

## Introduction to Nessus

by: Mitchell Rowton, 11/22/2004

<http://www.securitydocs.com/library/2730>

[Introduction to Nessus](#)

[Features of Nessus](#)

[Overview of Assessment Process](#)

[Nessus Server Installation](#)

[Configuring Nessus](#)

[Updating Nessus Plug-Ins](#)

[Using the Nessus Client](#)

[Starting a Nessus Scan](#)

[Generating Reports](#)

[Conclusion](#)

### Introduction

There are a number of security scanners available. Most are vendor specific and charge by the number of IP addresses it can scan. The most popular alternative to these scanners is Nessus.

Nessus is public domain software released under the GPL. Nessus is designed to automate the testing and discovery of known security problems. Allowing system administrators to correct problems before they are exploited. Historically, many in the corporate world have frowned on such public domain software, instead choosing "supported" products developed by established companies. Usually these packages cost thousands of dollars and the license is based upon the number of IP addresses scanned. However, many in the corporate world are now starting to realize that public domain software, such as Nessus, NMap, Apache, and MySQL, is often superior to similiar commercial products.

One of the very powerful features of Nessus is its client server technology. Servers can be placed at various strategic points on a network allowing tests to be conducted from various points of view. A central client or multiple distributed clients can control all the servers. The server portion will run on most any flavor of Unix. It even runs on MAC OS X and IBM/AIX, but Linux tends to make the installation simpler. These features provide a great deal of flexibility for the penetration tester. Clients are available for both Windows and Unix. The Nessus server performs the actual testing while the client provides configuration and reporting functionality.

### Features of Nessus

#### 1 **Up-to-date security vulnerability database**

The Nessus security checks database is updated on a daily basis and can be retrieved with the command `nessus-update-plugins`. An RSS feed of all the newest security checks allows you to monitor which plugins are added and when.

#### 1 **Remote AND local security.**

Traditional network security scanners tend to focus on the services listening on the network - and only on these. Now that viruses and worms are propagating thanks to flaws in mail clients or web browsers, this conception of security is getting outdated.

Nessus 2.1 is the only security scanner that has the ability to detect the remote flaws of the hosts on your network, but their local flaws and missing patches as well - whether they are

running Windows, Mac OS X or a Unix-like system.

#### | **Scalable**

Nessus has been built so that it can easily scale down to a single CPU computer with low memory to a quad-CPU monster with gigabytes of RAM. The more power you give to Nessus, the quicker it will scan your network.

#### | **Plug-ins**

Each security test is written as an external plugin, written in NASL. This means that updating Nessus does not involve downloading untrusted binaries from the internet. Each NASL plugin can be read and modified, to better understand the results of a Nessus report.

#### | **NASL**

The Nessus Security Scanner includes NASL, (Nessus Attack Scripting Language) a language designed to write security test easily and quickly. NASL plugins run in a contained environment on top of a virtual machine, thus making Nessus an extremely secure scanner.

#### | **Smart service recognition**

Nessus does not believe that the target hosts will respect the IANA assigned port numbers. This means that it will recognize a FTP server running on a non-standard port (ie: 31337), or a web server running on port 8080. Nessus is the first scanner on the market to have implemented this feature for all the security checks (and has been copied by many since then).

#### | **Multiples services**

If a host runs the same service twice or more, Nessus will test all of them. Believe it or not, several scanners on the market still consider that a host can only run one server type at once.

#### | **Full SSL support**

Nessus has the ability to test SSLized services such as https, smtps, imaps, and more. You can even supply Nessus with a certificate so that it can integrate into a PKI-fied environment. Nessus was one of the first security scanner on the market to provide this feature.

#### | **Non-destructive OR thorough**

Nessus gives you the choice between performing a regular non-destructive security audit on a routinely basis, or to throw everything you can at a remote host to see how will it withstand attacks from intruders. Many scanners consider their users to be too inexperienced to make that kind of choice, and only offer them to perform "safe" checks.

#### | **The biggest user base**

The most pessimistic computations, based on the number of downloads every day, give Nessus at least 50,000 users worldwide, but there might be even more - after all, Nessus is downloaded over 2,000 times every day

Our huge user base allows us to get the best feedback regarding security checks - and therefore to offer security checks which are reliable, non destructive and not prone to false positives.

#### | **Proven maturity**

The first public release of Nessus was in [1998](#). The technology behind it has been extensively tested and proven over time, on huge networks.

## **Overview of the Nessus Assessment Process**

While running Nessus you are doing a vulnerability assessment (or audit). This assessment involves three distinct phases.

## Scanning

In this phase, Nessus probes a range of addresses on a network to determine which hosts are alive. One type of probing sends ICMP echo requests to find active hosts, but does not discount hosts that do not respond - they might be behind a firewall. Port-scanning can determine which hosts are alive and what ports they have open. This creates a target set of hosts for use in the next step.

## Enumeration

In this phase, Nessus probes network services on each host to obtain banners that contain software and OS version information. Depending on what is being enumerated, username and password brute-forcing can also take place here.

## Vulnerability Detection

Nessus probes remote services according a list of known vulnerabilities such as input validation, buffer-overflows, improper configuration, and many more.

## Nessus Server Installation

One feature of Nessus is its client server technology. Servers can be placed at various points in a network allowing tests to be conducted from various points of view. A central client or multiple distributed clients can control all the servers. The server portion will run on most any flavor of Unix. The Nessus server performs the actual testing while the client provides configuration and reporting functionality.

Nessus offers a easy automated installation:

```
lynx -source http://install.nessus.org | sh
```

The above command should also be used periodically to upgrade Nessus as new versions are regularly released. You will be questioned about proxy servers, a download method (www or CVS), and the branch of the development tree to use; most of the time the defaults are the best choice. This is the simplest method of installation however; you are effectively giving the install.nessus.org server temporary root privileges.

For informaiton on how to install Nessus from scratch visit:

[http://www.nessus.org/nessus\\_2\\_0.html](http://www.nessus.org/nessus_2_0.html)

## Configuring Nessus

Once the server is installed, some basic configuration is required. First, if the server isn't started type *nessusd -D* Then, you need to add a user. A new user can be added by the *nessus-adduser* command. The script will question you for the authentication method. Authentication can be performed by several means, however a password is the simplest. The next question queries about rules to restrict the user account. When used across an enterprise, a user can be restricted and only allowed to scan specified IP addresses. However, for most uses this will be left blank, allowing the user to scan anything. A certificate also needs to be generated as well to be used to encrypt the traffic between the client and server. The *nessus-mkcert* command accomplishes this.

## Updatting Nessus Plug-Ins

Plug-in updates should be done frequently. New vulnerabilities are being discovered and disseminated all the time. Typically after a new vulnerability is released to the public, someone in the Nessus community writes a NASL plug-in, releases it to the public and submits it to

www.nessus.org. It is then reviewed by the developers and added to the approved plug-in list. For high risk, high profile vulnerabilities a plug-in is often released the same day the vulnerability information is publicly released. Updating plug-ins from the maintained list is fairly simple involving a simple command: *nessus-update-plugins*. This command must be done as root.

## Using the Nessus Client

There are three primary Nessus clients. This tutorial will cover using the native Unix GUI version, which is installed at server install time. In the native client, enter the server IP, username and password (created with the *nessus-adduser* command) and hit login.

If you have trouble logging in the try the following steps:

1. Ensure the server daemon is running. Type: `ps -A | grep "nessusd"`
2. If "nessusd" does not exist, start the nessus daemon with the command: `nessusd -D` (assuming that "nessusd" is in your PATH and you have enough privileges to start "nessusd".)
3. If "nessusd" does exist, verify the port number in use. The command `netstat -na` may be useful in this. The traditional port is 3001. The IANA assigned port is 1241.
4. Make sure that versions of the client and the server are in sync. Running a v1.0.x client against a v1.1.x server will not work

## Starting a Nessus Scan

After you connect the Nessus client to the server then you should take a look at the different plugins available in the Plugins tab.

Use the Filter button to search for specific plugin scripts. For example, it is possible to search for vulnerability checks that have a certain word in their description or by the CVE name of a specific vulnerability. It is up to the author of each specific vulnerability check to make sure he provides all appropriate information and places his script under the proper category. As you will note by looking at the descriptions of some of the vulnerability checks, some authors do not do a good job of filling in this information, so be careful.

There are also buttons to "Enable all plug-ins" or just "Enable all but dangerous plug-ins". Note that the author of the plug-in decides if it is dangerous or not. Most of the time, this has been very well chosen. However there are instances where the plug-in causes a DOS but it is not listed as dangerous. The native client denotes dangerous plug-ins with a caution triangle.

When starting a new scan session there are several optional areas to become familiar with (depending on your needs.) The wise decision is to go with the default options and test on non-production devices.

## Generating Reports

When Nessus finishes its scan, it will present you with a report. You can save it in a variety of formats: HTML (with or without graphics), XML, LaTeX, ASCII, and NBE (Nessus BackEnd). The items with a light bulb next to them are mere notes or tips that provide information about a service or suggest best practices to help you better secure your hosts. The items with an exclamation next to them are findings that suggest a security warning when a mild flaw is detected. Items that have the no-entry symbol next to them suggest a severe security hole. In case you are wondering, the authors of the individual scripts used by the Nessus plugins decide how to categorize the findings.

## Conclusion

To see how a particular vulnerability scan works, take a look at its corresponding .nasl script file located in /usr/local/lib/nessus/plugins. This can assist you in determining whether or not a finding is actually a false positive. As mentioned previously, you should always test new scanning preferences on a non-production devices. The author of this tutorial has crashed several production servers by not following this advice (even with safe checks enabled, and no dangerous plugins enabled).