;login:

THE MAGAZINE OF USENIX & SAGE
November 2000 • volume 25 • number 7

THEME ISSUE: SECURITY

edited by Rik Farrow

inside:

NESSUS: THE FREE NETWORK SECURITY SCANNER

USENIX & SAGE
The Advanced Computing Systems Association &
The System Administrators Guild

nessus: the free network security scanner

by Renaud Deraison

Renaud Deraison was tired of people complaining of the cost needed to bring their network to a decent level of security, so he started to write free tools to help them to achieve their goal at a much lower cost.

<deraison@nessus.com>

and Jordan Hrycaj

Jordan Hrycaj works as independent security consultant and joined the Nessus project in late 1998. He believes that clever system solutions are always born in the mind rather than designed with the latest development tool.

<jordan@nessus.com>

Renaud Deraison and Jordan Hrycaj are the authors of Nessus and the founders of the Nessus Consulting S.A.R.L.

A network scanner is a tool for analyzing network services, available on a given set of systems. With Nessus, a new breed of scanners has been published capable of running real attacks, often called exploits, in order to determine that well-known system deficiencies can be exploited when running the attack against the scanned systems.

History

When Nessus was born back in 1998, it was just cool to have a free network scanning and attacking tool with design goals similar to SATAN, written by Wietse Venema and Dan Farmer. Right from the start, Nessus was set up as a client-server tool endowed with its own communication protocol. The scanning and attacking workload was put onto the server, and the presentation of the data was done by the client, very similar to the design of SATAN.

In addition to that, the client realized better online control. So each host under scan and attack could be released from the scanning, individually, at any time. SATAN's design launched the server and waited for the scanning to complete, without any control over the process. The attacks used by Nessus only test for vulnerabilities and do not actually perform a "break-in."

Nessus was planned and introduced to be publicly supported as a free software project. Seen from an organizational standpoint, this only meant that the source code of both the client-server platform and the plugin code database (the implementation of the attacks and the scans) are open for public use and discussion.

Licensing Concept and Support Considerations

Nessus has been released under the GNU Library General Public License (renamed to Lesser GPL in 1999), which might be further restricted, partly by some contributions to Nessus.

Within one tool, a freely available set of working proof-of-concept attacks has been published. This is still unique, as the size of the Nessus database is far beyond that of any other scanner, even commercial collections.

The authors of Nessus strongly believe in the free and open-source approach. This has a clear impact on the general acceptance of and contributions to Nessus. Many bugs and exploits are probably found by individuals, favoring a public and open audience rather than making a quick buck with a company that solely handles the exploits as classified information.

The software can be deployed, tested, and modified freely. There is public bug-track management and a searchable mailing list. Additionally, professional software support is offered for commercial users to provide (legal) support contracts.

Implementation Notes

With the scanning and attacking database, Nessus aims to be as complete as possible. It currently performs over 500 security checks. This includes advanced Windows NT checks such as testing for permission to access the registry keys remotely, or for inappropriately shared partitions.

Nessus has been designed to be easily installed and handled by a user or an operator.

While attacking, the intention is not to miss any vulnerabilities whatsoever. For instance, nobody prevents you from opening a Telnet service on port 32 rather than 23, and a testing tool should be able to find that out. Nessus will actually probe open ports with unusual port addresses to see if Telnet or something like it is running there. Being that flexible has not been common for a long time and probably is still uncommon, especially with commercial software.

Nessus does not guess a host or operating-system type by reading the greeting message banner of the Telnet program. Long after QUESO and NMAP introduced the IP-stack fingerprinting approach, the banner method is still common practice with many other tools.

A Strategic Tool

As of today, Nessus has been used as a tool to enforce the security policy of a company site, institution, or organizational entity. Nessus goes much further than answering questions like "Does my firewall have the particular bug reported in the BugTraq list the other day?"

The Nessus project aims to provide a tool to check out and analyze the network as seen from a security standpoint that is

- comprehensive and reliable
- distributed
- continuously up-to-date
- well known
- cost effective

In the strategic setting up and running it has some similarities with network probes commonly installed and used to monitor data and voice traffic in quality and quantity.

Although the resulting reports are not always simple to grasp by nature, Nessus has been designed to be easily installed and handled by a user or an operator. It is possible to control a session in batch mode as well as with a full operator dialog. The server poses access restrictions upon the controlling operator using public-key technology. Once installed, the operator can have full and individual control over a farm of servers, possibly without the need to remember passwords (of course, the workstation needs physical access security, unless the keys are protected by a pass phrase).

With the arrival of public bug-registration sites like CVE, Nessus easily integrates and contributes to the worldwide network of security-relevant information systems that are freely available for everybody.

Architecture

CLIENT-SERVER COMPUTING

The server, named nessusd, is the smart part of the program, which is in charge of the security assessment, and is available for modern POSIX-like systems such as Linux, FreeBSD, OpenBSD, and Solaris. There might be more but they are not officially supported by the core team. The client, as supported by the same team, is additionally available for the Microsoft Windows releases 9x, NT4, and W2K.

The client is the controlling front end to the server. The communication between the server and the client is encrypted. Session negotiation and authentication on the server is based on public-key encryption technology.

The nessusd server manages its own user and access database, so different scan and attack privileges can be configured. It is, for example, possible to configure the nessusd server so that each user can test only her or his own computer.

PLUGINS

The nessusd server is an application platform for running a series of network-based test programs and attacks, the results of which are collected in a common database. These programs, called plugins, have access to this database. Apart from storing results, they also use it for communication and optimizing tests.

In a few cases, plugins are dynamically linked program fragments (usually called shared objects, or shared libraries.) Most commonly, though, they will be interpreter scripts in a language, called NASL (the Nessus Attack Scripting Language). These scripts can be run immediately and independently of any operating system by nessusd.

The NASL interpreter handles the communication between the scripts transparently through the database, mentioned above. The script language is limited in its power to implement applications different from network tests and attacks. It is not designed to run in a sandbox as TAINTPERL and Java do, but does control what actions can be carried out through the design of the interpreter.

Thanks to this architecture, updating a set of security checks for nessusd is usually just a matter of downloading some files and copying them to the appropriate place on disk. And this task is automated by shell scripts like nessus-update-plugins, which retrieves all the newest NASL scripts, installs them at the proper location, and reloads them into the nessusd server. The latest NASL scripts available are regularly published on the Nessus script page.

Deployment Topology and Interfaces

Currently, Nessus supports only the deployment of standalone nessusd servers with multisession support. Secure server-to-server communication for distributed attacks is possible but so far has been implemented at transport level only.

There are well-defined library APIs for the NASL interpreter and the PEKS-encrypted communication channel API. There is also a well-defined text form used for storing the scanning and attacking results. A database API has been under discussion for some time.

Availability Notes

The whole Nessus Package is about 16MB in source code; extra library packages needed, like gmp or pcap add about 4MB. The exploits-and-attack database is currently somewhat larger than 2MB of source code. Altogether, the gzipped sources make up a bit more than 3MB.

There is a network of worldwide FTP mirrors; the easiest way to access them is to browse any of the Nessus Web sites (<*http://www.nessus.org*> being the primary one). On these sites, some online installation instructions are also available, as well as the screen shots of a sample session.

Although version 1.0 was released not so long ago, Nessus is under active development. The next major release will have better handling of large networks (over 10,000 hosts), will offer the ability to do distributed scans, and will have better multilingual support. (Currently, most plugins have English and French descriptions and messages.)

Summary

Nessus is a free network-security scanner and attack tool with a clear strategic focus. Its main goal is to help enforce the security policy of the network site that is tested and attacked. Designed as a server-client system, many servers can play the role of monitoring devices controlled by one or more client operators.

Nessus is not a one-shot or standalone tool. It can be used that way, but is designed with clear interfaces and APIs. This allows further development and integration at a public or individual level.

Nessus has been developed in Europe, so there are currently no export restrictions whatsoever.