A quantitative evaluation of vulnerability scanning

Hannes Holm[1], Teodor Sommestad[1], Jonas Almroth[2], Mats Persson[2]

[1] Royal Institute of Technology, Industrial information and control systems Osquldas väg 10, 7 tr, 100 44 Stockholm, Sweden

[2] Swedish Defence Research Agency
Box 1165, SE-581 11 Linköping, Sweden

Corresponding author:
Hannes Holm
Email: hannesh@ics.kth.se
Telephone: +46-8-7906826
Fax: +46-8-7906839

Purpose: Evaluate if automated vulnerability scanning accurately identifies vulnerabilities in computer networks and if this accuracy is contingent on the platforms used.

Design/methodology/approach: Both qualitative comparisons of functionality and quantitative comparisons of false positives and false negatives are made for seven different scanners. The quantitative assessment includes data from both authenticated and unauthenticated scans. Experiments were conducted on a computer network of 28 hosts with various operating systems, services and vulnerabilities. This network was set up by a team of security researchers and professionals.

Findings: The data collected in this study show that authenticated vulnerability scanning is usable. However, automated scanning is not able to accurately identify all vulnerabilities present in computer networks. Also, scans of hosts running Windows are more accurate than scans of hosts running Linux.

Research limitations/implications: This paper focuses on the direct output of automated scans with respect to the vulnerabilities they identify. Areas such as how to interpret the results assessed by each scanner (e.g. regarding remediation guidelines) or aggregating information about individual vulnerabilities into risk measures are out of scope.

Practical implications: This paper describes how well automated vulnerability scanners perform when it comes to identifying security issues in a network. The findings suggest that a vulnerability scanner is a useable tool to have in your security toolbox given that user credentials are available for the hosts in your network. Manual effort is however needed to complement automated scanning in order to get satisfactory accuracy regarding network security problems.

Originality/value: Previous studies have focused on the qualitative aspects on vulnerability assessment. This study presents a quantitative evaluation of seven of the most popular vulnerability scanners available on the market.

1 Introduction

Software vulnerabilities are found in commonly used software products every day. More than 45 000 software vulnerabilities have been published by the National Vulnerability Database and over 4500 of these software vulnerabilities were published during 2010 alone (NVD 2011). Efficient management of vulnerabilities is thus an important activity in modern enterprises security efforts. To manually keep track of all vulnerabilities present in systems and remediate them appropriately is a daunting task (see e.g. Sommestad, Ekstedt, Holm, & Afzal, 2011). Fortunately, there are tools aimed to provide automated support for this process available (Al-Ayed et al. 2005).

One commonly applied solution involves the use of network vulnerability scanners (Werlinger et al. 2010). A network vulnerability scanner is an appliance or software which is used to scan the architecture of a network and report any identified vulnerabilities. The normal procedure of scanning a network with a vulnerability assessment tool generally involves three parts: network scanning, vulnerability scanning and vulnerability analysis (Manzuik et al. 2007).

Network scanning involves identifying which hosts that are alive in the computer network, which operating systems that they use, and what services they run. During the *vulnerability scan* a database of vulnerability signatures is compared to the information obtained from a network scan to produce a list of vulnerabilities that are presumably present in the network. Most tools thereafter attempt to verify the presence of these vulnerabilities through signatures – carefully constructed queries which aim to verify the vulnerability's presence without disrupting the service. Some tools also provide the possibility of actually exploiting vulnerabilities to fully verify their presence. *Vulnerability analysis* concern evaluating the severity of identified vulnerabilities. Organizations typically have a large number of vulnerabilities in their operational environment and some vulnerabilities lead to a higher security risk than others. For instance, some software vulnerabilities are easy to utilize, and some have dire consequences if they are exploited. It is thus important to assess the most significant problems and remediate these first. Powerful vulnerability analysis would certainly provide great value to the vulnerability management process. However, most vulnerability management tools are fairly immature in this respect. In academia there are several projects which have tried to solve this problem. A few examples include MulVAL (Ou et al. 2005; Homer & Ou 2009), NetSPA (Ingols et al. 2006; Chu et al. 2010), TVA-Tool (Noel et al. 2009; Jajodia & Noel 2005), incident response intelligence systems (Patsos et al. 2010) and work by Sommestad et al. (T. Sommestad et al. 2010). All these analysis tools require detailed data about the vulnerabilities present in a network.

The quality of the information produced by network and vulnerability scanning is thus of importance, regardless if an evaluation tool is used for the analysis or if it is done manually. Unfortunately, these scans are not always correct. Network and vulnerability scanning base their assessments on signatures of operating systems used, services running, and their corresponding vulnerabilities. These signatures do not always provide the correct result, which causes issues for the security management staff. Sometimes these issues result in failure to identify existing vulnerabilities (i.e. false negatives); sometimes they result in erroneously reporting inexistent vulnerabilities to be present (i.e. false positives). If scans produce vulnerability lists containing these errors it will impede efficient mitigation – false positives will result in efforts to manage nonexistent problems and false negatives may lead to unexpected security problems.

While vulnerability scanners depend on their accuracy there have been no thorough evaluations to assess their performance. Nor has it been tested if there is a difference in accuracy between scans of different operating systems. This paper compares seven popularly used vulnerability scanners and attempts to answer two research questions:

> RQ1: *Is automated vulnerability scanning able to accurately model vulnerabilities of network architectures?*
>
> RQ2: *Is there a difference in accuracy regarding vulnerability assessments in Linux and Windows?*

In order to fully interpret results of RQ1 there is a need to survey the nature of the functionality provided by vulnerability scanners. This paper thus also provides a qualitative study regarding this subject.

The rest of the paper is structured as follows. Section 2 gives an overview of related works. Section 3 presents the properties that were evaluated and Section 4 describes the evaluation method. Section 5 presents the results of the study. In Section 6 and Section 7 results are discussed and conclusions are drawn.

2 Related works

No previous academic work has been found that quantitatively evaluates the accuracy of vulnerability scanners. There are few quantitative studies by industry actors available and the ones that are available generally lack description of method and do not evaluate false positives. Several qualitative evaluations however exist, mainly in popular press and reports created by technology analysts (Skousen 2009; Brackin 2002; Doupé et al. 2010; Fong et al. 2008; Fong & Okun 2007; Manzuik et al. 2007; Stephenson 2010; C. Wang et al. 2010; Kavanagh et al. 2010; Welberg 2008). The most relevant evaluations are described in this chapter together with the properties investigated.

2.1 Qualitative studies

Welberg (Welberg 2008) qualitatively describes a very large variety of security evaluation tools, including vulnerability scanners. The author uses a taxonomy consisting of two main criteria: scope and analysis. Scope describes what data the tools can assess and how it conforms to different standards. The criterion is built on the sub criteria i) Software Platforms (e.g. Windows, UNIX), ii) Magnitude (number of hosts), Standards (e.g. if CCE and CVE standards are used) and iii) Type of vulnerabilities (configuration/source code/environment). The Analysis criterion is defined by i) Kind of analysis (e.g. compliance or vulnerability scanning), ii) Type of results (i.e. how it is presented) and iii) Information available (i.e. if there was enough information available for comparison). (Welberg 2008) compares the security evaluation tools with dichotomies, e.g. either the tool is able to scan Windows machines or it is not. A few tools are selected for more in-depth discussion, but this discussion is still on a very superficial level and no analysis is made regarding tool accuracy.

The Information Assurance Technology Analysis Center has conducted a comparison of a large set of security management tools, including various vulnerability scanners (Skousen 2009). The framework for comparison includes product features and a table of general information (e.g. on which operating systems and hardware that the tool can run or what type of tool it is). A problem with this report is that it is very difficult to compare products on more than a general level since each set of features is unique. Also, it does not provide any quantitative data on their accuracy.

Wang et al. (C. Wang et al. 2010) evaluates vulnerability assessment tools according to three main dimensions: i) *Current offering*, ii) *Strategy* and iii) *Market presence*. *Current offering* involves analyzing the tool capability on vulnerability assessment, both at the network/system level and at the application level; configuration compliance assessment; and any remediation capabilities (or support for remediation). It also includes features such as reporting, performance, mode of delivery, and support for risk management. *Strategy* includes an assessment of the high-level company strategy, near-term product road map, and the company's plan for a partner ecosystem. In terms of company strategy, the authors looked at the vendor's vision and its value proposition, how well it is executing this vision and delivering on the value proposition, and whether the strategy demonstrates industry thought leadership. *Market presence* was evaluated using traditional metrics, such as vendor revenues and number of customers' numbers. The report produced by Wang et al. (C. Wang et al. 2010) focuses more on assessing the market for vulnerability scanners than their capabilities. It is also very difficult to understand exactly what and how the authors measure, resulting in non-transparent results of little use. This area is also the focus of a report produced by Gartner (Kavanagh et al. 2010). Also Gartner's study is non-transparent with respect to method; it seems to be based primarily on the authors' (analysts') own judgment and completely lack any quantitative data.

2.2 Quantitative studies

Andress (Andress 2004) quantitatively compares eight vulnerability scanners. The author grades the detection rate of scanners according to a scale from one to five and concludes that vendors need to continue improving their scanning engines. However, the author does not describe methodology nor assess false positives. Forristal and Shipley (Forristal & Shipley 2001) compare the detection rate of seven scanners regarding 17 vulnerabilities, but do not look at false positives. Some aspects in the "Current offering" dimension in the report by Wang et al. (C. Wang et al. 2010) are quantitative and related to scanning accuracy. However, nor method or false positives are addressed.

3 A framework for comparison

This paper looks at the domain of vulnerability scanning partially as the *scope* criteria in Welberg's (Welberg 2008) paper and partially as the vulnerability assessment attributes of the *current offering* criterion by Wang et al. (C. Wang et al. 2010). In addition to comparing functionality this paper also addresses how accurate the scanners' vulnerability reports are. The properties evaluated are categorized under the criteria *Functionality*, i.e. what a scanner can do, and *Accuracy*, i.e. how accurately it identifies vulnerabilities.

3.1 Functionality

This criterion is defined through a number of properties described in Table 1. Most of these are self-explanatory; those that might need a more elaborate definition are described below.

Passive scanning involves if the tool can infer vulnerability information by examining existing network traffic and thereby avoid the potential interference caused by queries generated by the scanner. If the scanner can execute *exploit*s it can validate assessed vulnerabilities by actually exploiting them.

In *authenticated scanning* the scanner is given authentication parameters (i.e. credentials) of hosts to enable more detailed and presumably also more accurate scans. Authenticated scans are typically less intrusive and provide a higher degree of accuracy. However, it is not always the case that all credentials are readily available for the individual(s) performing a scan. Another property evaluated is the number of *vulnerability signatures* the tool has in its database. However, this number is not a very good indicator of capability as some tools have one signature for several vulnerabilities and other tools have unique signatures for all vulnerabilities. An important factor is if the tool complies with the Security Content Automation Protocol, *SCAP* (Quinn et al. 2010). SCAP is a suite of six commonly used protocols developed by the National Institute of Standards and Technology (NIST): i) Extensible Configuration Checklist Description Format (XCCDF), ii) Open Vulnerability and Assessment Language (OVAL), iii) Common Platform Enumeration (CPE), iv) Common Configuration Enumeration (CCE), v) Common Vulnerabilities and Exposures (CVE) and vi) Common Vulnerability Scoring System (CVSS). SCAP also include a database of security flaws and configuration error reference data provided by the National Vulnerability Database (NVD). A tool which supports the SCAP standard thus at minimum cover (and make references to) the type of data included in the six abovementioned standards.

Table 1. Properties related to Functionality.

Property	Description	Range
Software flaws	If the tool can identify software flaws.	{True, False}
Configuration errors	If the tool can identify configuration errors.	{True, False}
All ports	If all UDP and TCP ports can be probed by the tool.	{True, False}
Active scanning	If the tool can query the network for vulnerabilities.	{True, False}
Passive scanning	If the tool can perform passive scans, i.e. network sniffing.	{True, False}
Exploits	If the tool can validate the assessed vulnerabilities with exploits.	{True, False}
Authenticated scanning	If the tool can perform credentialed scanning of both Linux-based and Windows-based systems.	{True, False}
Vulnerability signatures	How many vulnerability signatures the tool has in its database.	Natural numbers
Web application scans	If the tool has built-in web crawler functionality.	{True, False}
Applications assessment	If the tool guesses the applications (and their versions) running on ports, i.e. not just packet banner information and vulnerability data.	{True, False}
Patch deployment	If the tool can perform patch deployment.	{True, False}
SCAP compliance	If the tool is SCAP-approved by NIST. A SCAP-approved tool supports the standards XCCDF, OVAL, CPE, CCE, CVE and CVSS.	{True, False}

3.2 Accuracy

Detection rate is related to false negatives: how many existing vulnerabilities that the scanner manages to identify. False alarms are related to false positives: how many non-existing vulnerabilities that the scanner falsely identifies as actual vulnerabilities.

Table 2. Properties related to data quality.

Property	Description	Range
Detection rate	How many existing vulnerabilities the scanner manages to identify.	0-100%
False alarms	How many non-existing vulnerabilities the scanner identifies as actual vulnerabilities (i.e. false alarms).	0-100%

4 Method

This chapter describes the data collected during the study.

4.1 Vulnerability scanners included

There are various vulnerability scanners available today. The more renowned ones include AVDS (Beyond Security 2011), Core Impact (Core Security 2011), FusionVM (Critical Watch 2011), Patchlink Scan (Lumension 2011), nCircle (nCircle 2011), Nessus (Tenable 2011), NeXpose (Rapid7 2011), ProtectPoint (Still Secure 2011), QualysGuard Enterprise (Qualys 2011), Retina Network Security Scanner (eEye 2011), SAINT (Saint corporation 2011), SecureFusion Portal (Symantec 2011), Trustwave (Trustwave 2011), McAfee Vulnerability manager (McAfee 2011) and GFI Languard (GFI 2011). All the above mentioned vendors were asked to be a part of the study and seven agreed to participate with fully functional products. The authors had no previous relations to any of the vendors and believe that the evaluated tools provide a representative sample of the domain. The evaluated tools are: AVDS, Patchlink scan, Nessus, NeXpose, QualysGuard, SAINT and McAfee VM. All scanners were used with latest version available as of January 2011.

4.2 Experimental setup

The experimental setup was designed by the Swedish Defense Research Agency in Linköping, Sweden, with the support of the Swedish National Defense College. Additionally, a group of computer security specialists and computer security researchers from various northern-European governments, military, private sectors and academic institutions were active in designing it.

The environment was set to describe a typical critical information infrastructure at a small electrical power utility. The environment was composed of 20 physical servers running a total of 28 virtual machines, divided into four VLAN segments. Various operating systems and versions thereof were used in the network, e.g. Windows XP SP2, Debian 5.0 and Windows Server 2003 SP1. Each host had several different network services operating, e.g. HTTP, HTTPS, SMTP, FTP, Streaming Media Server, RDP, SSH, SMB and VNC. Furthermore, every host was more or less vulnerable through software flaws and/or poor configurations.

4.3 Measurement method

Qualitative data was collected for the criterion *Functionality* through available documentation of the products, correspondence with product vendors and manual inspections.

Quantitative data was collected for the *Accuracy* criterion through experimental evaluations of the performance during both unauthenticated and authenticated scans on the described network architecture. The evaluated vulnerability scanners were provided by vendors either as stand-alone installers or as complete appliances. Scanners provided as stand-alone installers were run on dedicated virtual machines with 100GB disk space, 34GB random access memory, two quad-core processors and installed on operating systems (OS) and with settings as recommended by each vendor. All guest OS were run on VMware Vsphere ESX 4.0 with only the performing scanner active and network adapters set to function in promiscuous mode. To increase performance no services expect those needed by the tools were run.

The network architecture was identical for all scans and also completely untouched during the time of the scans. Firewalls separating subnets were set to allow all traffic. The scanners were configured to scan for services on all TCP-ports and all vulnerability signatures enabled. Furthermore, only safe checks were utilized (i.e. test modes that could create problems for services were not used). This setting was chosen due to the fact that most organizations cannot afford to have crashed services due to a security audit, especially in the critical infrastructure domain.

Some scanners inform about potential vulnerabilities (i.e. vulnerabilities they could not confirm) and verified vulnerabilities (vulnerabilities that could be confirmed). To make the scanners output comparable this study makes no distinction between these types and all indications of vulnerabilities are treated in the same way – as indications of vulnerabilities being present.

False negatives were assessed by comparing a list of 50 existing vulnerabilities with each scanners assessment. These 50 vulnerabilities were selected through independent random sampling from the entire pool of existing vulnerabilities. Only vulnerabilities of High severity according to the Common Vulnerability Scoring System (CVSS) (Mell et al. 2007) were assessed. Focus was placed on High severity vulnerabilities since vulnerabilities of Low and Medium severity rarely can be exploited to gain administrator privileges of machines, and therefore are of less significance to administrators.

False positives were evaluated through examining 40 vulnerabilities assessed by each scanner during the unauthenticated scans. These vulnerabilities were also chosen through independent random sampling. A few of the evaluated vulnerabilities were rated Medium by CVSS rather than high. These Medium severity vulnerabilities were included when less than 40 high-severity vulnerabilities was listed in the scanners output. Finally, any false positives from the unauthenticated scans were compared to the output of each corresponding authenticated scan to get a measurement of the rate of false positives for authenticated scans.

5 Results

This chapter is divided in three sections. The first section describes the tools on a more qualitative level while the second section contains the quantitative results from the actual tests (*RQ1*). The third section provides the evaluation of accuracy in scanning Windows and Linux hosts (*RQ2*).

5.1 Functionality

The results from the qualitative evaluations can be seen in Table 3. The vendors provide fairly similar tools in terms of base functionality.

5.2 Accuracy

The overall results from the different scanners can be seen in Table 4. As can be seen in this table, all scanners found a different number of vulnerabilities. The differences are in some cases astonishing. The authenticated scans greater capability of assessing vulnerabilities is clearly displayed.

Since the false positives and false negatives are described through binary variables (i.e. either a vulnerability is identified/correctly negated, or it is not) it is possible to assume a binomial distribution. That is, the vulnerabilities and alarms examined in this study come from an independent sequence of yes/no answers corresponding to correct detection or false alarms.

The properties of the binomial distribution made it possible to approximate normal distributions and perform one-way analysis of variance (ANOVA) (Warner 2008). ANOVA is a technique for assessing if there are statistical differences between groups of data. When a p-value is mentioned in this chapter it refers to results from ANOVA. The p-value is used to explain the statistical difference between two or more sources of data, and a p-value of less than *0.05* is a commonly used reference value for claiming that there is a significant difference, e.g. regarding the output of seven scanners. A p-value of less than 0.05 implies that there is less than 5% probability that the assessed differences between two or more sources of data are due to random variation.

Table 3. Qualitative data.

Property	AVDS	McAfee VM	Nessus	NeXpose	Patchlink scan	QualysGuard	SAINT
Software flaws	x	x	x	x	x	x	x
Configuration errors	x	x	x	x	x	x	x
All ports	[4]	x	x	x	[4]	x	x
Active scanning	x	x	x	x	x	x	x
Passive scanning			[2]				
Exploits				x			x
Authenticated scanning	[3]	x	x	x	x	x	x
Vulnerability signatures	6000	22000	41000	53000	500	6000	40000
Web application scans	x	x	x	x		x	
Applications assessment				x			
Patch deployment					x		
SCAP compliance	[1]	x	x	x	x	x	x

[1] While AVDS is not in the list of SCAP validated tools defined by NIST it seems to be as compliant with SCAP as the other tools.

[2] While Nessus itself cannot carry out passive scanning, another tool by Tenable can.

[3] Only for Windows hosts.

[4] Only for TCP, not for UDP.

Table 4. Overview of identified vulnerabilities.

Scanner	Unauthenticated scan				Authenticated scan			
	High	Medium	Low	Potential	High	Medium	Low	Potential
AVDS	46	140	306		291	990	393	
McAfee VM	143	169	64		2028	2033	126	
Nessus	145	82	889		2221	468	1256	
NeXpose	180	391	106		1073	969	126	
Patchlink scan	1	4	15		814	328	313	
QualysGuard	73	125	151	284	753	1891	342	313
SAINT	81	60		223	114	65		257

5.2.1 Unauthenticated scans

Results from the study regarding false negatives for unauthenticated scans can be seen in Table 5.

There were significant differences between how many issues the scanners managed to detect, from 0% to 36% of the existing vulnerabilities. An ANOVA of the overall detection rate shows that there is a statistical difference between the tools ($p = 3.5 \ast 10^{-6}$).

Table 6 describes the false alarm rate of the tools. The frequency of false alarms was in general fairly low, indicating that the tools often fail to assess actual vulnerabilities but are reliable when they do.

There was not any statistical difference between the results (p = 0.085). However, the rather low p-value suggests that future studies might show different results.

Table 5. Detection rate for unauthenticated scans with corresponding 95% lower confidence intervals (CIL) and 95% upper confidence intervals (CIU).

Tool	Detection %	Stdev	CIL (95%)	CIU (95%)	Detection % (Linux)	Detection % (Win)	n
AVDS	6	24	0	12	0	11	50
McAfee VM	8	27	2	14	0	15	50
Nessus	20	40	11	29	4	33	50
NeXpose	24	43	14	34	22	30	50
Patchlink scan	0	0	0	0	0	0	50
QualysGuard	24	43	14	34	17	30	50
SAINT	36	48	25	47	43	30	50

Table 6. False alarms for unauthenticated scans with corresponding 95% lower confidence intervals (CIL) and 95% upper confidence intervals (CIU).

Tool	False alarm (%)	Stdev	CIL (95%)	CIU (95%)	False alarm % (Linux)	False alarm % (Win)	n
AVDS	0	0	0	0	0	0	40
McAfee VM	3	16	0	7	0	3	40
Nessus	5	22	0	12	18	3	40
NeXpose	5	22	0	12	11	0	40
Patchlink scan	6	24	0	17	0	8	17
QualysGuard	13	33	2	23	15	11	40
SAINT	15	36	4	26	11	18	40

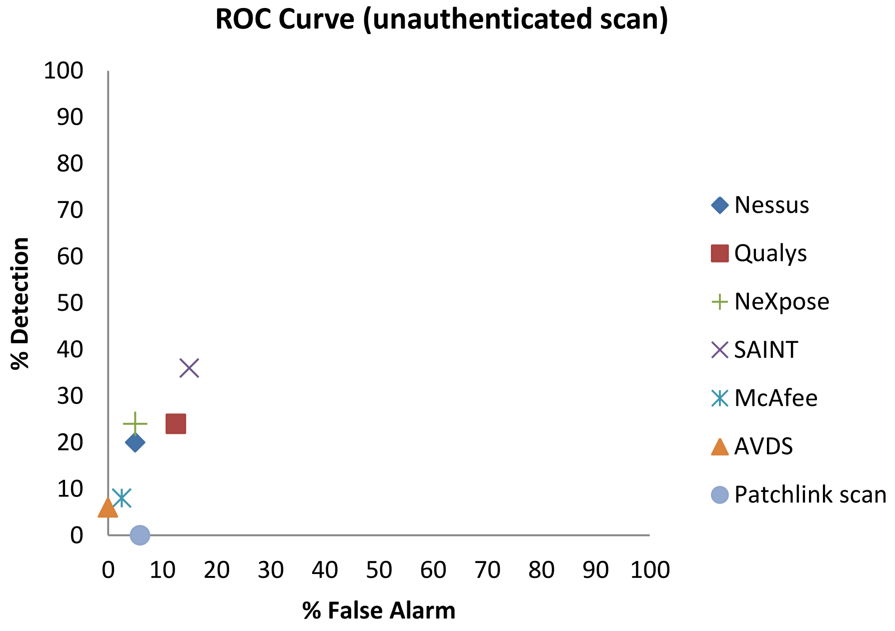

Figure 1. ROC curve for the unauthenticated scans.

Figure 1 display a Receiver Operating Characteristic (ROC) curve (see. e.g. Lazarevic, Ertoz, Kumar, Ozgur, & Srivastava, 2003) of the unauthenticated checks. Informally speaking, it seems that there is a strong connection between the detection rate and rate of false alarms. This suggests that a higher rate of false positives can be traded for increased detection rate (and vice versa) to suit different user needs.

5.2.2 Authenticated scans

As can be seen in Table 7, all confidence intervals regarding the authenticated scans and detection rate fully overlap. Thus, there is no reason to believe that the scanners perform statistically different when it comes to finding vulnerabilities using credential scans. An analysis using ANOVA confirms this assumption; there is no statistical basis for saying that one tool performs better than the other (p = 0.43).

Table 7. Detection rate for authenticated scans with corresponding 95% lower confidence intervals (CIL) and 95% upper confidence intervals (CIU).

Tool	Detection %	Stdev	CIL (95%)	CIU (95%)	Detection % (Linux)	Detection % (Win)	n
AVDS	34	48	23	46	0	67	50
McAfee	36	49	23	46	0	70	50
Nessus	43	50	31	55	9	75	50
NeXpose	43	50	31	55	22	63	50
Patchlink scan	36	49	25	48	0	71	50
QualysGuard	55	50	43	67	17	92	50
SAINT	43	50	31	55	57	29	50

Table 8 provides the results from the authenticated scans regarding false positives. Two of the false alarms during the unauthenticated scans were not mentioned this time (Qualys and SAINT). There is no statistical difference between the scanners (p = 0.25).

Table 8. False alarms for authenticated scans with corresponding 95% lower confidence intervals (CIL) and 95% upper confidence intervals (CIU).

Tool	False alarm (%)	Stdev	CIL (95%)	CIU (95%)	False alarm % (Linux)	False alarm % (Win)	n
AVDS	0	0	0	0	0	0	40
McAfee	3	16	0	7	0	3	40
Nessus	5	22	0	12	18	3	40
NeXpose	5	22	0	12	11	0	40
Patchlink scan	6	24	0	17	0	8	17
QualysGuard	10	30	1	19	15	7	40
SAINT	13	33	2	23	6	18	40

The ROC curve for the authenticated scan (Figure 2) is very similar to the unauthenticated ROC curve in the sense that it displays a relationship between detection rate and false alarms.

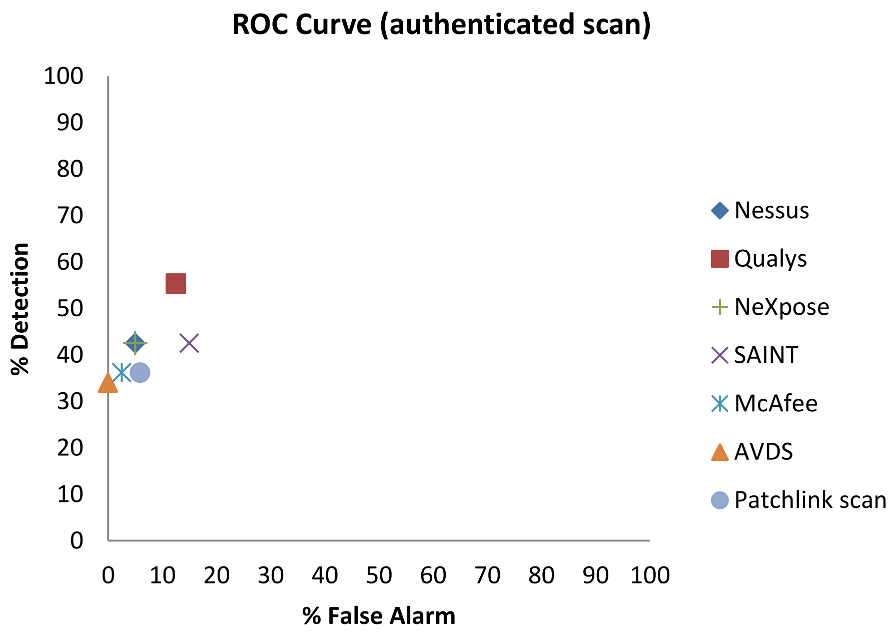

Figure 2. ROC curve for the authenticated scans.

5.2.3 Vulnerability assessment of Windows and Linux hosts

This section provides data regarding how the scanners performed on the assessed Windows and Linux machines. The detection rate when doing both unauthenticated and authenticated scans are significantly higher on Windows machines ($p = 0.041$ for unauthenticated scans and $p = 8.4 * 10^{-22}$ for authenticated scans, Table 9).

Table 9. Detection rate for authenticated scans with corresponding 95% lower confidence intervals (CIL) and 95% upper confidence intervals (CIU).

Scan type	OS	Detection %	Stdev	CIL (95%)	CIU (95%)	n
Authenticated	Linux	15	36	9	20	161
	Windows	62	49	56	69	189
Unauthenticated	Linux	12	33	7	18	161
	Windows	21	41	15	26	189

There were approximately twice as many false alarms for the Linux hosts than for the Windows hosts, both for authenticated and unauthenticated scans (Table 10). The ANOVA does however not show any statistical difference ($p = 0.15$ for the authenticated scans and $p = 0.099$ for the unauthenticated scans), but the fairly low p-values suggest that further studies could prove otherwise.

Table 10. False alarm rate for with corresponding 95% lower confidence intervals (CIL) and 95% upper confidence intervals (CIU).

Scan type	OS	False alarm %	Stdev	CIL (95%)	CIU (95%)	n
Authenticated	Linux	11	32	3	19	63
	Windows	5	22	2	8	194
Unauthenticated	Linux	10	30	2	17	63
	Windows	5	21	2	8	194

6 Discussion

This chapter is structured in three parts. The two first parts concern the two research questions; the last concerns the validity and reliability of the study.

6.1 RQ1: Accuracy

In general it seems that automated vulnerability scanners are more accurate in terms of false alarms than detection rate. That is, assessments using vulnerability scanners identifies approximately half of all vulnerabilities present in a network, in general with only a few false positives. This suggests that scanning is useful and that it can aid network administrators in decreasing the attack surfaces of their networks without having to spend much effort handling false alarms. Both false positives and false negatives come with potential costs and the tools seem to strive for an optimal tradeoff between the two with respect to their users' general needs. However, in practice some users might favor few false alarms in front of high detection and some might work according to the precautionary principle (Stewart 2004), and thus favor detection rate in front of false alarms (e.g. because of extreme security requirements). One way to satisfy all potential users' needs could be to assign a "probability of existence" value to each assessed vulnerability. Such a value would make it possible to customize the output from a scan according to different users' particular needs. Two of the tested scanners (Qualys and SAINT) state if a vulnerability is "potential". However, what "potential" means in terms of a probability is not apparent.

Our results show that vulnerability scanners only find a subset of the vulnerabilities present in computer networks. Their accuracy can be improved by giving them credentials to the scanned hosts, but the information they offer is still far from perfect. However, their inaccuracy during unauthenticated scanning is not necessarily a bad thing. If the unauthenticated scan is the modus operandi for the attacker and the authenticated scan is the modus operandi for the system administrator this offers a significant advantage to those protecting computer networks. That is, providing a limited picture of potential security holes to an unauthenticated user could lessen the risk of successful compromise more than it impedes network administrators from securing their systems. This line of argumentation can find support in Furnell et al. (Furnell et al. 2001). However, even if such an argumentation holds there is still need for ample of improvement regarding accuracy of the authenticated scans. Certainly, few network administrators would feel contempt with identifying somewhere around 30-50 percent of the vulnerabilities in their network.

A combined scan using all the included tools yields a mean of 80% detection rate for credentialed scans (Table 11). This suggests that a joint scan using several appliances can be a potent solution when in need of highly accurate scans (preferably by exporting scanned results to a unified database in order to eliminate redundant alarms). It should also be noted that the rate of false alarms is kept low also for combined scans (7% for the unauthenticated scans and 6% for the authenticated scans).

Table 11. Detection rate for a combination of all scanners with corresponding 95% lower confidence intervals (CIL) and 95% upper confidence intervals (CIU).

Scan type	Detection %	Stdev	CIL (95%)	CIU (95%)	Detection % (Linux)	Detection % (Win)
Authenticated	80	41	69	89	65	92
Unauthenticated	44	50	32	56	52	37

It also seems that the scanners are significantly better at assessing common products (e.g. Apache webserver) than the bit more uncommon ones. A few examples in this study include the Icecast media server which no scanner properly identified during the unauthenticated scan and the Programmable Logic Controller, a type of network component used for automation of electromechanical processes, which every scanner identified as a different product (and none as the correct product). This problem is logical considering the circumstances; it would take too much of an effort to have unique signatures for all products available on the market.

6.2 RQ2: Linux and Windows

There were significant differences in detection rate, both for authenticated and unauthenticated scans, between Linux and Windows OS. There are also notable differences in false positives, although not statistically significant. One reason for these differences might be that patches of Windows hosts result in more significant changes in code, while patches applied to many Linux distributions result in only minor changes. For example, a Debian patch might change code but keep the version number, thus making automated scanning significantly more difficult. Another possible reason could be that the vulnerability scanner vendors see a greater potential in the market of Windows hosts and thus have focused their development efforts in that domain.

6.3 Validity and reliability

The network architecture was a virtual environment. Running things in a virtual environment can decrease performance, among other things packet loss. This problem is however mainly evident in very large virtual environments and not small subnets as evaluated in this study, and should thus be a minor issue (Ye et al. 2010; McDougall & Anderson 2010; G. Wang & Ng 2010).

Two appliances were not virtualized. Indeed, an appliance running on hardware specifically designed by the vendor is likely to have an optimized environment. All appliances run in virtualized environments were however set up with environments and parameters according to recommendations by each vendor. This should next to eliminate any problems regarding appliance setups.

This study can only evaluate the vulnerability scanners ability to assess the vulnerabilities in the specified network architecture. This study only covers a very small amount of the vulnerabilities which are currently available (there are e.g. more than 45 000 listed vulnerabilities in the NVD). The operating systems, services and vulnerabilities implanted in the network architecture were however of diverse nature. Thus, we believe that this study gives a good hint towards the accuracy of the evaluated tools.

7 Conclusions and future work

This paper presents an evaluation of seven popular vulnerability scanners and found interesting results, e.g. significant differences between scans of Linux and Windows hosts and accuracy of the scanners during both authenticated and unauthenticated scans. Results show that automated scanning, while useful, only find a subset of the vulnerabilities present in a network. The output from a vulnerability scan should therefore be interpreted with care and, if possible, be complemented by other efforts.

Several important attributes were not evaluated in this study and should be further studied. First, it would be interesting to study the reasons behind the lack of accuracy in more detail. Second, the aim of all vulnerability scanners is to provide the decision maker with information in such a way that vulnerabilities can be remediated and managed efficiently. Hence, it is important that the scanner is seen as both useful and easy to use. This can be evaluated through e.g. the Technology Acceptance Model (Venkatesh et al. 2003) or experiments involving decision makers. Third, a variable that sometimes is of great importance is if a scanner affects the services running on the network. A common problem, especially for unauthenticated scans, is that they unintentionally perform successful denial-of-service attacks against services. Passive vulnerability scanning is a very interesting topic which should be studied further. Passive vulnerability scanning is naturally especially important for organizations working with critical infrastructure. Passive and active scans' intrusiveness on normal operation could be evaluated. Four, many tools have special analysing and reporting features available, e.g. analysis of internal or external vulnerability trends. These might reduce the need for manual analysis or increase the value of results obtained from a scan. Features such as these are naturally also important to look at and should be further studied.

8 References

Al-Ayed, A. et al., 2005. An automated framework for managing security vulnerabilities. *Information Management & Computer Security*, 13(2), pp.156-166.

Andress, M., 2004. Network vulnerability assessment management. *Network World*.

Beyond Security, 2011. AVDS. Available at: http://www.beyondsecurity.com/ [Accessed February 13, 2011].

Brackin, C., 2002. *Vulnerability Management: Tools, Challenges and Best Practices*.

Chu, M. et al., 2010. Visualizing attack graphs, reachability, and trust relationships with NAVIGATOR. In *Proceedings of the Seventh International Symposium on Visualization for Cyber Security*. ACM, p. 22–33.

Core Security, 2011. Core Impact. Available at: http://www.coresecurity.com [Accessed February 13, 2011].

Critical Watch, 2011. FusionVM. Available at: http://www.criticalwatch.com/products/vulnerability-management-overview.aspx.

Doupé, A., Cova, M. & Vigna, G., 2010. Why Johnny Can't Pentest: An Analysis of Black-box Web Vulnerability Scanners. *Detection of Intrusions and Malware, and Vulnerability Assessment*, p.111–131.

eEye, 2011. Retina Network Security Scanner. Available at: http://www.eeye.com/html/products/Retina/ [Accessed February 13, 2011].

Fong, E. et al., 2008. Building a Test Suite for Web Application Scanners. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*. IEEE, p. 478.

Fong, E. & Okun, V., 2007. Web Application Scanners: Definitions and Functions. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*. IEEE, p. 280b.

Forristal, J. & Shipley, G., 2001. Vulnerability Assessment Scanners. *Network Computing*.

Furnell, Steven M., Chiliarchaki, P. & Dowland, Paul S., 2001. Security analysers: administrator assistants or hacker helpers? *Information Management & Computer Security*, 9(2), pp.93-101.

GFI, 2011. GFI LANguard. Available at: http://www.gfi.se/lannetscan/ [Accessed February 13, 2011].

Higgins, H.N., 1999. Corporate system security: towards an integrated management approach. *Information Management & Computer Security*, 7(5), pp.217 - 222.

Homer, J. & Ou, X., 2009. SAT-solving approaches to context-aware enterprise network security management. *IEEE JSAC Special Issue on Network Infrastructure Configuration*.

Ingols, K., Lippmann, R. & Piwowarski, K., 2006. Practical attack graph generation for network defense. *Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual*.

Jajodia, S. & Noel, S., BO'Berry, 2005. Topological analysis of network attack vulnerability. *Managing Cyber Threats*, pp.247-266.

Kavanagh, K.M., Nicolett, M. & Pescatore, J., 2010. *MarketScope for Vulnerability Assessment*.

Lazarevic, A. et al., 2003. A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the Third SIAM International Conference on Data Mining*. p. 25–36.

Lumension, 2011. Patchlink Scan. Available at: http://www.lumension.com/vulnerability-management.jsp [Accessed February 13, 2011].

Manzuik, S. et al., 2007. *Network security assessment: from vulnerability to patch*, Syngress.

McAfee, 2011. McAfee Vulnerability Manager. Available at: http://www.mcafee.com/us/products/vulnerability-manager.aspx [Accessed February 13, 2011].

McDougall, R. & Anderson, J., 2010. Virtualization performance: perspectives and challenges ahead. *ACM SIGOPS Operating Systems Review*, 44(4), p.40–56.

Mell, P., Scarfone, K. & Romanosky, S., 2007. A Complete Guide to the Common Vulnerability Scoring System (CVSS), Version 2.0, Forum of Incident Response and Security Teams.

nCircle, 2011. nCircle IP360. Available at: http://www.ncircle.com/index.php?s=products_ccm.

Noel, S. et al., 2009. Advances in Topological Vulnerability Analysis. In *Conference For Homeland Security, 2009. CATCH'09. Cybersecurity Applications & Technology*. IEEE, p. 124–129.

NVD, 2011. National Vulnerability Database. Available at: http://nvd.nist.gov/ [Accessed February 13, 2011].

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

Patsos, D., Mitropoulos, S. & Douligeris, C., 2010. Expanding Topological Vulnerability Analysis to Intrusion Detection through the Incident Response Intelligence System. *Information Management & Computer Security*, 18(4).

Qualys, 2011. QualysGuard Enterprise. Available at: http://www.qualys.com/products/risk_compliance/qgent/.

Quinn, S. et al., 2010. *Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1. 0*.

Rapid7, 2011. NeXpose. Available at: http://www.rapid7.com/nexpose/uvm.jsp [Accessed February 13, 2011].

Saint corporation, 2011. SAINT. Available at: http://www.saintcorporation.com/products/software/saintScanner.html [Accessed February 13, 2011].

Skousen, R.A., 2009. *Information Assurance Report - Vulnerability Assessment*.

Sommestad, T., Ekstedt, M. & Johnson, P., 2010. A probabilistic relational model for security risk analysis. *Computers & Security*, 29(6), p.659–679.

Sommestad, Teodor et al., 2011. Security mistakes in information system deployment. *Information Management & Computer Security*.

Stephenson, P., 2010. Vulnerability assessment. *SC Magazine.*

Stewart, A, 2004. On risk: perception and direction. *Computers & Security*, 23(5), pp.362-370.

Still Secure, 2011. ProtectPoint. Available at: http://www.stillsecure.com/services/vulnerability_management.php [Accessed February 13, 2011].

Symantec, 2011. SecureFusion Portal. Available at: http://www.gideontechnologies.com/SecureFusion_portal.asp [Accessed February 13, 2011].

Tenable, 2011. Nessus. Available at: http://www.nessus.org/nessus/ [Accessed February 13, 2011].

Trustwave, 2011. Trustwave. Available at: https://www.trustwave.com/vulnerabilityScanning.php [Accessed February 13, 2011].

Venkatesh, V. et al., others, 2003. User acceptance of information technology: Toward a unified view. *Management Information Systems Quarterly*, 27(3), p. 425–478.

Wang, C., Balaouras, S. & Coit, L., 2010. *The Forrester Wave™: Vulnerability Management, Q2 2010*.

Wang, G. & Ng, T., 2010. The impact of virtualization on network performance of amazon ec2 data center. In *INFOCOM, 2010 Proceedings IEEE*. IEEE, p. 1–9.

Warner, R.M., 2008. *Applied statistics: From bivariate through multivariate techniques*, Sage Publications, Inc.

Welberg, S., 2008. *Vulnerability management tools for COTS software-A comparison*, Twente.

Werlinger, R. et al., 2010. Preparation, detection, and analysis: the diagnostic work of IT security incident response. *Information Management & Computer Security*, 18(1), p.26–42.

Ye, K. et al., 2010. Analyzing and Modeling the Performance in Xen-Based Virtual Cluster Environment. In *12th IEEE International Conference on High Performance Computing and Communications (HPCC),*. IEEE, pp. 273-280.