases that can happen during correlation of IDS alerts with vulnerability information.

8. CONCLUSION

In this paper we reviewed why vulnerability management is important for organizations and argued why it is valuable when tools support correlated analysis. We performed a two-stage comparison for COTS based vulnerability management tools.

First, thirty tools, both commercial and free-ware, were compared (Table 2). We found that standards are well supported and compliance seems to be a major selling point. However, only three tools supported correlated analysis. Second, the set of three tools which supported correlated analysis were the scope for a focused descriptive comparison. We found that one tool correlates the output of vulnerability scanning with the output from other security devices, and two tools correlate vulnerabilities. However, the comparison shows that scalability and the amount of manual input required are the biggest concerns at this moment for tools supporting correlated analysis.

9. ACKNOWLEDGEMENTS

I would like to thank Virginia Nunes Leal Franqueira for her extensive work as my supervisor for writing this paper.

10. REFERENCES

1. Amenaza, *Fundamentals of Capabilities-based Attack Tree Analysis*. 2005, Amenaza Technologies Limited: Calgary, Alberta. p. 14.
2. Amenaza, *Advanced Attack Tree Analysis*. 2006, Amenaza Technologies Limited: Calgary, Alberta. p. 13.
3. Amenaza. *Amenaza Technologies Limited*. 2007, Accessed 3-12-2007, http://www.amenaza.com/.
4. Amoroso, E.G., *Fundamentals of computer security technology*. 1994, PTR Prentice Hall: Englewood Cliffs, N.J. p. 15-29.
5. Anderson, R. and T. Moore, *Information Security Economics – and Beyond*, in *Advances in Cryptology - CRYPTO 2007*. 2007. p. 68-91.
6. Bace, R.G., *Intrusion detection*. 1999, Indianapolis, IN: Macmillan Technical Pub.
7. Basili, V.R. and B. Boehm, *COTS-based systems top 10 list*. Computer, 2001. **34**(5): p. 91-95.
8. Belarc. *Belarc Advisor - Free personal PC audit*. 2007, Accessed 20-11-2007, http://www.belarc.com/free_download.html.
9. BigFix. *Discovery7*. 2007, Accessed 04-12-2007, http://www.bigfix.com/products/coreservices/discovery7.php.
10. Boehm, B. and C. Abts, *COTS integration: plug and pray?* Computer, 1999. **32**(1): p. 135-138.
11. Brackin, C., *Vulnerability Management:Tools, Challenges and Best Practices*. 2003, SANS Institute. p. 17.
12. Buttner, A. and N. Ziring, *Common Platform Enumeration (CPE) – Specification*. 2007, MITRE Corporation. p. 1-34.
13. CA. *CA Vulnerability Manager*. 2007, Accessed 12-12-2007, http://ca.com/us/products/product.aspx?ID=4707.
14. CIS. *Center for Internet Security*. 2007, Accessed 10-11-2007, http://www.cisecurity.org/ngtoolmembers.html.
15. Cisco. *Cisco IntelliShield Alert Manager Service - Cisco Systems*. 2007, Accessed 03-12-2007,

http://www.cisco.com/en/US/products/ps6834/serv_group_home.html.

16. Configuresoft. *ECM*. 2007, Accessed 12-11-2007, http://www.configuresoft.com/ecm.aspx.

17. CoreSecurity. *Core Security*. 2007, Accessed 1-12-2007 (not reachable), http://www.coresecurity.com.

18. eEye. *Retina Network Security Scanner*. 2007, Accessed 14-11-2007, http://www.eeye.com/html/products/Retina/.

19. Evans, S., et al., *Risk-based systems security engineering: stopping attacks with intention.* Security & Privacy Magazine, IEEE, 2004. **2**(6): p. 59-62.

20. FIRST. *Common Vulnerability Scoring System (CVSS)*. 2007, Accessed 2007 10-10-2007, http://www.first.org/cvss/index.html.

21. GFI. *Vulnerability management: Network vulnerability assessment, scanning and fixing with GFI LANguard*. 2007, Accessed 20-10-2007, http://www.gfi.com/vulnerabilitymanager/.

22. Gideon. *Gideon Technologies*. 2007, Accessed 27-10-2007, http://www.gideontechnologies.com/SecureFusion_portal.asp.

23. IBM. *IBM Tivoli Security Compliance Manager*. 2007, Accessed 10-12-1007, http://www-306.ibm.com/software/tivoli/products/security-compliance-mgr/.

24. IBM. *Internet Scanner*. 2007, Accessed 15-11-2007, http://www.iss.net/products/Internet_Scanner/product_main_page.html.

25. Lippmann, K.P. and K.W. Ingols, *An Annotated Review of Past Papers on Attack Graphs*. 2005, Massachusetts Institute of Technology: Lexinton, Massachusetts. p. 39.

26. Lumension. *Patchlink Scan - Identifies and Inventories Assets with Certified Network-based Scanner*. 2007, Accessed 03-12-2007, http://www.lumension.com/vulnerability-management.jsp.

27. Manzuik, S., K. Pfeil, and A. Gold, *Vulnerability Management Tools*, in *Network Security Assessment: From Vulnerability to Patch*. 2006, Syngress. p. 171 - 188.

28. Martin, R.A., *Managing vulnerabilities in networked systems.* Computer, 2001. **34**(11): p. 32-38.

29. Martin, R.A. *Managing vulnerabilities in your commercial-off-the-shelf (COTS) systems using an industry standards effort*. in *Digital Avionics Systems Conference, 2002. Proceedings. The 21st*. 2002.

30. McAfee. *McAfee - Enterprise - McAfee Foundstone Enterprise*. 2007, Accessed 10-12-2007, http://www.mcafee.com/us/enterprise/products/vulnerability_management/foundstone_enterprise.html.

31. McAfee. *McAfee Policy Auditor and Remediation Manager*. 2007, Accessed 10-12-2007, http://www.mcafee.com/us/local_content/datasheets/1pa_rm_0407.pdf.

32. Microsoft. *Microsoft Baseline security analyzer (MBSA)*. 2007, Accessed 01-12-2007, http://www.microsoft.com/technet/security/tools/mbsahome.mspx.

33. MITRE. *Common Configuration Enumeration (CCE)*. 2007, Accessed 10-10-2007, http://cce.mitre.org/.

34. MITRE. *Common Vulnerabilities and Exposures (CVE)*. 2007, Accessed 09-10-2007, http://cve.mitre.org.

35. MITRE. *Common Weakness Enumeration (CWE)*. 2007, Accessed 2007 09-10-2007, http://cwe.mitre.org.

36. Moore, A., R. Ellison, and R. Linger, *Attack Modeling for Information Security and Survivability*. 2001, CERT.

37. nCircle. *nCircle Pducts - Configuration Compliance Manager*. 2007, Accessed 15-12-2007, http://www.ncircle.com/index.php?s=products_ccm.

38. NetIQ. *NetIQ Products - Configuration Management - Risk & Compliance center*. 2007, Accessed 25-10-2007, http://www.netiq.com/products/rcc/default.asp.

39. NIST. *CWE structure*. 2007, Accessed 2007 10-10-2007, http://nvd.nist.gov/cwe.cfm.

40. NIST. *Extensible Configuration Checklist Description language Format (XCCDF)*. 2007, Accessed 2007 10-10-2007, http://nvd.nist.gov/xccdf.cfm.

41. NIST. *National Vulnerability Database Home*. 2008, Accessed 07-01-2008, http://nvd.nist.gov/.

42. Qualys. *GualysGuard Enterprise*. 2007, Accessed 1-12-20007, http://www.qualys.com/products/risk_compliance/qgent/.

43. Rapid7. *Nexpose Unified Vulnerability Management*. 2007, Accessed 16-11-1007, http://www.rapid7.com/nexpose/uvm.jsp.

44. Richardson, R., *CSI Computer Crime and Securiy Survey*. 2007, CSI.

45. Schneier, B., *Attack Trees.* Dr. Dobb's Journal, 1999. **24**(12): p. 9.

46. SCMagazine. *Security News and Security Product Reviews - SC Magazine*. 2007, Accessed 02-01-2008, http://www.scmagazine.com.

47. SE. *Secure Elements - C5 Compliance platform overview*. 2007, Accessed 1-12-2007, http://www.secure-elements.com/products/C5_Platform_Overview.htm/.

48. Shavlik. *Shavlik - ARM*. 2007, Accessed 13-12-2007, http://www.shavlik.com/active_vulnerability_management.aspx.

49. Shavlik. *Shavlik - NetChk Compliance*. 2007, Accessed 10-12-2007, http://www.shavlik.com/netchk-compliance1.aspx.

50. Skybox, *Security Risk Management (SRM) with Skybox view*. 2004, Skybox Security Inc. p. 6.

51. Skybox, *Skybox Secure Data Sheet*, Skybox, Editor. 2007, Skybox: San Jose, California. p. 4.

52. Skybox. *Skybox Security Secure - Automated Risk Lifecycle Management*. 2007, Accessed 1-12-2007, http://www.skyboxsecurity.com/products/secure.html.

53. Swiler, L.P., et al. *Computer-attack graph generation tool*. in *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01. Proceedings*. 2001.

54. Tenable. *Tenable Network Security - Nessus 3.0*. 2007, Accessed 10-11-2007, http://www.nessus.org/nessus/.

55. Tenable. *Tenable Network Security - Security Center 3.0*. 2007, Accessed 11-11-2007, http://www.nessus.org/products/sc/.

56. Tenbable, *Security Event Management*. 2007, Tenable Network Security: Columbia. p. 10.

57. Threatguard. *Secutor Prime Products Page*. 2007, Accessed 28-11-2007, http://www.threatguard.com/secutor-prime.htm.

58. Threatguard. *Threatguard Products*. 2007, Accessed 28-11-2007, http://www.threatguard.com/products.htm.

59. Threatguard. *Threatguard Products - Vulnerability management*. 2007, Accessed 28-11-2007, http://www.threatguard.com/products.htm#Vulnerability_Management.

60. Weiss, J.D. *A System Security Engineering Process*. in *14th National Computer Security Conference*. 1991.

61. Wilander, J. and M. Kamkar, *A Comparison of Publicly Available Tools for Dynamic Buffer Overflow Prevention*, in *10th Network and Distributed System Security Symposium (NDSS)*. 2003.

62. Xinming, O., F.B. Wayne, and A.M. Miles, *A scalable approach to attack graph generation*, in *Proceedings of the 13th ACM conference on Computer and communications security*. 2006, ACM: Alexandria, Virginia, USA.