TENABLE Network Security®

Reliability and Uniqueness of Tenable Nessus Technology

February 7, 2007
(Original paper first released in 2002)

Table of Contents

Introduction

Tenable's Nessus Vulnerability Scanner is counted among the world's premier security scanners. An active security scanner is a piece of software that connects to network machines and determines if the machine is vulnerable to any flaws which might place it at risk of being successfully attacked. The job of the Nessus Vulnerability Scanner is to help the security team and administrators gain an understanding of the current level of security on the network.

The Nessus technology goes well beyond finding vulnerabilities and generating pretty reports. There are many unique features of the Nessus technology which can help any organization to assess and remediate threats. This document serves as a reference to highlight and explain the unique qualities of the Tenable Nessus technology. From a technical standpoint, Tenable has made many new changes to the core engine and scripting API (NASL) for improved efficiency and performance. When looking at scanning technologies, it is important to understand the technical merits of the scanner in order to ensure that you get the best results. Scanners are typically evaluated for their:

- Accuracy
- Stability
- Speed
- Ability to detect network and host-based flaws

Accuracy & Speed

Accuracy is More Important than Speed

The Nessus Vulnerability Scanner was engineered for accuracy. While speed is an important factor, the accuracy of a scanner is the true measure of its effectiveness.

Nessus 3, Tenable's latest release, contains substantial improvements over previous versions. Tenable has designed or reengineered many of the NASL features. At the time of this writing, NASL has been through two full revisions. NASL3 is **16 times** as fast as NASL2 and a full **256 times** as fast as NASL1.

Efficient Speed is Best

Raw speed, in and of itself, is not enough. A scanner should be both fast and smart. The Nessus Vulnerability Scanner utilizes **efficient** speed. Beginning with Nessus 2, Tenable engineered the scanner for both speed and efficiency. This trend has continued with Nessus 3. With Nessus 3, the network is truly the limiting factor. For a scanner to be truly efficient it must only initiate necessary network connections and then ensure that the returned data is valid and complete. Many scanners blast **every** host with **every** vulnerability check. In sending so much raw data, these scanners often miss valid responses from the clients that they are scanning. The Nessus Vulnerability Scanner is intelligent enough to gather information regarding the services and operating system **prior** to launching more sophisticated checks. This means that the Nessus Vulnerability Scanner is intelligent enough to limit irrelevant checks while giving full attention to relevant checks. This is, in a word – efficient.

What are some other ways that the Nessus Vulnerability Scanner differentiates itself from other scanners?

1. Nessus uses a “smart” receive function.

   That is, while other scanners will allow a scanned host to return data of nearly any length, the Nessus Vulnerability Scanner caps the input at a user-specified amount. So, for example, you could call the “recv()” function and tell it to only read 512 bytes from the scanned host.

   How is this an advantage? Other scanners do not cap the amount of return data. This leads to what is classified as “funny admin” or “hack-a-hack attacks”. That is, the remote administrator or user can interfere with the corporate scanner by purposefully flooding the scanning machine with bogus reply data. The Nessus Vulnerability Scanner stops these sorts of attacks by using timeouts on plugins as well as limits on input data. Further, the Nessus “receive” functions use a decreasing timeout in order to make a best effort in collecting the specified amount of data. So, for example, if you called “recv()” with a desired read length of 42 bytes and a default timeout of 5 seconds, the function would return immediately if a single packet with greater than 42 bytes of data were returned. If, on the other hand, the remote server was slow in returning data and was using a small TCP sliding window (or using a protocol that sends small bursts of data), the Nessus Vulnerability Scanner would note the first X incoming bytes and increment the timeout value to allow for more data to arrive. As soon as 42 bytes of data is received, the recv() function exits. This is what is meant by “smart recv”. It is a significant improvement over other scanners which use raw C/C++ with rudimentary or non-existent timeout algorithms.

2. Nessus uses protocol-specific speed enhancements.

   What does this mean? To improve speed, the Nessus Vulnerability Scanner respects the protocol. A lot of protocols advertise the amount of data they send (HTTP has a Content-Length field, SMB has a two byte packet length field and so on.). As a result, the Nessus Vulnerability Scanner has implemented a lot of protocols in NASL. A non-exhaustive list of these protocols would include, but not be limited to:

   - HTTP
   - SMB
   - RPC
   - NFS
   - FTP
   - SMTP
   - SNMP
   - Kerberos
   - DNS
   - SSH

   Some plugins still implement a protocol by themselves. However, when more NASL scripts use the same protocol, the Nessus Vulnerability Scanner nearly always creates a separate protocol API.

Other Advantages

Encapsulating the protocol allows us to take advantage of every one of its features quite easily. As an example, NASL scripts use HTTP KeepAlive requests when present, and then transparently use HTTP 1.0 or HTTP 1.1 depending on the remote host. This not only speeds up the Nessus Vulnerability Scanner, it also reduces traffic to the remote scanned machine. While other scanners will setup and teardown multiple TCP socket connections for a single check, the Nessus Vulnerability Scanner achieves these results, many times with a "single" session. The following are a few examples of the level of abstraction that NASL offers when writing security checks:

- FS/SMB: mount(), umount(), opendir(), readdir(), read() and close() are all totally independent from the underlying OS.
- FTP: ftp_close(), ftp_recv_line(), ftp_get_pasv_port()
- HTTP: http_keepalive_send_recv(), http_get(), http_post(), http_recv()
- SMTP: smtp_send_socket(), smtp_send_port(), smtp_recv_banner()

"Smart" UDP

UDP is a connectionless protocol. On a normal network these packets are often lost due to decreasing bandwidth, low port density, etc. During a scan, this is even more true. As scans use more bandwidth, the chance of losing UDP packets increases.

What does the Nessus Vulnerability Scanner do in order to elegantly and efficiently ensure that every effort is made to receive UDP traffic? The Nessus Vulnerability Scanner utilizes the same smart recv() function denoted above. Further, during the "timeout" period, The Nessus Vulnerability Scanner resends the original UDP packet every second until a response (ICMP unreachable or UDP response) is received.

Intrusiveness

A scanner that always crashes the scanned machines can be a problem for companies scanning "mission critical" systems. The Nessus Vulnerability Scanner believes in reusability of information found via other plugins. More specifically, the Nessus Vulnerability Scanner has implemented a Knowledge Base (KB) which can be updated by any plugin. So, for example, the scanner will not arbitrarily connect to a port in order to read a port banner or find a response. Instead, it will look in the KB to see if any other plugins have already found and recorded such a session. By reusing this information, the Nessus Vulnerability Scanner minimizes the number of connections to a given service port.

Service Port Diffing

The Nessus Vulnerability Scanner has a dedicated plugin (check_ports.nasl) which will examine all known open ports and notify the scan administrator if a service port is closed at the end of a scan. So, for example, if the scanner found port 23 (telnet) to be open and later closed, it would report on the port closing.

Safe Mode

The Nessus Vulnerability Scanner has a mode for safe_checks(). If this is enabled by the scan administrator, then the scanner will not attempt any overflow tests, deferring instead to a port banner check. This can be very useful when scanning older (legacy) machinery such as Mainframes and older Operating systems.

Stability

We alluded to this earlier, but it is worth repeating. Namely, the risk of a malicious admin is becoming greater and greater. By "funny admin", we mean: "a user or Administrator who takes active measures to interrupt a security scan". To do so, he may set up "rogue hosts" which will attempt to "kill" the network scanner itself. We do not want a rogue host to stall our scan (one host blocking the scan of the rest of the network). We do not want a rogue host to crash our scan (one host preventing the scan of all the other hosts). And finally, we do not want a rogue host to crash and execute arbitrary code on the scanning host.

How does the Nessus Vulnerability Scanner protect against these sorts of attacks (Unique to the Nessus Vulnerability Scanner)?

- Each scripts runs as a separate process with a total timeout.
- Each script has a limitation in memory.
- NASL offers a sandbox which utilizes dynamic memory allocation to prevent buffer overflows, array oversubscription, format string attacks, etc.
- Each host is tested in its own individual process. Scripts do share the same process space; however, due to the VM implementation of Nessus 3, there is a perfect containment of the scripts
- Every network-related loop has a counter.
- A NASL script can only use a maximum of 80 Megs of memory. This is a large value and, in actuality, most scripts use less than 512 kb.
- The NASL3 VM is very secure. A poorly written NASL script is not vulnerable to any buffer/stack overflows or memory corruption. This is due to the fact that the language itself prevents these sorts of errors from occurring.
- Nessus utilizes cryptographic "signing" of scripts. There are 2 levels of script functions. Harmless functions are those which cannot interact with the local system. Privileged functions (those which can interact with the local system) are signed by Tenable Network Security. This measure ensures that rogue scripts do not gain privileges on the local system.
- The Nessus "IDS Evasion" mode is no longer supported with Nessus 3
- Nessus 3 ships with more than 10,000 security checks.
- Protocol sanity checking – For example, if the remote HTTP server has more than 1024 lines of "header" information, the Nessus Vulnerability Scanner closes the connection. There are similar sanity checks for FTP and other protocols.

Decreasing False Positives

The Nessus Vulnerability Scanner looks at port banners and uses application and network-based fingerprinting. In addition, the Nessus KB can also be queried to find SNMP variables, port banners, OS fingerprints, etc. In using a hybrid approach with "information reuse", the Nessus Vulnerability Scanner elegantly makes a best guess regarding the scanned operating system and relevant security data which should be associated with the found fingerprint.

Nessus Plugins

With the release of Nessus 3, there are more than 10,000 plugin checks. Nessus plugins often include cross-references with Security Focus (Bugtraq ID), CVE, OSVDB, IAVA and more. Many Nessus plugins also include CVSS severity rankings. These CVSS rankings allow an organization to quickly categorize their level or Risk.

Nessus' Dustier Corners

The following functionality is built into the Nessus Vulnerability Scanner and Tenable Network Security has dedicated resources to the documentation of this functionality.

- SMB – The Nessus Vulnerability Scanner has its own SMB API. When scans are run with DOMAIN credentials, the scanner can read the registry, read any file on the hard drive, query user accounts, etc. That is, almost anything that can be done with SMB can be easily coded into the Nessus SMB API.
- WEB checks – Many competitors sell products which **only** check web applications. With the Nessus Vulnerability Scanner, this is but one of the core competencies.
- Spyware Checks – Yes, the Nessus Vulnerability Scanner can check remote systems for common Spyware and P2P packages.
- Mail and News – Another “core competency” built into Nessus server.
- Detection of Wireless access points – the Nessus Vulnerability Scanner will detect known Access Points via the legacy (or wired) network.
- Application Testing – The Nessus Vulnerability Scanner does layer 7 Application layer testing. This includes mirroring of website, common CGI blackbox overflow techniques, SQL Injection, Blind SQL Injection, and more.
- “DESTRUCTIVE” attack blackbox testing – What does a company do to test its homegrown (or not well known) applications? There are not many scanners which tests custom applications. Nessus will. The Nessus Vulnerability Scanner will attempt to detect the underlying protocol. Once the protocol is noted, it will scan it for common exploit mechanism. This may include generic buffer overflows, web-based form testing, generic VPN/IKE testing, etc. While these attacks are very dangerous, they can be well utilized in a lab testing environment to ensure that a modicum of security is present on the device. The Nessus Vulnerability Scanner has, on more than one occasion, found unique, previously unpublished bugs via its “DESTRUCTIVE” checks. This dedication to scanning for more than just the “low hanging fruit” separates the Nessus Vulnerability Scanner from other vulnerability scanners.

About Tenable Network Security

Tenable, headquartered in Columbia, Md., USA, is the world leader in Unified Security Monitoring. Tenable provides agent-less solutions for continuous monitoring of vulnerabilities, configurations, data leakage, log analysis and compromise detection. For more information, please visit us at http://www.tenablesecurity.com.

TENABLE Network Security, Inc.
7063 Columbia Gateway Drive
Suite 100
Columbia, MD 21046
TEL: 410-872-0555
http://www.tenablesecurity.com