System Vulnerabilities in the Enterprise and The Scavenger Project

By:

Matthew A. Kwiatkowski
Argonne National Laboratory

(This page intentionally left blank)

Table of Contents:

Introduction:

About two years ago, in late 2004, Argonne National Laboratory (ANL) was running a commercial version of a computer vulnerability assessment and remediation program. Deficiencies were found in that program for the Laboratory's needs, and features were asked of the company who wrote the program. It took months to get features added to the program because of the closed source. In addition, the types of technology used to code the program made integration with that commercial vulnerability program very difficult. What was decided was to write our own custom vulnerability assessment and remediation program. Thus, Scavenger was born.

Why Do Vulnerability Assessments and Tracking?:

To start, what is vulnerability scanning? Vulnerability scanning is very similar to port scanning and packet sniffing. Vulnerability scanning is using tools to identify weaknesses in an operating system or network to mount an attack against [10]. Like packet sniffing or network sniffing, vulnerability scanning scans all the systems that respond on a network and runs tests against those systems to find a weakness. If a weakness is found it is recorded so that a report can be generated. Along with the weakness and system it was found on there can be other data as well. For instance, discovery time, risk level, system information, serivice information, other weaknesses, and remediation information. The next logical question would be: Why do vulnerability scanning? According to the FBI Computer Crime Survey, 52% of companies surveyed had unauthorized use of computers in their organization. Also, 65% of attacks on networks and systems were virus related totaling to a financial loss of over 15 million dollars [1]. Usually, viruses take advantage of vulnerabilities in systems and networks. The FBI continues to say that only 43% of organizations use an Intrusion prevention system [1]. This means 57% of companies do not have an intrusion prevention application. The most likely reason is cost. The cost to purchase, implement and maintain a comprehensive intrusion prevention system can be prohibitive for some smaller organizations. The availability of a vulnerability detection and remediation tracking system may not be cost effective. The goal of this project was to provide an organization an effective, free option, a new tool called Scavenger.

Vulnerability assessment is a critical part of the comprehensive look at cyber security. The main idea behind vulnerability scanning is for administrators to find vulnerabilities in their systems before any of the "bad" guys do [6]. The FBI survey states that 82% of the respondents use security audits by internal staff as an effective technique to evaluate the security stance of an organization [1]. Cisco Systems takes the approach that an organization must do penetration testing and vulnerability assessment in some form or another. The scanning of computer systems is one major part of vulnerability assessment. They even say that the cyber team should think like a criminal and avoid the "Titanic syndrome": do not think that the ship was built so well it won't

sink. [3]. The same holds true for a network. Do not think that a perimeter firewall will be the final answer to an organization's cyber defense. The Internet is always changing, and so must one's cyber security defense mechanisms. Some examples of what Cisco thinks are dynamic considerations for vulnerability testing is:

1. Proliferation of viruses and Trojans
2. Wireless LANS
3. Complexity of networks
4. Frequency of software updates
5. Ease of hacking tools
6. The nature of open source
7. Reliance on the Internet
8. Unmonitored mobile users and visitors
9. Industry Standards
10. Cyber Warfare

The speed of today's new Trojans, viruses and worms is staggering. The Sasser virus was one of the most damaging viruses in 2004. It caused flights to be delayed and trains to come to a halt, yet, the virus was written by a German teenager [3].

Wireless LAN's are becoming more and more popular as mobile computing network speeds are increasing. Wireless networks go beyond typical physical boundaries and can extend beyond the physical walls of an organization.

The complexity of networks to support multiple protocols, services, and platforms intensifies the need to verify issues that may arise with such a diverse network that is designed to interoperate. Even web services have become more complex. Previously, a web server held information in a static page. Today, there are web portals, applications, and dynamic web content that is ever-changing. Web authoring and creating web applications is a normal business practice, which needs to be audited [3].

Software updates are so frequently released for all the platforms that exist on a network that it is becoming a challenge for system and network administrators to keep up with patching and compliance. Auditors need real-time information to verify that administrators are keeping up with patches. According to the FBI Survey, 70% of actions needed after a detected computer intrusion was simply to apply a patch [1].

Hacking tools have become easier to use [3]. The evolution of these tools are turning everyday people into crackers and fostering a whole new breed of cyber-criminals..

The nature of open source operating systems, applications, and services have given rise to freely distributed software. While this can be a good thing for organizations, the availability of source code to certain open source applications can help crackers identify potential backdoors and buffer overflows [3].

Since most companies and organizations rely heavily on the Internet for their public-facing identity, there must be a certain level of systems that is open to the Internet, including crackers. Also many software updates come from the Internet in an automated process, so companies cannot just be disconnected from the Internet.

With wireless visitor networks on the rise, monitoring these users and distinguishing them from the organization's employees is becoming harder to do with mobile computing becoming so popular.

Industry standards may dictate how an organization has to secure and report its network and system infrastructure. Some examples of reporting standards that are industry based are Health Insurance Portability and Accountability Act (HIPAA) and Sarbanes–Oxley Act of 2002 (SOX). For Argonne National Laboratory, we follow the Federal National Institute of Standards and Technology (NIST) guidelines and report to the Computer Incident Advisory Capability (CIAC) for the Department of Energy.

Finally, Cyber warfare is a reality today. Companies and federal institutions are attacked on a daily basis by outside competitors or countries seeking to gain unauthorized access into the networks and systems. This hacktivism usually focuses on the forward-facing websites of popular corporate dot-coms and government agencies for religious and political reasons [3]. Having a vulnerability scanner, penetration testing, and compliance tool solution solves most if not all of these problems.

To summarize, vulnerability scanning and penetration testing is an essential part of securing an organization's network and information systems. Now, how does one go about scanning a network? Microsoft suggests that a network scope its cyber attack targets ahead of time [4]. As we will see, one of Scavenger's requirements will be to make the application configurable to match an organization's infrastructure in an effort to ensure that the identified targets match the characteristics of the organization. Usually, the essential network and system issues to investigate include the following [4]:

1. Network and host discovery
2. Port scanning
3. Password attacks
4. Application attacks
5. Database attacks
6. Web attacks
7. Email attacks
8. Domain controller attacks
9. VPN threats

A single tool, Nessus, can help address all of these issues. Nessus is classified as a vulnerability scanner which is defined as: "A vulnerability scanner is a computer program designed to search an application, computer or network for weaknesses. The scanner systematically engages the target in an attempt to discover vulnerabilities. The program can be used either to find holes and plug them before they are exploited or to find holes and exploit them [7]. " We will see how the selection of Nessus functions as the core of the Scavenger product. We will also see that the default full list of plug-ins selected can address all these attack vectors. How can Nessus check for password complexity and things of that nature? The answer is that Nessus has compliance checks built into the application that can check for password complexity, registry settings on windows and Active Directory, UNIX audit tests for running processes, a user security policy, and the content of selected files [5]. Most popular operating systems are covered by the audit and compliance checks that are part of Nessus, such as all MS Windows NT 4.0 and higher, Mac OS X, Linux, HP/UX, and Solaris, just to name a few [5]. The criteria that Nessus uses for these compliance checks are based on industry standards set by the National Institute of Standards and Technology (NIST) and the National Security Agency (NSA) and by laws like Sarbanes–Oxley Act of 2002 (SOX) [5]. Nessus was also designed to

not require any sort of local authentication on a machine, specifically Windows, to perform an accurate vulnerability assessment of a system. Instead of verifying permissions, Nessus will attempt to exploit a known vulnerability in a safe manner to make a determination, if a host is indeed vulnerable [6]. This feature was important to the ANL Cyber Security Office, since, in many cases the Cyber Office does not have full administrative access to all ANL machines.

The Beginning of Scavenger:

The name Scavenger is purposely suggestive: the program "scavenges" the vulnerabilities from the network like a vulture and reports them to administrators and expects a response. There are two types of vulnerability scanners, passive and active. The nature of finding vulnerabilities and reporting them to the respective administrators, would classify Scavenger as a passive vulnerability scanner [8]. If one of Scavenger's goals were to fix vulnerabilities in real time, it would then fall into the active scanner class. Scavenger's broad project goals were the following:

1. Be Configurable
    a. Code the Scavenger application to be easily changed so that as the organization changes the application can change with it. Also, if new requirements are created during the application life cycle, they can be easily integrated into the product.
2. Provide real-time vulnerability scanning
    a. All ANL networks must be scanned in near real time. All nets must be included to make this a comprehensive solution.
    b. All VPN connections/upon connection must be scanned in real time. Education, notification, and controls must complement the scan. This will focus on home users who connect to the VPN to increase awareness and security with home users connecting to ANL networks.
    c. All visitor connections/upon network registration should be scanned before they are allowed onto the network. This will enable us to create a process that will allow and deny visitors access to ANL visitor networks based on the heath of their machines. The more secure the machines are connecting to the visitor network, the fewer problems we have with rogue systems or compromised visitor systems on our visitor network.
3. Keep vulnerability false positives low. Provide a mechanism to allow Cyber Representatives to mark Scavenger results as false positives.
4. Keep track of scanning results in an Open Source Data Base MYSQL. We have in-house support mechanisms and expertise with MYSQL. Furthermore, with these software choices, acquisition costs will be nothing
5. Use an Open Source Scanner. Nessus was the Scanner of choice because it allows us to write custom checks, modify existing checks, integrate easily with other open source technologies, and has a track record [11] for having more accurate scanning results according to other National Laboratories.

6. Have a web interface for the front end
    a. Populate interface with scanner results. This allows us to take the raw scanner results, which are hard to read and put them in a format that is friendly to system and network administrators.
    b. Keep track of vulnerabilities across the Laboratory. ANL is distributed in its network architecture, so we wanted to keep track of individual divisional/departmental progress with keeping up with vulnerabilities. This will also enhance our reporting capabilities if we separate into multiple divisions.
    c. Provide Administrative functions for Scavenger Administrators. The Cyber Security Program Office (CSPO) must see the divisional progress and have the reports generated for us. We have the responsibility to report the vulnerability assessments of the entire laboratory to the Department of Energy (DOE), so we need global access into the Scavenger system.
    d. Provide auditing tools/graphs. This would complement our reporting features that the CSPO must provide to DOE.
    e. Keep track of vulnerability remediation/answers. In order to comply with DOE orders, we must keep track of who, when, and why vulnerabilities were purged from the Scavenger interface.
7. Use an Open Source Web Server. Apache is the choice for the CSPO office. We have supported and maintained Apache we servers with in house expertise on those servers, so leveraging the experience we have onsite was the best choice.

Prerequisites for Scavenger:

What Scavenger does not do is find the systems on the network to scan, in real time. That has to come from somewhere else. Depending on the size of the network, finding new hosts can be done in different ways. The network infrastructure and the type of work that Argonne does required Scavenger to have very targeted scans. In order to be real time, we could not scan the entire network space that Argonne owns as it is too large and not all of it is used. For a smaller rollout, scanning a subnet or two can be done very quickly and efficiently. For example, if you only have 255 or 500 hosts on a network, you could easily scan every host IP every 15-20 minutes looking for new hosts and vulnerabilities. Argonne owns three Class B networks, which means that it has 65534 * 3 or 196,602 hosts to scan. Instead of scanning the entire space, which could literally take days, we choose to leverage an existing Cisco technology of ARP tables. ARP, or Address Resolution Protocol, runs on all the Cisco switches. There is infrastructure that Cisco provides where you can copy these tables to a database with IP (Internet Protocol) Address, MAC (Media Access Control) address, time, and date information to derive active hosts, new hosts, and so forth.

The most critical dependency is the Nessus scanner. Prior to version 3, Nessus was open source. The most current version, including the first release of version 3, is closed source. However, the majority of the plug-ins that are used for the scans have remained open source. Nessus is a scanner that satisfies requirements set out by Microsoft and Cisco when choosing a vulnerability and penetration-testing platform [3, 4]. The scanner can scan for open ports, easily guessed passwords, database

misconfigurations, web site and application misconfiguration or coding issues, patch levels on multiple platforms, known vulnerabilities, Trojan horses, and certain type of virus infections. Nessus can be considered the core of the Scavenger system. Nessus has proven time and again that it, as an open source project, detects more vulnerabilities than any comparable commercial vulnerability scanner [11].

Scavenger also depends on LAMP technology. LAMP stands for Linux, Apache, MySql, and PHP or Perl. LAMP is the "technological glue" used to integrate the Nessus scanner with other open source technologies to make Scavenger a viable solution to the vulnerability and penetration testing and tracking problem. To keep Scavenger open source, we decided to use open source technology. Not only does this make the integration easier, it uses technology that can be readily modified for our use. Since we are contemplating releasing Scavenger to the open source community, basing it on open source technologies was just a natural fit.

Scavenger Setup at Argonne National Lab:

Scavenger is setup to do a multitude of scans. There are two nessus configurations that Scavenger uses. The full scan consists of every plug-in that nessus is capable of doing minus the DOS (Denial of Service) scans. The scan does not include the denial-of-service plug-ins because the CSPO does not want to disrupt systems. The DOS plug-ins have been tested to cause certain systems to crash and become unresponsive, and that is not our goal. Also, for the full scans, we turn on the safety feature which allows nessus to throttle a scan so it does not disrupt normal services on the servers and/or clients. The other type of scan that Scavenger uses through nessus is what the Cyber Security Program Office, CSPO, at Argonne likes to call the Low Hanging Fruit Scan or LHF. This is a targeted list of vulnerability checks where there is known exploit code out in the wild. Essentially, LHF is our high-risk vulnerabilities that could allow local or remote root compromises with little or no effort. The CSPO uses multiple information feeds to generate this list. For example, we use the external source like the SysAdmin, Audit, Network, Security or SANS Institute top twenty list [2]. We also use internal information sources to the Department of Energy. This list is reviewed on a regular basis, based on patch releases and watching bug-track lists for major applications used at the laboratory. This list is also shared throughout the Department of Energy National Laboratory Complex for review. This list does vary slightly from lab to lab, based on each lab's installed application base. This list of LHF will also be available once the project goes open source. A sample of the Low Hanging Fruit is shown in Figure 1.

Nessus Plugin ID:	Description	Date Added
22194	MS06-040 Vulnerability	2006-08-10 09:55:00
21696	MS06-025 RAS Vulnerability	2006-06-22 10:54:00
21327	Retrospect Client Buffer Overflow	2006-06-07 13:56:00
10660	Oracle tnslsnr security	2006-06-02 16:25:00
21564	RealVNC Authentication Bypass Vulnerability	2006-05-15 17:00:00
19427	Veritas default root	2006-04-17 14:18:00
12011	BetterInternet Detected	2006-04-13 11:32:23
20368	MS04-042	2006-04-13 11:32:23
20006	MS05-046	2006-04-13 11:32:23
20182	Veritas NetBackup Vuln	2006-04-13 11:32:23
15970	MS04-045	2006-04-13 11:32:23
12055	MS04-007	2006-04-13 11:32:23
12014	Free Community Detected	2006-04-13 11:32:23
19407	MS05-043	2006-04-13 11:32:23
20728	MSDE Weak sa P/W	2006-04-13 11:32:23
12015	IPInsight Detected	2006-04-13 11:32:23
20008	MS05-051	2006-04-13 11:32:23
16230	Veritas BO	2006-04-13 11:32:23
12012	CyDoor Running	2006-04-13 11:32:23
11995	Bonzi Buddy Running	2006-04-13 11:32:23
13852	MS Scheduler Vuln	2006-04-13 11:32:23
19948	Open X server	2006-04-13 11:32:23
10673	MSSQL sa blank pw	2006-04-13 11:32:23
12010	BargainBuddy Running	2006-04-13 11:32:23
11026	AP Detection	2006-04-13 11:32:23
18502	MS05-027	2006-04-13 11:32:23
11883	Gator Running	2006-04-13 11:32:23
11998	Gator Running	2006-04-13 11:32:23
11996	Brilliant Digital Running	2006-04-13 11:32:23
12063	Bagle.B Detected	2006-04-13 11:32:23
19408	MS05-039	2006-04-13 11:32:23
18027	MS05-017	2006-04-13 11:32:23
16337	MS05-007	2006-04-13 11:32:23

Figure 1: Sample LHF database

Additionally, Argonne not only has two levels of scanning, but it also has multiple frequencies of scans. Also, because of the shear size of the ANL network, 3 class B subnets, the Scavenger application has the capability of having multiple scanning nodes feed a single reporting repository. The scanning nodes have been placed where scanning traffic going over an internal firewall layer may impact performance. So instead of scanning from outside a certain division, we place a scanner inside the divisional network firewall. This way we can send the results file back to the central repository instead of passing the entire scan traffic across their networks. For example, Scavenger has two main scanner resources, one inside the firewall and one outside the firewall. This allows us to perform what we like to call a pre- and post- conduit Scavenger scan. Then, the main inside firewall scanner is broken down into smaller divisional scanners for divisions that have more than 2,500 hosts. Argonne has an online form for system and network administrators who are requesting services that will be available through the firewall. The requests go through a formal process, which includes a Scavenger “health” check.

For example, let's say an administrator needs a new web server to be allowed out to the Internet. They go to the online form and fill out the necessary information for the request for a conduit in the firewall. Once they submit the form Scavenger does a full scan from inside the firewall and generates a report that is reviewed by the Cyber Office, Networking, and other Managers to verify that the services requested are configured in a secure manner. An example of this pre-conduit scan report is given in Figure 2.

Scavenger – FW Pre-Conduit Scan

Scavenger <scavenger@anl.gov>

To: cspo@anl.gov

Nessus Scan Report
This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to eradicate these threats.

Scan Details	
Hosts which were alive and responding during test	1
Number of security holes found	0
Number of security warnings found	1

Host List	
Host(s)	**Possible Issue**
[redacted]7.2.75	Security warning(s) found

[return to top]

Analysis of Host		
Address of Host	**Port/Service**	**Issue regarding Port**
[redacted]7.2.75	ssh (22/tcp)	Security notes found
[redacted]7.2.75	general/tcp	Security notes found
[redacted]7.2.75	sometimes-rpc14 (32775/udp)	Security notes found
[redacted]7.2.75	unknown (899/udp)	Security notes found
[redacted]7.2.75	sunrpc (111/udp)	Security notes found
[redacted]7.2.75	unknown (890/tcp)	Security notes found
[redacted]7.2.75	iclcnet_svinfo (887/udp)	Security warning(s) found
[redacted]7.2.75	general/udp	Security notes found
[redacted]7.2.75	general/icmp	Security notes found
[redacted]7.2.75	sometimes-rpc9 (32773/tcp)	Security notes found
[redacted]7.2.75	ideafarm-chat (902/tcp)	Security notes found
[redacted]7.2.75	sunrpc (111/tcp)	Security notes found
[redacted]7.2.75	http (80/tcp)	Security notes found
[redacted]7.2.75	smtp (25/tcp)	Security notes found

Figure 2: Pre-conduit scan

After the report is reviewed and all is found to be acceptable, the conduit for the web server is entered into the firewall. This is done through another process in the Networking Group. Once the process has been completed, Scavenger is automatically notified that a new system has been added to pass traffic through the firewall. What happens next is that the same full Nessus scan from Scavenger is done but from *outside* the firewall. This will give the reviewing committee a post-conduit installation view of

the host. It will verify that the conduit in the firewall was entered correctly and that only the requested services are seen. An example of a post-conduit scan is shown in Figure 3:

Scavenger – FW Post-Conduit Scan

Scavenger <scavenger@anl.gov>

To: cspo@anl.gov

Nessus Scan Report

This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to eradicate these threats.

Scan Details

Hosts which were alive and responding during test	1
Number of security holes found	0
Number of security warnings found	0

Host List

Host(s)	Possible Issue
[redacted].2.75	Security note(s) found

[return to top]

Analysis of Host

Address of Host	Port/Service	Issue regarding Port
[redacted]2.75	general/tcp	Security notes found
[redacted].2.75	general/icmp	Security notes found

Security Issues and Fixes: [redacted].2.75

Type	Port	Issue and Fix
Informational	general/tcp	[redacted]2.75 resolves as [redacted]anl.gov. Nessus ID : 12053
Informational	general/tcp	Information about this scan : Nessus version : Unknown (NASL_LEVEL=2202) Plugin feed version : 200608301215 Type of plugin feed : Direct Scanner IP : [redacted]186.4 Port range : default Thorough tests : no Experimental tests : no Paranoia level : 1

Figure 3: Post-conduit scan

Once the internal and external views are reviewed and verified, the system is allowed to have a conduit through the firewall, allowing web services to be accessible by the Internet community.

Another type of scan that Argonne has in the Scavenger system is referred to as the Bi-Weekly Firewall scans. This scan type is done from outside the firewall, like the post-conduit request scans, but scans all firewall conduit-enabled machines. The nessus scanner uses the Full vulnerability setup as described earlier. This verifies on a bi-weekly schedule that Internet accessible servers maintain a secure configuration.

The next type of scan is what is called the 15-minute scan. This scan is fed from the ARP tables that were described earlier in the text. What is generated from the ARP database is a list of hosts that have never been seen before as a MAC/IP combination.

Every fifteen minutes, that list of hosts is scanned for LHF. This list consists of new hosts that have never been seen by Scavenger before.

One of the final types of scans is the 24-hour scan. This scan is also generated from the ARP database. All the new systems that were seen in the previous 24 hours AND systems that generated at least one packet of network traffic on any switch on the network will get scanned at least once every 24 hours for LHF. This catches systems that Scavenger has seen before, but does not fall into the 15-minute scans.

Scavenger also does VPN scans. This allows Argonne to protect and assess systems that are connecting to the VPN, which may consist of ANL configured machines and/or home machines that may be grossly out of date. Scavenger is linked up to the VPN concentrator that Argonne maintains. In real-time, once a user connects to the VPN and is assigned an internal ANL IP address, it is scanned for LHF by Scavenger.

The final scan type for Scavenger is the Manual Scan. This was a feature request from the network administrators at ANL. A feature was added to Scavenger to allow a system or network administrator to perform a nessus scan without having to maintain his or her own local nessus scanner. This also allows reports and status of the scan to be kept track of by the Scavenger system.

These are the types of Scans that Argonne uses. These scan frequencies are not hard-coded into Scavenger, so the configuration, frequency and depth of the scans are completely configurable to serve networks of various topologies and sizes.

Scavenger Inner Workings:

Now that you have an idea of how Scavenger works, how does Scavenger *really* work? Here is what one will need to run Scavenger.

First, a LINUX distribution of your favorite flavor of LINUX is required. Hardware requirements will be based on how many nessus scanners you will want and if you want to keep your nessus scanners separate from the Scavenger code and Database. The beauty of this is that Scavenger is very scalable based on the LAMP technology chosen. At the very least, you will need one box dedicated to Scavenger. Scavenger then will need either a local copy of nessus installed and/or remote copies of nessus installed if you wish to have internal and external scanner views relative to a firewall. This piece is also very scalable: if you have large network space to scan, you can have multiple instances of nessus installed on machines scattered throughout the network to decrease scan times for large networks.

Next, you will need a MySQL Database server. This, again, could be local to one system, or you could use a MySQL Database server that is remote. The database is needed to store the nessus results and provide the reporting features to the administrators of the systems and network. The nessus scans, after they are complete, get fed into the database for reporting, instance counts, and remediation tracking. The database, as with any application, is the repository for the data to be used. The Scavenger database ERD Diagram is shown in Figure 4.

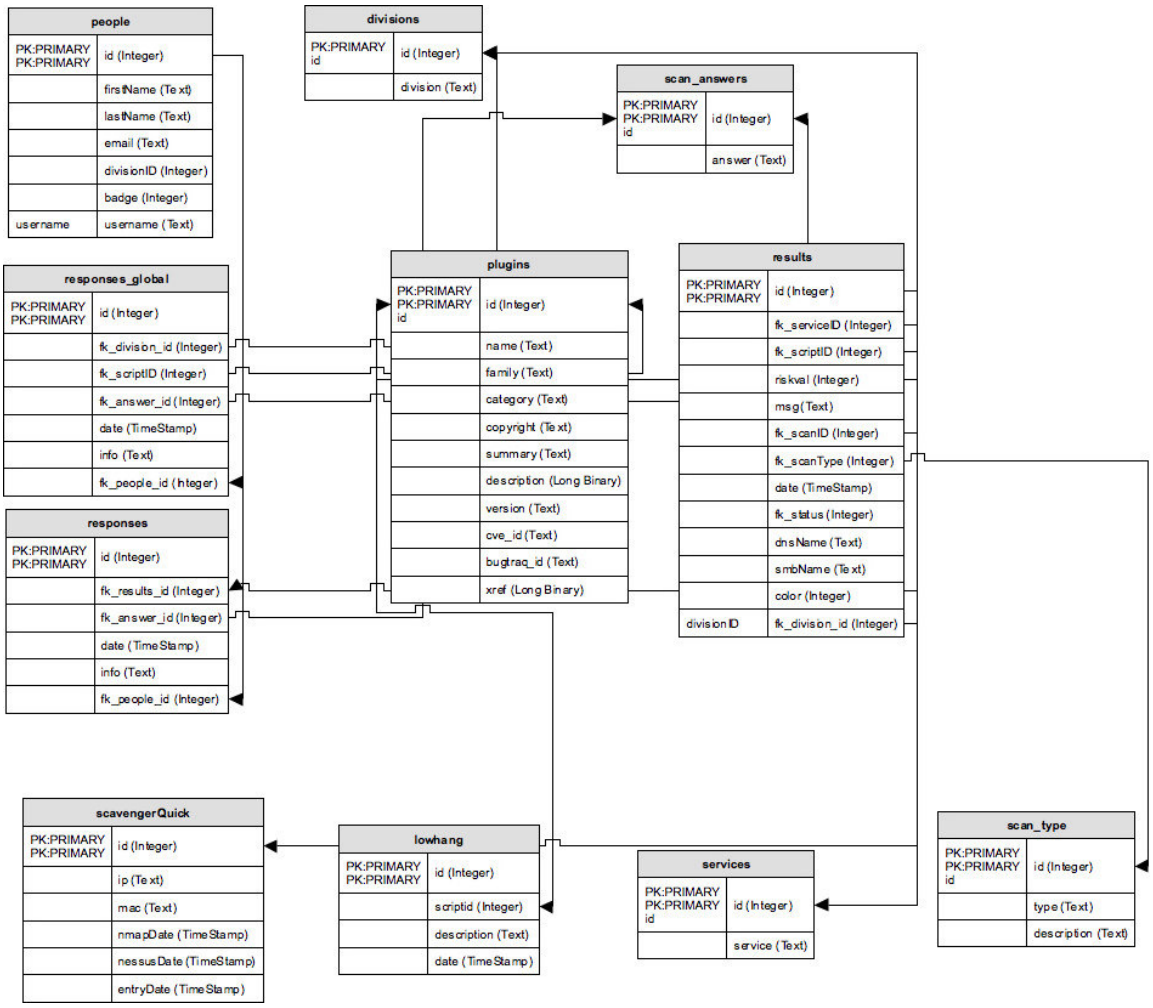

Figure 4: Scavenger database schema

Next, are the pieces needed for main code of Scavenger. This is the Apache web server, PHP (PHP: Hypertext Preprocessor), Perl, and associated MySQL modules for PHP and Perl to allow these scripting languages to create connections back to the MySQL database.

An optional piece is the infrastructure of getting real time hosts to Scavenger. This is really up to a Networking group. If you have a small set of hosts, Scavenger could be configured to scan all hosts on a regular basis. However, if you have a large set of IPs to scan, you may want to look at Cisco technologies that can provide Scavenger with targeted scans for active and new hosts that show up on the network.

Scavenger does not have authentication for the front-end built in. What is recommended is to use the Apache Lightweight Directory Access Protocol (LDAP) plug-in authentication module and restrict access to the main folder of the Scavenger front end by an LDAP group and then use the LDAP server for authentication, like Open LDAP or Microsoft ® Active Directory, for example.

The frequency of the nessus scans for Scavenger is simply controlled by a cron job on the LINUX server. Cron is the UNIX application that controls the scheduling of

automated scripts and tasks. So modifying the cron job will modify the frequency at which Scavenger runs its scheduled scans.

Once the Scavenger scans have taken place and are uploaded into the database, it is now up to the system and network administrators to address the vulnerabilities that were found. This is done by automatic notification to the administrators via email. Once the administrators have received a notification by Scavenger to respond to a vulnerability, it is up to the administrator to log into the front-end interface to address the vulnerability. Scavenger does not fix vulnerabilities, but it is a vulnerability tracking system, so the administrator will look at what vulnerabilities were found and provide an appropriate answer. Possible responses the Administrator can make include Accept, False Positive and Addressed. There are three possible outcomes.

1. Accept: The administrator accepts the vulnerability and the risk. For example, if a web server is identified, there is an inherent risk to running web server. If the administrator accepts that risk, he or she fills out a risk assessment form, which is part of Scavenger, and a record is kept [9].
2. False Positive: The administrator deemed the check as being inaccurate and the service is not running on the machine or there was an error with the scan result.
3. Addressed: The administrator fixed the vulnerability, applied a patch, or provided a layer of protection to address the risk involved.

The typical protocol Scavenger follows is summarized here:

1. Read in the list of IPs to be scanned. This can either be hard coded or fed from another technology like Cisco ARP table lists.
2. Feed the IP list to the proper Scavenger script to feed to Nessus
3. Scavenger will run Nessus in the correct mode and from the correct location based on the type of scan
    a. LHF vs. Full Scan
    b. Inside vs. Outside Firewall
4. Scavenger will import the raw results of the scan into the MySQL database
5. Post-process the results:
    a. If the vulnerability has been seen before, update the instance count of the original vulnerability.
    b. If the vulnerability has been marked as Accept, update the “last seen” date of the accepted record. Last seen refers to when Scavenger last saw a vulnerability: either new, accepted, or still outstanding.
    c. If the vulnerability had been deemed as a false positive, throw the update out.
    d. If the vulnerability has a global auto-answer, auto-answer the vulnerability. An example of a global auto-answer is if the divisional administrator had a public Simple Network Management Protocol (SNMP) string called “public” assigned to all his printers. Administrative programs to query systems for certain types of information use the SNMP protocol. Now, if the administrator does not want Scavenger to flag all his/her printers as having an easily guessable public SNMP string, the administrator can globally answer all the SNMP vulnerabilities for the printers on their network. So even if new printers are configured and

found by Scavenger, if the global auto-answer is enabled for the vulnerability, Scavenger will auto-answer on that administrator's behalf. This will continue to occur for all types of scans until the divisional administrator decides to turn off that particular global auto-answer.

e. If the vulnerability has been marked addressed, unmark addressed and put the vulnerability back in unanswered mode.

6. Send email to administrators with unanswered vulnerabilities based on the configured time line of the notification
    a. 15-minute
    b. Bi-weekly Firewall
    c. Daily Summaries
    d. Other timed configurations based on the organization
7. Wait for answers from administrators
8. Do Daily Reporting to administrators and Cyber Office
    a. Daily Global Summary Overview
    b. Graphing output and trends
    c. Other administrative notifications and graphs based on the organizations need

Scavenger Reporting Features:

One of Scavengers main features is the reports that it can generate. Microsoft states that a vulnerability and penetration system must be able to display the results in a manner that matches the risks assigned by the organization [4]. Since Scavenger is a vulnerability tracking system, reports are very important. Scavenger satisfies the requirements and planning suggested by Microsoft and Cisco for a penetration and vulnerability testing application [3, 4]. There are a slew of reporting features in Scavenger. The reports essentially fall into two groups. Administrative reports about the overall heath of the laboratory, which is sent only to the CSPO office, and reports that are given to the individual system and network administrators of vulnerabilities found in their respective divisions. The reports are logically broken up into the same categories as the scan types. The reports that local system administrators see are:

1. 15-minutes Scan reports, which are sent every 15 minutes until the vulnerability is addressed in the Scavenger system. See Figure 5 for an example.

Cyber Security Program Office
Cyber Security Program Office
Continuous Scanning Project - Scavenger
Type of Scan: Inside FW - Every 15 Minutes

The following list of vulnerabilities were identified by the Cyber Security continuous scanning project for hosts that your are responsible for. These vulnerabilities identified are of a high risk to your systems and the Laboratory. Please address these vulnerabilities as soon as possible and record your actions in the online tracking system located at this site.

Host	DNS	SMB	Last Seen Vulnerable	Vulnerability
[redacted].38.14	[redacted].anl.gov		2006-09-07 10:30:01	Vulnerability in Server Service Could Allow Remote Code Execution (921883) - Network check

The following email was generated by the CSPO system Scavenger. If you have any questions regarding this email or the system, please contact scavenger@anl.gov or via phone at 2-3456.

Figure 5: Example of a 15-minute report

2. Daily Summary Scan Reports, which are sent out every 24 hours. See Figure 6 for an example.

Cyber Security Program Office
Cyber Security Program Office
Continuous Scanning Project - Scavenger
Type of Scan: Summary Vulnerability Report (Contains Outside FW, Inside FW and Daily Scan Results)

The following list of vulnerabilities were identified by the Cyber Security continuous scanning project for hosts that your are responsible for. These vulnerabilities identified are of a high risk to your systems and the Laboratory. Please address these vulnerabilities as soon as possible and record your actions in the online tracking system located at this site.

Type	Host	DNS	SMB	Last Seen Vulnerable	Vulnerability
Inside FW - Daily	[redacted].241.204	[redacted].anl.gov		2006-09-06 00:30:01	Vulnerability in Server Service Could Allow Remote Code Execution (921883) - Network check
Inside FW - Daily	[redacted].150.173	[redacted].anl.gov		2006-09-06 00:30:01	Vulnerability in Server Service Could Allow Remote Code Execution (921883) - Network check
Inside FW - Daily	[redacted].244.31	[redacted].anl.gov		2006-09-06 00:30:01	Vulnerability in Server Service Could Allow Remote Code Execution (921883) - Network check
Inside FW - Daily	[redacted].245.12	[redacted].anl.gov		2006-09-06 00:30:01	Vulnerability in Server Service Could Allow Remote Code Execution (921883) - Network check
Inside FW - Daily	[redacted].240.7	[redacted].anl.gov		2006-09-05 00:30:01	Open X11 Server
Inside FW - Daily	[redacted].244.169	[redacted].anl.gov		2006-09-06 00:30:01	Vulnerability in Server Service Could Allow Remote Code Execution (921883) - Network check

The following email was generated by the CSPO system Scavenger. If you have any questions regarding this email or the system, please contact scavenger@anl.gov or via phone at 2-3456.

Figure 6: Example of a 24-hour report

3. Bi-Weekly Firewall Scan Reports, which are sent out twice every week. See Figure 7 for an example.

Cyber Security Program Office
Cyber Security Program Office
Continuous Scanning Project - Scavenger
Type of Scan: Outside FW - Bi-Weekly

The following list of vulnerabilities were identified by the Cyber Security continuous scanning project for hosts that your are responsible for. These vulnerabilities identified are of a high risk to your systems and the Laboratory. Please address these vulnerabilities as soon as possible and record your actions in the online tracking system located at this site.

Host	DNS	SMB	Last Seen Vulnerable	Vulnerability
.228.119	.anl.gov		2006-05-12 08:46:39	Deprecated SSL Protocol Usage
228.19	.anl.gov		2006-05-12 08:46:39	Deprecated SSL Protocol Usage
228.3	.anl.gov		2006-09-01 09:29:36	Portable OpenSSH PAM timing attack
.228.12	.anl.gov		2006-08-30 00:17:01	Portable OpenSSH PAM timing attack
.228.14	.anl.gov		2006-08-30 00:17:01	Portable OpenSSH PAM timing attack
228.7	.anl.gov		2006-08-30 00:17:01	Portable OpenSSH PAM timing attack
.228.78	.anl.gov		2006-09-04 09:29:27	Portable OpenSSH PAM timing attack
.228.13	.anl.gov		2006-09-04 09:29:27	Portable OpenSSH PAM timing attack
.228.252	.anl.gov		2006-09-04 09:29:27	Mailman Detection
228.252	.anl.gov		2006-09-04 09:29:27	Directory Scanner

The following email was generated by the CSPO system Scavenger. If you have any questions regarding this email or the system, please contact scavenger@anl.gov or via phone at 2-3456.

Figure 7: Example of a bi-weekly firewall scan report

4. VPN Scan Reports, which are sent out in real-time if a user with a vulnerable system connects to the VPN. This report is a little special as it is sent to the end user in addition to the network administrator who is responsible for that VPN network. See Figure 8 for an example.

Michael,

During your recent connection to the Argonne VPN, your connecting computer was found to have an open vulnerability that leaves your computer system and the Laboratory at risk. It is highly recommend that you address this vulnerability in a timely manner.

If you are running a windows machine, please verify that your patches are up-to-date AND you have Windows Update turned on to automatically download and install patches. Guidance for this can be found at Microsoft's site: http://www.microsoft.com/athome/security/protect/chooseos.mspx

If you have further questions, please contact your CSPR or the Argonne Cyber Security Program Office at 2-3456 or cyber@anl.gov if you need further help in securing your system.

Host	DNS	Username	Last Seen Vulnerable	Vulnerability	Remediation Help
.234.84	.anl.gov	mo i	2006-10-25 15:45:01	Vulnerability in Server Service Could Allow Remote Code Execution (921883) - Network check	Nessus Remediation Information

The following email was generated by the CSPO system Scavenger. If you have any questions regarding this email or the system, please contact scavenger@anl.gov or via phone at 2-3456.

Figure 8: Example of a VPN report

There is also a set of reports designed to provide Administrators an overview of the health of the system. The administrative reports include the following:

1. The daily LHF Report shows counts of open vulnerabilities based on the type of scan which is sent to the CSPO to get an overall health of the laboratory. See Figure 9 for an example

Previous Days Discovered Vuln Counts
==========

Quick 15 Min. Scan
++++++++++++++++++
Discovered Yesterday (2006-09-06) = 1
Of Discovered Yesterday, How Many Closed = 1
Total Remain Open Total = 0
Remain Open Total > 3 Days = 0
Remain Open Total > 5 Days = 0

Full/Daily Inside Firewall Scan
++++++++++++++++++++++++++
Discovered Yesterday (2006-09-06) = 25
Of Discovered Yesterday, How Many Closed = 4
Total Remain Open Total = 25
Remain Open Total > 3 Days = 23
Remain Open Total > 5 Days = 2

Semi Annual Inside Firewall Scan
++++++++++++++++++++++++++
Discovered Yesterday (2006-09-06) = 0
Of Discovered Yesterday, How Many Closed = 0
Total Remain Open Total = 34
Remain Open Total > 3 Days = 0
Remain Open Total > 5 Days = 34

Outside Firewall Scan
++++++++++++++++++
Discovered Yesterday (2006-09-06) = 0
Of Discovered Yesterday, How Many Closed = 0
Total Remain Open Total = 12
Remain Open Total > 3 Days = 4
Remain Open Total > 5 Days = 8

Total OPEN LHF sorted by description
++++++++++++++++++++++++++++++++

Host	DNS	Scan Type	Last Seen Vulnerable	Description
[redacted]198.223	[redacted].anl.gov	Semi-Annual	2006-08-10 13:46:00	MS06-040 Vulnerability
[redacted]198.103	[redacted].anl.gov	Scavenger Daily	2006-09-06 00:30:01	MS06-040 Vulnerability

Figure 9: Daily LHF report

2. A set of graphs, which are generated on-demand with the newest numbers based on the time of the web request. One of these graphs is the total number of vulnerabilities experienced each day. See Figure 10 for an example.

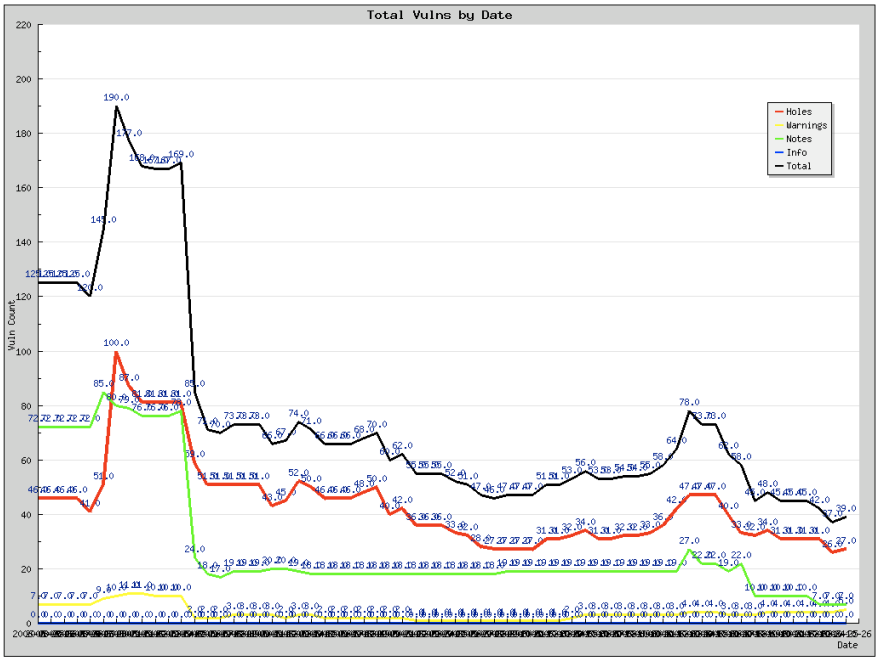

Figure 10: Daily vulnerabilities chart

3. Another graph depicts how well certain divisions are responding to vulnerabilities. The graph breaks down the vulnerabilities based on division. See Figure 11 for an example.

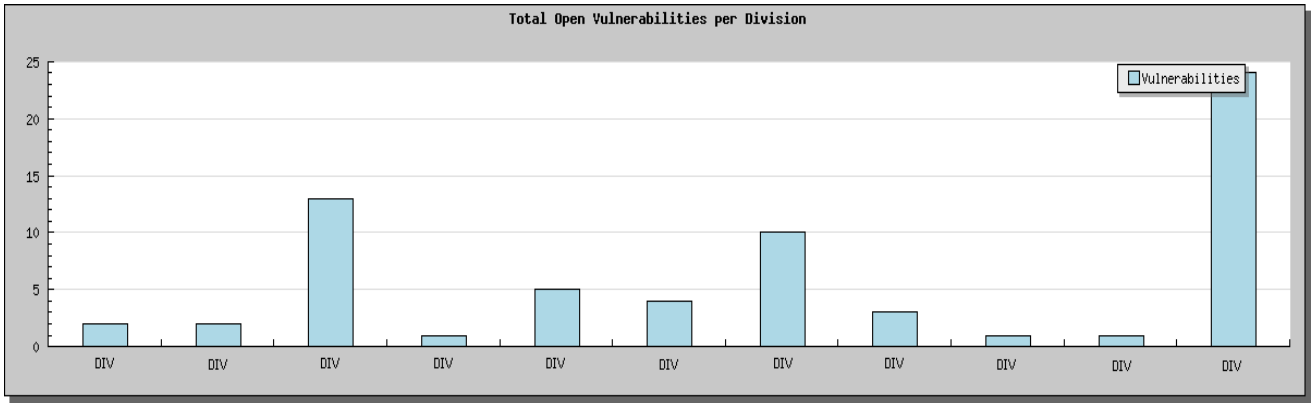

Figure 11: Chart showing response by division

4. Lastly is the VPN graph, which shows how many connections out of the total had LHF vulnerability. See Figure 12 for an example.

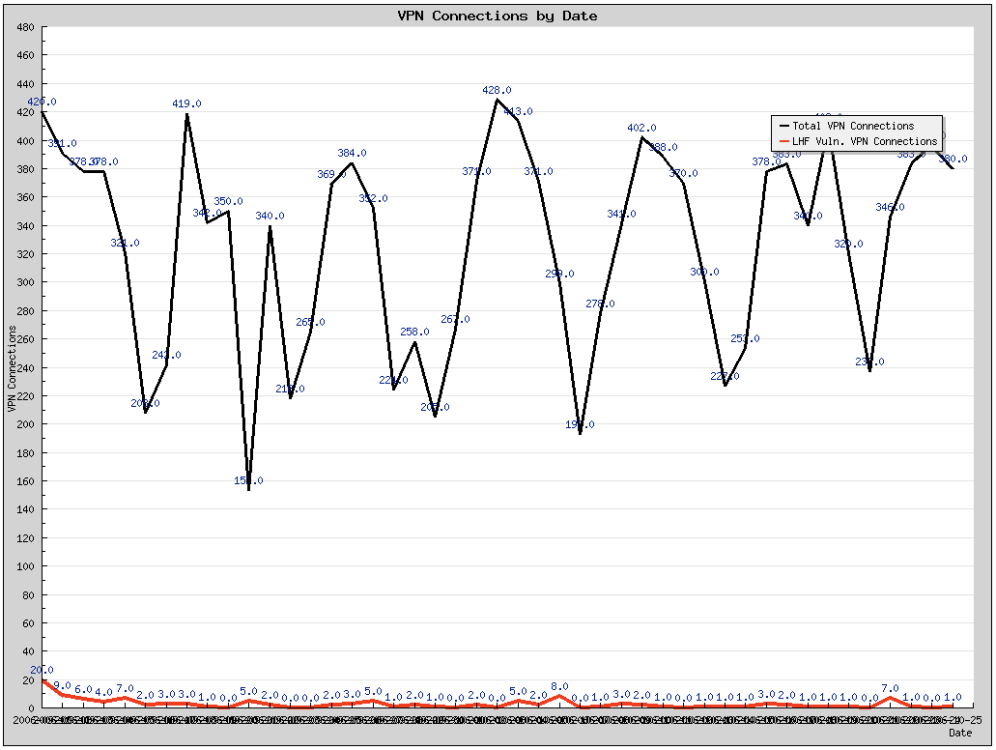

Figure 12: VPN vulnerability chart

These are the basic reporting features that are built into Scavenger. The beauty of this is that custom reports and graphs can be created with just a little knowledge of SQL queries. The frequency of the how often email reports are sent is based on a cron job. Each email report I highlighted is a separate process, so they can be sent out at intervals that fit one's organization. So if notifying an administrator every fifteen minutes seems too extreme for a LHF vulnerability, one can change the cronjob to notify every hour or as they see fit.

Future of Scavenger:

The future of Scavenger seems very promising. We have begun the process of making Scavenger more portable. This way, it will be easier to package and install in a new site and network infrastructure. We have also begun a campaign to start distributing Scavenger to other national laboratories in the DOE complex. Furthermore, we have begun using the Subversion package to keep track of development changes. Subversion is an open source package for keeping track of development projects and code. In addition to making Scavenger more portable and easier to install, we are migrating the main back-end code from the PHP interpreter to the Perl interpreter. Perl has a couple advantages over PHP. The first advantage is that the requirement for having Apache installed with the PHP modules will no longer be required for the back-end code. This allows the configuration of the scanning nodes to have less software being installed so that the attack 'footprint' on the vulnerability scanning system is less. Also Perl has less computing overhead than the command line PHP interpreter. PHP runs through the Apache web server process, which is an inefficient way of executing code that is not

viewed on a browser. Another major upgrade is to rewrite all the SQL queries to support the views that are available through MYSQL 5. This will decrease the query time for large select statements that we join multiple tables together.

A security upgrade to the Scavenger project will be how Scavenger sends out email. Email notification is a major component. To make this feature more reliable, we intend to include non-repudiation and encryption to the email. We are currently working on giving the option of creating Pretty Good Privacy (PGP) key sets to digitally sign and encrypt the vulnerability notifications to prevent email spoofing and to hide vulnerabilities from unauthorized access through the email system.

We are hoping to make this code completely open source in the near future. This largely depends on the time dedicated to the project and priorities of Argonne and the Argonne Cyber Security Office.

Works Cited:

[1] (2006, July 14). CSI/FBI Computer Crime and Security Survey. Retrieved October 15th, 2006, from the Computer Security Institute Website: http://www.gocsi.com/forms/fbi/csi_fbi_survey.jhtml;jsessionid=2QHNTL4HTG440QSNDLPCKHSCJUNN2JVN

[2] The Top 20 Most Critical Internet Security Vulnerabilities (Updated) - The Experts Consensus. Retrieved October 15th, 2006, from The SANS Institute Website: http://www.sans.org/top20/

[3] Andrew Whitaker, Daniel Newman (2006, November) Penetration Testing and Network Defense. Indianapolis, IN: Cisco Systems Inc.

[4] Kevin Lam, David LeBlanc, and Ben Smith (2004) Assessing Network Security. Redmond, WA: Microsoft Press

[5] Compliance Checks Frequently Asked Questions (FAQ). Retrieved October 15th, 2006, from the Nessus.org Website: http://www.nessus.org/documentation/index.php?doc=compliance

[6] Introduction to Vulnerability Scanning. Retrieved October 15th, 2006, from the Net Security Website: http://netsecurity.about.com/cs/hackertools/a/aa030404.htm

[7] vulnerability scanner. (n.d.). Wikipedia, the free encyclopedia. Retrieved November 12th, 2006, from Reference.com website: http://www.reference.com/browse/wiki/Vulnerability_scanner

[8] Jordan Wiens (August, 4th, 2005) Vulnerability Assessment Scanner. Retrieved October 15th, 2006, from the Network Computing Website: http://www.networkcomputing.com/showitem.jhtml?docid=1615buyers

[9] Stephen Northcutt, Judy Novak (2002, September) Network Intrusion Detection: Third Edition. Indianapolis, IN: New Riders Publishing

[10] Tony Bradley. Introduction to Vulnerability Scanning. Retrieved Nov. 19th, 2006, from Networksecurity.com website: http://netsecurity.about.com/cs/hackertools/a/aa030404.htm

[11] Jeff Forristal, Greg Shipley (January 8th, 2001) Vulnerability Assessment Scanners. Retrieved Oct. 15th, 2006, from Network Computing Website: http://www.networkcomputing.com/1201/1201f1b1.html