



## **Doing More with Less: Nessus.**

*by Christian Houle CISSP, TICSA*  
[christian@localareasecurity.com](mailto:christian@localareasecurity.com)

**D**uring the last few years, I have been fascinated with the subject of Commercial vs. Open Source information security tools. One of the most interesting things that I have noticed over the years is the amount of information security tools that require some serious dollars to be invested. I believe it is time to rediscover an Open Source vulnerability assessment tool that has been around for quite some time and should not put a serious whole in your wallet.

Before we continue, I would like to expound upon the word 'TOOL'. Too often, people are expecting a silver bullet. An analogy would be to fix all of your cars problems with only a wrench at your disposal. This is a 'TOOL' that can help you in the process of risk management, specifically risk identification.

Nessus is well known for its capabilities as a vulnerability assessment scanner. Most of us mortals understand Nessus as a vulnerability finder/identifier on workstations, servers and other networked devices. By taking a different approach to employing Nessus, we can perform a greater variety of activities that can aid an organization in more than just identifying a vulnerability on a server.

Nessus can be used in various configurations including privileged vs. non-privileged mode. This can be loosely compared to one of the differences between a network based vulnerability assessment and a host based vulnerability assessment tool. The first normally has an external, no knowledge approach to the assessment. The second has internal and valid knowledge (account) directly on the system. This knowledge and privileged approach permits us to gain valuable information that can help us better understand the potential risk.

I will not reiterate what has already been said, so here are a few documents on how and why to perform assessments with Nessus from a privileged stand point on Windows Operating Systems.

**Why and How:**

[http://www.nessus.org/doc/nessus\\_windows\\_scanning.pdf](http://www.nessus.org/doc/nessus_windows_scanning.pdf)

**Detailed How-to:**

[http://www.nessus.org/doc/nessus\\_domain\\_whitepaper.pdf](http://www.nessus.org/doc/nessus_domain_whitepaper.pdf)

The following list contains some suggestions/ideas on how Nessus can be used in ways that you might not have thought about. There are a number of different situations where Nessus can be called upon and yield interesting results:

- **Remote Virus Detection**
- **Remote Service Identification**
- **Improper Configurations detection**
- **Policy Compliance Identification**
- **Rogue Wireless Access Point detection**
- **Distributed Scanning**
- **Company wide Threat Assessment**
- **Network Service Mapping**

Let's briefly discuss each of the above. . .

### **\*Remote Virus Detection\***

Nessus already has a number of scripts/plugins that can detect viruses and backdoors left by viruses. Now this does not mean that you can throw away your anti-virus solution. This simply means you can add value to your existing solution by proactively searching and identifying viruses before they actually start performing Denial of Service attacks or other mischief. In an unprivileged assessment situation, Nessus will be mostly limited to identifying backdoors created by a virus. In a privileged environment, Nessus will be able to detect viruses that do not necessarily have a backdoor, but have installed themselves on your system. One of the most common ways that this is achieved is searching through the remote registry for specific keys.

### **\*Remote Service Identification\***

Nessus already has a number of scripts/plugins that can enumerate and identify services running on a remote host. This means you can add value to your existing solution by proactively searching for and identifying unauthorized services. In an unprivileged assessment Nessus will be effective at identifying a wide range of services. In a privileged environment, Nessus will be able to enumerate running Windows services or running processes as well.

**An interesting unprivileged example:**

*In a corporate environment it might be valuable to more than just the IT security people to know who is running KAZAA. I'll bet the people who run the network would love to know where and even regain a percentage of the bandwidth lost to non-business related file sharing services. Or who is wasting time using instant messengers such as AOL, Yahoo, MSN, etc.*

## **\*Improper Configuration Detection\***

Nessus already has a number of scripts/plugins that can identify improper configurations. This means you can add value to your existing solution by proactively searching and identifying specific configuration problems. In an unprivileged assessment Nessus will be effective at identifying a wide range of configuration issues. In a privileged environment, Nessus will be able to get that extra “inside” information.

### **An interesting unprivileged example:**

*Although it probably never happens in your network, it might be important to know that a core router in your infrastructure has SNMP enabled with default read and write community strings. Setting up a non service affecting SNMP sweep of the whole network is easy as long as you plan properly.*

## **\*Policy compliance\***

Nessus has a limited number of scripts that directly look into policy compliance issues. Although with knowledge of one of the popular programming languages, the Nessus Attack Scripting Language and the existing 2000 scripts to learn from, it is possible to create your own policy compliance verification scripts.

Such a script running in a privileged environment could confirm that the anti-virus service is currently running on all Windows PCs. You could fancy up the script and include registry read capabilities in order to “grab” the version of the anti-virus signatures database/file. With the information gathered, you are in a better position to determine the risk faced by virus XYZ to that system.

## **\*Rogue Wireless Access Point detection\***

Nessus has a few scripts that can help you in identifying over 40 different types of wireless access points and bridges through a wired network. Once more you can add value by proactively searching and identifying rogue WAPs. The detection of the access points rely on OS detection and/or service recognition. One of the main problems faced with searching for WAPs is that most of them have firewalls protecting them by default, thus dropping all packets of information sent. An interesting approach is to send unusual packets to trigger even the smallest TCP exchange permitting OS detection guess. There is definitely more work to be done here but it's a good step in the right direction.

**A paper on why and how to detect WAP with Nessus:**

[http://www.tenablesecurity.com/white\\_papers/wap-id-nessus.pdf](http://www.tenablesecurity.com/white_papers/wap-id-nessus.pdf)

## **\*Distributed Scanning\***

Nessus client/server based architecture permits distributed scanning. This can be achieved with the help of commercial or non-commercial web based consoles which provide a centralized and simplified control of your vulnerability assessments.

The magic of distributed assessments is that you can start to identify specific or multiple risks much faster while limiting the impact on your network resources.

**More information on distributed scanning:**

[http://www.tenablesecurity.com/dist\\_vuln.html](http://www.tenablesecurity.com/dist_vuln.html)

**Commercial:**

<http://www.tenablesecurity.com/console.html>

**Open Source (License: GNU/GPL):**

[http://www.inprotect.com/download/nessus\\_automation.pdf](http://www.inprotect.com/download/nessus_automation.pdf)

## **\*Company Wide Threat Assessment\***

Effective threat assessment can be performed by adding some of the previously mentioned suggestions together. Nessus is relatively fast when it comes to releasing scripts for new issues. Using the freshly released plugin and your distributed architecture, you will be able to quickly determine the number of systems that could be affected/infected by the next worm or vulnerability.

**Interesting papers on how to set up an open source distributed scanning architecture:**

[http://nessus.org/doc/detached\\_scan.html](http://nessus.org/doc/detached_scan.html)

[http://nessus.org/doc/diff\\_scan.html](http://nessus.org/doc/diff_scan.html)

[http://nessus.org/doc/kb\\_saving.html](http://nessus.org/doc/kb_saving.html)

## **\*Network Service Mapping\***

Along the lines of wide scope threat assessment, Nessus connected to a backend database can store an interesting amount of data about your network including services. How about a different type of service detection verification every month of the whole network? Today we can perform a simple probe to determine the versions of all our known and unknown web servers in our network. Next month, we can look at SMTP services/servers. Why?

1. You have a better understanding of what is on your network.
2. You can start assessing risk much faster (database query) when the next vulnerability is released knowing it affects only an old version that is no longer being used on your network.

A number of these suggestions can be expanded to meet your needs, especially if you can master the Nessus Attack Scripting Language (NASL). If you do, share your scripts with the community. Most of the suggested ideas in this document are made possible because of the excellent Open Source programs and community supporting them. The non-commercial version, does not limit the amount of hosts you can audit. This makes investigating a large network infrastructure for single or multiple risks inexpensive and accessible.

I am positive that there are a number of other situations where Nessus can be very useful.

If you have an idea, comment or suggestion, please contact me ([christian@localareasecurity.com](mailto:christian@localareasecurity.com)).