SCHUBERG PHILIS

Seccubus

Seccubus

Repeated vulnerability scans made easy

for:	Public release
date:	18 March 2010
version:	1.0
author:	Frank Breedijk
status:	Public

Schuberg Philis BV
Star Parc,Boeing Avenue 271
1119 PD Schiphol-Rijk
T +31 20 750 65 00
F +31 20 750 65 50
The Netherlands
www.schubergphilis.com

SCHUBERG PHILIS
MISSION CRITICAL OUTSOURCING

Table of contents

1 Introduction

During my work as security engineer at Schuberg Philis I repeatedly made use of vulnerability scanners like Nessus and OpenVAS. To me these scanners came across as real power tools unfortunately they are nowhere near perfect. Every scan results in a 'report', in one form or the other, with a large number of findings. Each of these findings will have to be investigated in order to determine if the finding is:

1. an issue
2. not an issue or
3. a false positive.

These investigations take time and effort. Because of this involvement in time and effort, vulnerability scans are often conducted on an ad-hoc basis, e.g. before go-life of an infrastructure or after a major change. However the dynamics of IT, where change is the only constant, made me want to regularly perform vulnerability scans. Obviously with the intention to spend as little time and effort as possible. This challenge has pushed me to write Seccubus, a tool to schedule vulnerability scans and process their results more easily.

1.1 What is a vulnerability scanner?

In the introduction of this paper I regularly used the term vulnerability scanner, but just to be clear; what do I mean by vulnerability scanner? A vulnerability scanner is a software program that aims to find vulnerabilities in software or an infrastructure, often by simulating an attack. Nessus and OpenVAS are well known vulnerability scanners, these two programs are aimed at finding vulnerabilities in infrastructures over the network.

Nessus and OpenVAS use a five step approach to finding vulnerabilities:

1. Determine if an IP address is active on the network. This is done by using technologies such as ping, arp scan, or a simple port scan.
2. Try to determine which services are offered by the IP address and which operating system is used.
3. Determine if known vulnerabilities are present on the system. Technologies to determine the presence of vulnerabilities range from comparing the version number of the daemon to a list of known vulnerable version , but also by performing step 4
4. Simulate or actually abuse the vulnerability to determine its existence
5. Report findings.

Known versus unknown

Unfortunately these scanners can only find known vulnerabilities. Known means in this context, vulnerabilities which have been programmed into the tool. A good penetration tester will, aided by his human creativity be able to find vulnerabilities for which automated test do not, yet, exist. These scanners can also impact the availability of the tested infrastructure. No vendor can 100% guarantee that the test subject will not be adversely affected by the test.

1.2 Why scan?

The potential risk of a vulnerability scan is still used as an argument to not scan at all or to only have these test performed by a security testing company. While I understand these concerns I can really

recommend companies and IT departments to perform their own scanning. For two reasons:
A; the risk is not that high
B; anybody can scan you.

Abybody can scan you?

There are plenty of freely available vulnerability scanners and penetration testing tools: Nessus, OpenVAS, NMAP, Nikto, Metasploit and others can be freely downloaded and thus used by just about anyone. And, even though it's illegal to use these tools on an infrastructure without the owner's permission, anybody who has ever seen a firewall log knows that just about anything connected to internet is scanned regularly. You could therefore argue that the information obtained from a scan is publicly available and since the information is 'on the street' you might as well get your own copy.

Scanning dangerous?

When performing a vulnerability scan, one has to weight all aspects of the information security triangle (Confidentiality, Integrity and Availability). During a scan, the availability of the test subject may be reduced, but at least cannot be guaranteed 100%; however you will get an accurate picture of all three aspects of the triangle.

The risk of a negative impact on the availability has to be put in context. If the system is connected to the internet it is very likely that it has been scanned previously without the oowners knowledge or consent. Also, it is my experience that these scans do not often significantly disrupt an infrastructure. In the 4.5 years that I have frequently used these tools, I have only really disrupted an infrastructure twice. In both cases where was no structural damage to any systems and the scan could be resumed later with a lowered intensity.

Vulnerability scanning is not the only activity that can cause disruptions to an infrastructure. Changes e.g. can also have a negative impact. By planning vulnerability scans in the same way as planned changes, they can often be fitted into the schedule.

2 More scanning, more work?

Testing an infrastructure for vulnerabilities should not be a one-off activity. Every single IT infrastructure I know changes. But, even if an infrastructure does not change, the threat landscape around it changes, e.g. new vulnerabilities are discovered daily. If a system is scanned today, and no vulnerabilities are found, the same system may turn out to be vulnerable tomorrow when we test it with and updated scanner.

Performing a vulnerability scan is easy; the vulnerability scanner software itself does most of the work. Unfortunately analyzing the findings of a scan is a lot of work. I have scanned an infrastructure with 130 IP addresses that offer no services to the internet. Yet the report generated by the vulnerability scanner consisted of 52 pages that contained over 400 findings. Even with sufficient experience with vulnerability scanning digesting such a report takes two hours or more and writing a formal report would take double that time. Clearly this does not scale well.

2.1 How does Seccubus improve this?

Seccubus is a tool that allows OpenVAS or Nessus vulnerability scans to be executed at set times. But besides this Seccubus reduces the time needed to analyze subsequent scans of the same infrastructure by computing the delta between the results of the current and previous scan.

Let say that we are scanning a new infrastructure for the first time, the scanner runs and sends its results back to Seccubus. Seccubus will produce the standard scanner reports and make them available for download in the Seccubus web interface.

Besides standard reports, Seccubus will also parse the scan results and put each individual finding in the web interface and assign it the status ‘New’. It is now up to the assessor to assign a new status to these findings. ‘No Issue’ if the finding does not pose a security risk or ‘Open’ if the finding does.

After a while, hopefully after some of the findings have been addressed, another scan can be performed. Seccubus will again import the findings in the web interface, but this time the status is dependant or the previous scan. Seccubus will change the status of the finding to:

- New – If finding was not present in previous scan.
- Changed – If finding was present in previous scan, but has changed
- Gone – If finding was present in a previous scan, but not in current scan.

Instead of having to examine all findings again, the assessor now only has to deal with the findings with these three statuses.

2.2 What is the gain?

Automating vulnerability scanning with Seccubus has the following advantages:

- Scans can be scheduled and start without the need for a human to ‘push the button’;
- The effectiveness of scanning is improved. Less effort is spent on scanning and more scans can be done with the same resources;
- The quality of the analysis improves; since less time is spend on mundane tasks.

2.3 Case: Schuberg Philis

Since I work at Schuberg Philis we were also the first to use Seccubus in August 2007. We scan all external IP addresses of all our customers. In total over 4000 IP addresses, resulting in a total of 8777 findings. Without Seccubus we would simply be unable to do this efficiently.

2.4 Seccubus' vital statistics

Program name: Seccubus
Website: www.seccubus.com
Mailing list: seccubus-list@seccubus.com
License: GPLv3
Copyright holder: Schuberg Philis
Author: Frank Breedijk
Support: Via website and mailing list
Downloads so far (March 2010): 2370

3 Conclusion

Seccubus automates execution and analysis of vulnerability scans so more scans can be performed with the same resources while maintaining accuracy. More scanning can be performed in less time. Because infrastructures can be scanner more often, more vulnerabilities are identified and remediated. This means the overall security is increased.

4 About the author

Frank Breedijk B ICT is currently employed as Mission Critical Engineer Security at Schuberg Philis, a leader in mission critical outsourcing services. He is responsible for the technical information security of Schuberg Philis' services, including security awareness, vulnerability management, internal security consultancy and technical audits and the development of Seccubus.

Frank has been professionally active in IT since 1997 when he stared as a programmer for PTS Software. His career in IT security stared in 2000 when he became ICT Security Officer for InterXion. Frank has worked as IT security consultant and managed Unisys' Security Operation Center for managed security for EMEA.

Besides his day job Frank is active on Twitter as @Seccubus, writes blog entries for www.Cupfighter.net and develops and maintains Seccubus.

He can be reached via his Twitter account or email him at fbreedijk@schubergphilis.com