White Paper

What is a vulnerability assessment?

Prepared on July 17, 2011 by:

Demyo Inc. is one hundred percent American owned and one hundred percent IT security oriented company with headquarters in Miami, Florida, USA.

Demyo Inc. delivers comprehensive penetration testing, vulnerability assessment, incident response, and compliance audit services just to name a few. Find out more at:

www.demyo.com

info@demyo.com

Tel. +1 201 665 6666

Miami, Florida, USA

Scope

Paper aims to provide basic concepts in regards of vulnerability assessments, reader should have some basic technical IT knowledge to understand them, however this paper does not aim to be an in depth technical reference.

What is it?

In computer security, the term vulnerability is applied to a weakness in a system, which allows an attacker to violate the integrity of that system. Vulnerabilities may result from weak passwords, software bugs, software misconfigurations, a computer virus or other malware (malicious software), a script code injection, or a SQL injection just to name the few.

A security risk is classified as vulnerability if it is recognized as a possible means of attack. A security risk with one or more known instances of working and fully implemented attacks is classified as an exploit. Constructs in programming languages that are difficult to use properly can be a large source of vulnerabilities.

Vulnerabilities existed all the time, but when Internet was at its early stage they were not as often used and exploited. Media did not report any news about hackers who are getting put in jail for "hacking" into servers and stealing vital information. Back then all nodes on the network were trusted, secure protocols such as SSH, SCP, SSL did not exist, but telnet, FTP and plain text HTTP were used to interexchange sensitive data. Who could think about sniffing? ARP poisoning? MITM?

Vulnerability assessment may be performed on many objects, not only computer systems/networks. For example physical buildings can be assessed so it would be clear what parts of the building have what kind of flaw. If the attacker **can** bypass the security guard at the front door and get into the building via back door it is definitely a vulnerability. If he actually **does** that – it is an exploit. The physical security is one of the most important aspects to be taken into the account. If the attackers have physical access to the server - the server is not yours anymore! Why? Because if the server is stolen, the attacker does not need to evade IDS, does not need to evade IPS, does no have to figure out the way on how to dump 10T of data, it is right here on the server. Full disk encryption would help, but it is not common use for servers. Make absolutely sure to do FDE (Full Disk Encryption) on all your laptops, also known as WDE (Whole Disk Encryption).

Just by stating ‘your systems/networks’ are vulnerable doesn't provide any useful information. Vulnerability assessment without a comprehensive report is pretty much useless. It is easy to use automatic tools to scan networks, make reports out of the tool and send it out, but that does not provide much value as report can easily run into thousand of pages. It is much better to make top 10 vulnerabilities out of all of them and make a report. Vulnerability assessment report should include:

- Identification of vulnerabilities and vulnerable systems

It is enough to find one critical vulnerability and the whole network is at risk, just like if one link is broken in the chain, and the whole chain is broken:

Figure 1: One critical vulnerability affect

Vulnerabilities should be sorted by severity and then by servers/services. Critical vulnerabilities should be on the top of the report and should be listed in descending order i.e. critical, then high, medium and low. [1]

Reporting

Reporting capability is of growing importance to administrators, in a documentation-oriented business climate where you must not only be able to do your job, but also provide written proof of how you've done it. In fact, respondents to Sunbelt's survey indicate that flexible and prioritizing reporting is their number one favorite feature.

A scan might return hundreds or thousands of results, but the data is useless unless it is organized in a way that it can be understood. That means that ideally you will to be able to sort and cross-reference the data, export it to other programs and formats (such as CSV, HTML, XML, MHT, MDB, Excel, Word, and/or Lotus), view it in different ways, and easily compare it to the results of earlier scans.

Comprehensive, flexible and customizable reporting is used within your department to provide a guideline of technical steps you need to take, but that's not all. Good reports also give you the ammunition you need to justify the costs of implementing security measures to management.

The "It Won't Happen to Us" Factor

Practical matters aside, CEOs, CIOs and administrators are all human beings and thus subject to normal human tendencies – including the tendency to assume that bad things happen to "other people," not to us. Organizational decision makers assume that their companies aren't likely targets for hackers ("Why would an attacker want to break into the network of Widgets, Inc. when they could go after the Department of Defense or Microsoft or someone else who's much more interesting?").

Why vulnerability assessment?

Organizations have a tremendous opportunity to use information technologies to increase their productivity. Securing information and communications systems will be a necessary factor in taking advantage of all this increased connectivity, speed and information. However, no security measure will guarantee a risk free environment in which to operate. In fact, many organizations will need to provide easier access by users to portions of their information systems, thereby increasing potential exposure. Administrative error, for example, is a primary cause of vulnerabilities that can be exploited by a novice hacker, whether an outsider or insider in the organization. Routine use of vulnerability assessment tools along with immediate response to problems identified will alleviate this risk. It follows, therefore, that routine vulnerability assessment should be a standard element of every organization's security policy. Vulnerability assessment is used to find out unknown problems in the systems. The main purpose of vulnerability assessment is to find out what systems have flaws and take action in order to mitigate the risk. Some industry standards such as DSS PCI require organizations to perform vulnerability assessments on their networks. Let's take a brief look of what is DSS PCI compliance:

DSS PCI compliance

PCI DSS stands for Payment Card Industry Data Security Standard. This standard was developed by leading credit card companies to help the merchants be secure and follow common security criteria in order to protect sensitive customers' credit card data. Before that every credit card company had a similar standard to protect customers' data on merchants' side. Any company that does transactions via credit cards needs to be PCI compliant. One of the requirements to be PCI compliant is to Regularly test security systems and processes. This can be achieved via vulnerability assessment. Small companies that don't process a lot of transactions are allowed to do self assessment via questionnaire. Big companies that process a lot of transactions are required to be audited by 3rd parties. [2]

Penetration Testing vs. Vulnerability Assessment

There seems to be a certain amount of confusion within the security industry about the difference between penetration testing and vulnerability assessment, they are often classified as the same thing when in fact they are not. Penetrations testing sounds a lot more exciting, but most people actually want a vulnerability assessment and not a penetration test; many projects are labeled as penetration tests when in fact they are 100% vulnerability assessments. A penetration test mainly consists of a vulnerability assessment, but it goes one step further. A penetration test is a method of evaluating the security of a computer system or network by simulating an attack by a malicious hacker. The process involves an active analysis of the system for any weaknesses, technical flaws or vulnerabilities. This analysis is carried out from the position of a potential attacker, and will involve active exploitation of security vulnerabilities. Any security issues that are found will be presented to the system owner together with an assessment of their impact and often with a proposal for mitigation or a technical solution. A vulnerability

assessment is what most companies generally do, as the systems they are testing are live production systems and can't afford to be disrupted by active exploits, which might crash the system. Vulnerability assessment is the process of identifying and quantifying vulnerabilities in a system. The system being studied could be a physical facility like a nuclear power plant, a computer system, or a larger system (for example the communications infrastructure or water infrastructure of a region). Vulnerability assessment has many things in common with risk assessment. Assessments are typically performed according to the following steps:

1. Cataloging assets and capabilities (resources) in a system

2. Assigning quantifiable value and importance to the resources

3. Identifying the vulnerabilities or potential threats to each resource

4. Mitigating or eliminating the most serious vulnerabilities for the most valuable resources

This is generally what a security company is contracted to do, from a technical perspective, not to actually penetrate the systems, but to assess and document the possible vulnerabilities and recommend mitigation measures and improvements. Vulnerability detection, mitigation, notification, and remediation are linked as follows [3]:

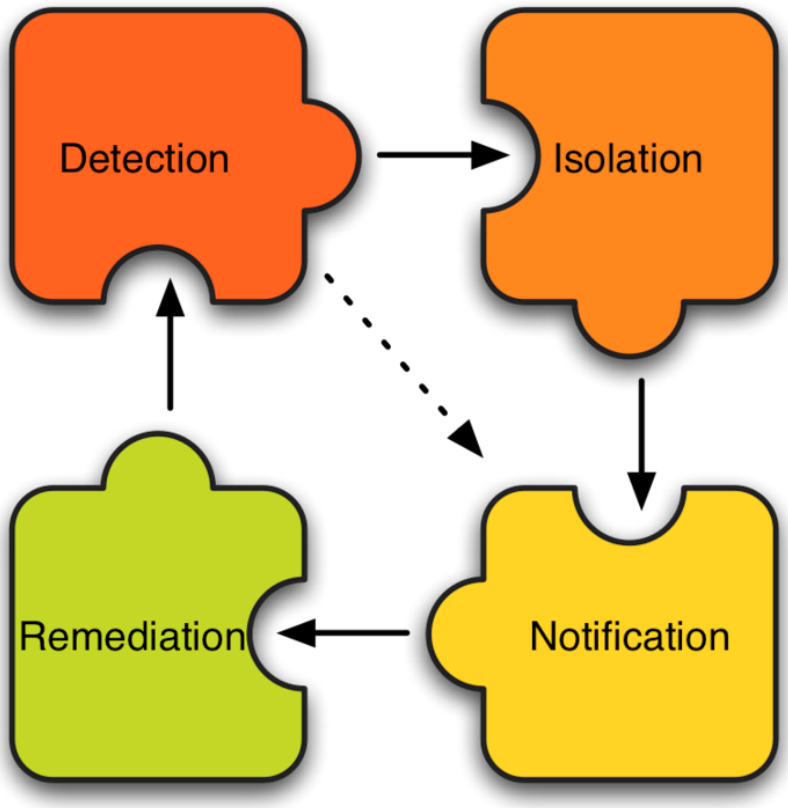

Figure 2: Vulnerability mitigation cycle

Vulnerability Assessment Goal

The theoretical goal of network scanning is elevated security on all systems or establishing a network wide minimal operation standard. The following diagram shows how usefulness is related to ubiquity:

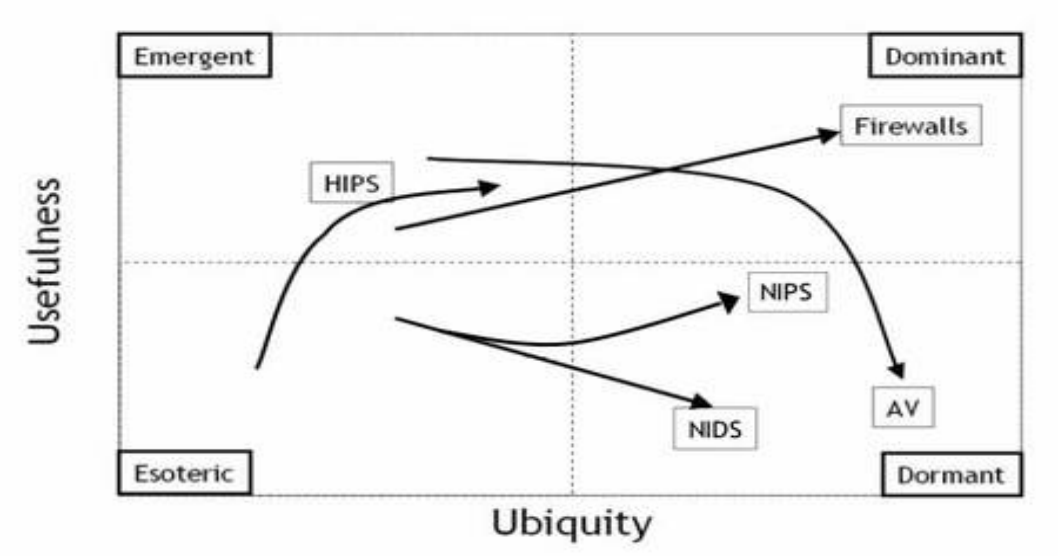

Figure 3: Usefulness - Ubiquity relation

HIPS – Host-Based Intrusion Prevention System

NIDS – Network-Based Intrusion Detection System

AV – Anti-Virus

NIPS – Network-Based Intrusion Prevention System

Mapping The Network

Before we start scanning the network we have to find out what machines are alive on the network. Most of the scanners have built in network mapping tool, usually it is nmap network mapping tool running behind the scenes. The Nmap Security Scanner is a free and open source utility used by millions of people for network discovery, administration, inventory, and security auditing. Nmap uses raw IP packets in novel ways to determine what hosts are available on a network, what services (application name and version) those hosts are offering, what operating systems they are running, what type of packet filters or firewalls are in use, and more. Linux Journal and Info World named Nmap "Information Security Product of the Year". Hackers in the movies Matrix Reloaded, Die Hard 4, and Bourne Ultimatum also used it. Nmap runs on all major computer operating systems, plus the Amiga. Nmap has a traditional command-line interface:

```
sh-3.2# nmap -sV scanme.nmap.org

Starting Nmap 5.59BETA1 ( http://nmap.org ) at 2011-07-15 13:49 EDT
Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.081s latency).
rDNS record for 74.207.244.221: li86-221.members.linode.com
Not shown: 992 closed ports
PORT      STATE    SERVICE      VERSION
22/tcp    open     ssh          OpenSSH 5.3p1 Debian 3ubuntu6 (protocol 2.0)
80/tcp    open     http         Apache httpd 2.2.14 ((Ubuntu))
111/tcp   filtered rpcbind
1720/tcp  filtered H.323/Q.931
2000/tcp  filtered cisco-sccp
5060/tcp  filtered sip
9929/tcp  open     nping-echo   Nping echo
31337/tcp open     tcpwrapped
Service Info: OS: Linux

Service detection performed. Please report any incorrect results at http://n
Nmap done: 1 IP address (1 host up) scanned in 9.17 seconds
```

Figure 4: Nmap command line interface

Zenmap is the official Nmap Security Scanner GUI. It is a multi-platform (Linux, Windows, Mac OS X, BSD, etc.) free and open source application, which aims to make Nmap easy for beginners to use while providing, advanced features for experienced Nmap users. Frequently used scans can be saved as profiles to make them easy to run repeatedly. A command creator allows interactive creation of Nmap command lines. Scan results can be saved and viewed later. Saved scan results can be compared with one another to see how they differ. The results of recent scans are stored in a searchable database.

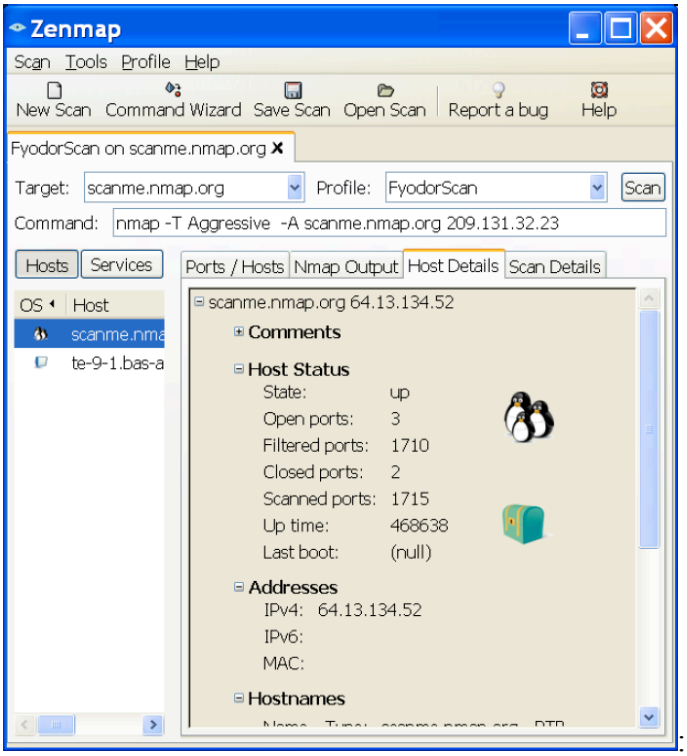

Figure 5: Zenmap graphical user interface

Gordon Lyon (better known by his nickname Fyodor) released Nmap in 1997 and continues to coordinate its development. He also maintains the Insecure.Org, Nmap.Org, SecLists.Org, and SecTools.Org security resource sites and has written seminal papers on OS detection and stealth port scanning. He is a founding member of the Honeynet project and co-authored the books "Know Your Enemy: Honeynets" and "Stealing the network: How to Own a Continent". Gordon is President of Computer Professionals for Social Responsibility (CPSR), which has promoted free speech, security, and privacy since 1981. [4][5]

Some systems might be disconnected from the network. Obviously if the system is not connected to any network at all it will have a lower priority for scanning. However it shouldn't be left in the dark and not being scanned at all, because there might be other non-network related flaws, for example firewire exploit can be used to unlock the Windows XP SP2 system. Exploit works like this: attacker approaches locked Windows XP SP2 station, plugs firewire cable into it, and uses special commands to unlock the locked machine. This technique is possible because firewire has direct access to RAM. The system will accept any password and unlock the computer. [6]

Selecting The Right Scanners

Scanners alone don't solve the problem, scanning should be used only as starting point in vulnerability assessment. Start with one scanner but consider more than one. It is a good practice to use more than one scanner. This way you can compare result from a couple of them. Some scanners are more focused on particular services and/or services. The typical scanner architecture is shown below:

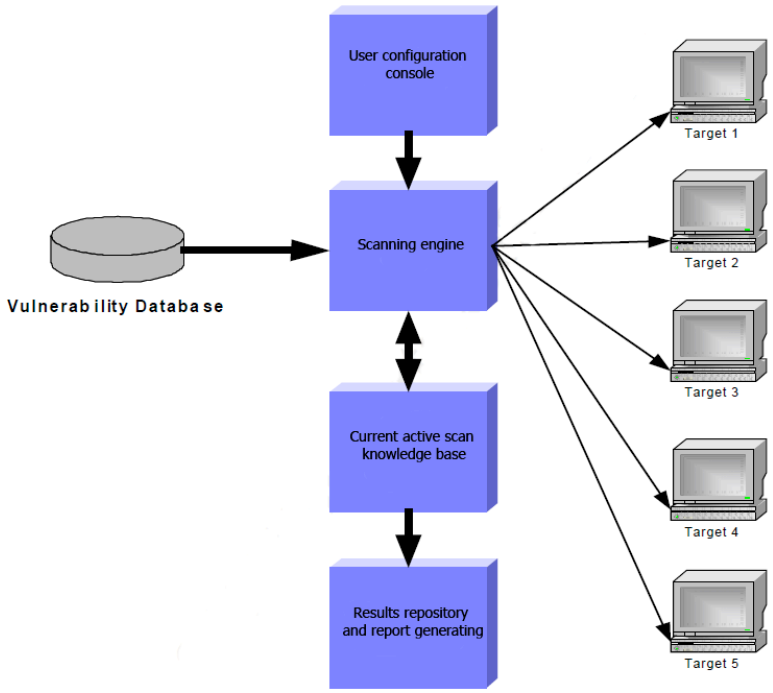

Figure 6: Typical scanner architecture

For example Nessus is an outstanding general-purpose scanner, but a web application oriented scanners like HP Web Inspect [7] or Hailstorm [8] will do a much better job when scanning a web server. In an ideal situation, scanners would not be needed because everyone would maintain well-patched and tested hosts, routers, and gateways, workstations and servers. However real world is different, we are humans and we tend to forget install updates, patch systems and/or configure systems properly. Malicious code will always find a way into your network! If a system is connected to the network that means there is a possibility this system will be infected at some time in the future. The chances might be higher or lower depending on the maintenance level system has. The system will never be secure 100%. There is no such thing as 100% security, if well maintained it might be 99.9999999999% secure, but never 100%. There is a joke which says if you want to make computer secure, you have to disconnect it from the network and power outlet and then put it into the safe and lock it. This system will be almost 100% secure (although useless), because social engineering cons may call your employees and ask to remove that system from the safe and plug it back into the network. [9]

Central Scans vs. Local Scans

The question arises – should we scan locally or centrally? Should we scan the whole network at once, or should we scan network based on sub domains and virtual LANs? The following table shows pros and cons of each method.

	Centrally controlled and access	**Decentralized scanning**
Pros	Easy to maintain	Scan managers can scan at will
Cons	Slow, most scans must be queued	Patching of the scanner is often overlooked

Localized scanning with central scanning verification. Central scanning becomes a verification audit. The question arises – should we scan locally or centrally? The answer is both. Central scans give the overall visibility into the network. Local scans may have higher visibility into the local network. Central driven scans serve as the baseline. Local driven scans are key to vulnerability reduction. Scanning tools should support both methodologies. Scan managers should be empowered to police own area and enforce policy. So what hackers will target? Script kiddies will target any easy exploitable system, while dedicated hackers will target some particular network/organization:

Who is the target?

"We are not a target". How many times have you heard this? Many people think that they don't have what to hide, they don't have secrets and, thus, nobody will hack them. Hackers are not only after secrets, but after resources as well. They may want to use your machine for hosting files, use it as a source to attack other systems or just to try some new exploits against it.

If you don't have any juicy information you might not be a target for a skilled hacker, but you will always be a target for script kiddies. In hacker culture term Script Kiddie is used to describe inexperienced hacker who is using available tools, usually with GUI, to do any malicious activity. Script Kiddies lack technical expertise to write/create any tools by themselves. They try to infect/deface as many systems as they can with least effort available. If they can't hack your system/site in a couple of minutes, usually they move to an easier target. This is rather different with skilled hackers, who seek financial or other benefit from hacking the system. They spend a lot of time just to explore the system and collect as much information as possible before trying to hack it. The proper way of hacking is data mining and writing scripts, which will automate the whole process thus making it fast and hard to respond to.

Defense In Depth Strategy

Defense in Depth is an Information Assurance (IA) strategy in which multiple layers of defense are placed throughout an Information Technology (IT) system [10]. It addresses security vulnerabilities in personnel, technology and operations for the duration of the system's lifecycle. The idea behind the Defense in Depth approach is to defend a system against any particular attack using several, varying methods. It is a layering tactic, conceived by the National Security Agency (NSA) as a comprehensive approach to information and electronic security. Defense in depth is originally a military strategy that seeks to delay, rather than prevent, the advance of an attacker by yielding space in order to buy time. The placement of protection mechanisms, procedures and policies is intended to increase the dependability of an IT system where multiple layers of defense prevent espionage and direct attacks against critical systems. In terms of computer network defense, Defense in Depth measures should not only prevent security breaches, but buys an organization time to detect and respond to an attack, thereby reducing and mitigating the impact of a breach.

Using more than one of the following layers constitutes Defense in Depth:

- Physical Security (e.g. dead bolt locks)
- Authentication and password security
- Antivirus software (host based and network based)
- Firewalls (hardware or software)

- DMZ (demilitarized zone)
- IDS (intrusion detection systems)
- Packet filters (deep packet inspection appliances and stateful firewalls [16])
- Routers and Switches
- Proxy servers
- VPN (Virtual Private Networks)
- Logging and Auditing
- Biometrics
- Timed access control
- Proprietary software/hardware not available to the public

Vulnerability Assessment Tools

There are many vulnerability assessment tools. Top 10 tools according to www.sectools.org are listed below. Each tool is described by one or more attributes:

 Generally costs money. A free limited/demo/trial version may be available.

 Works natively on Linux.

 Works natively on OpenBSD, FreeBSD, Solaris, and/or other UNIX like systems.

 Works natively on Apple Mac OS X.

 Works natively on Microsoft Windows.

Features a command-line interface.

 Offers a GUI (point and click) interface.

 Source code available for inspection.

#1 

Nessus: A Premier vulnerability assessment tool. There were 2.x version, which was open, sourced, starting from 3.x version it is closed source product. Unix like implementations have CLI/GUI and Windows implementations has GUI only. Nessus was a popular free and open source vulnerability scanner until they closed the source code in 2005 and removed the free "registered feed" version in 2008. A limited “Home Feed” is still available, though it is only licensed for home network use. Some people avoid paying by violating the “Home Feed” license, or by avoiding feeds entirely and using just the plugins included with each release. But for most users, the cost has increased from free to $1200/year. Despite this, Nessus is still the best UNIX vulnerability scanner available and among the best to run on Windows. Nessus is constantly updated, with more than 45,000 plugins. Key features include remote and local (authenticated) security checks, client/server architecture with a graphical interface, and an embedded scripting language for writing your own plugins or understanding the existing ones.

#2 

GFI LANguard: A commercial network security scanner for Windows GFI LANguard scans IP networks to detect what machines are running. Then it tries to discern the host OS and what applications are running. I also tries to collect Windows machine's service pack level, missing security patches, wireless access points, USB devices, open shares, open ports, services/applications active on the computer, key registry entries, weak passwords, users and groups, and more. Scan results are saved to an HTML report, which can be customized / queried. It also includes a patch manager, which detects and installs missing patches. A free trial version is available, though it only works for up to 30 days.

#3 

Retina: Commercial vulnerability assessment scanner by eEye Like Nessus, Retina's function is to scan all the hosts on a network and report on any vulnerabilities found. It was written by eEye, who are well known for their security research.

#4 

Core Impact: An automated, comprehensive penetration testing product. Core Impact isn't cheap (be prepared to spend tens of thousands of dollars), but it is widely considered to be the most powerful exploitation tool available. It sports a large, regularly updated database of professional exploits, and can do neat tricks like exploiting one machine and then establishing an encrypted tunnel through that machine to reach and exploit other

boxes. If you can't afford Impact, take a look at the cheaper Canvas or the excellent and free Metasploit Framework. Your best bet is to use all three.

#5 

ISS Internet Scanner: Application-level vulnerability assessment Internet Scanner started off in '92 as a tiny open source scanner by Christopher Klaus. Now he has grown ISS into a billion-dollar company with a myriad of security products.

#6 

X-scan: A general scanner for scanning network vulnerabilities. It is a multi-threaded, plug-in-supported vulnerability scanner. X-Scan includes many features, including full NASL support, detecting service types, remote OS type/version detection, weak user/password pairs, and more. You may be able to find newer versions available here if you can deal with most of the page being written in Chinese.

#7 

Sara: Security Auditor's Research Assistant SARA is a vulnerability assessment tool that was derived from the infamous SATAN scanner. They try to release updates twice a month and try to leverage other software created by the open source community (such as Nmap and Samba).

#8 

QualysGuard: A web-based vulnerability scanner Delivered as a service over the Web, QualysGuard eliminates the burden of deploying, maintaining, and updating vulnerability management software or implementing ad-hoc security applications. Clients securely access QualysGuard through an easy-to-use Web interface. QualysGuard features 5,000+ unique vulnerability checks, an Inference-based scanning engine, and automated daily updates to the QualysGuard vulnerability Knowledge Base.

#9 

SAINT: Security Administrator's Integrated Network Tool SAINT is another commercial vulnerability assessment tool (like Nessus, ISS Internet Scanner, or Retina). It runs on UNIX and used to be free and open source, but is now a commercial product.

#10 

MBSA: Microsoft Baseline Security Analyzer. Microsoft Baseline Security Analyzer (MBSA) is an easy-to-use tool designed for the IT professional that helps small and medium-sized businesses determine their security state in accordance with Microsoft security recommendations and offers specific remediation guidance. Built on the Windows Update Agent and Microsoft Update infrastructure, MBSA ensures consistency with other Microsoft management products including Microsoft Update (MU), Windows Server Update Services (WSUS), Systems Management Server (SMS) and Microsoft Operations Manager (MOM). Apparently MBSA on average scans over 3 million computers each week. [11]

Scanner Performance

A vulnerability scanner can use a lot of network bandwidth, so you want the scanning process to complete as quickly as possible. Of course, the more vulnerabilities in the database, and the more comprehensive the scan, the longer it will take, so this can be a trade-off. One way to increase performance is through the use of multiple scanners on the enterprise network, which can report back to one system that aggregates the results.

Scan verification

The best practice is to use few scanners during your vulnerability assessment, use more than one scanning tool in order to find more vulnerabilities. Scan your networks with different scanners from different vendors and compare the results. Also consider penetration testing i.e. hire white/grey hat hackers to hack your own systems. [12][13]

Scanning Cornerstones

All orphaned systems should be treated as hostile. Something in your organization that is not maintained or touched poses the largest threat. For example you have a web server and you inspect every byte of DHTML and make sure it has no flaws, but you totally forget to maintain the SMTP service with open relay it is also running. Attackers might not be able to deface or harm web page, but they will be using SMTP server to send out spam emails via your server. As a result your company's IP ranges will be put into spammers lists like spamhaus and spamcop. [14][15]

Network Scanning Countermeasures

Company wants to scan their own networks, but at the same time the company should take counter measures to protect itself from being scanned by hackers. Here is a checklist

of countermeasures to use when considering technical modifications to networks and filtering devices to reduce the effectiveness of network scanning and probing undertaken by attackers:

• Filter inbound ICMP message types at border routers and firewalls. This forces attacker to use full-blown TCP port scans against all of your IP addresses to map your network correctly.

• Filter all outbound ICMP type 3 unreachable messages at border routers and firewalls to prevent UDP port scanning and firewalking from being effective.

• Consider configuring Internet firewalls so that they can identify port scans and throttle the connections accordingly. You can configure commercial firewall appliances (such as those from Check Point, NetScreen, and WatchGuard) to prevent fast port scans and SYN floods being launched against your networks. On the open source side, there are many tools such as port sentry that can identify port scans and drop all packets from the source IP address for a given period of time.

• Assess the way that your network firewall and IDS devices handle fragmented IP packets by using fragtest and fragroute when performing scanning and probing exercises. Some devices crash or fail under conditions in which high volumes of fragmented packets are being processed.

• Ensure that your routing and filtering mechanisms (both firewalls and routers) can't be bypassed using specific source ports or source-routing techniques.

• If you house publicly accessible FTP services, ensure that your firewalls aren't vulnerable to stateful circumvention attacks relating to malformed PORT and PASV commands.

If a commercial firewall is in use, ensure the following:

• The latest service pack is installed.

• Antispoofing rules have been correctly defined, so that the device doesn't accept packets with private spoofed source addresses on its external interfaces.

• Fastmode services aren't used in Check Point Firewall-1 environments.

• Investigate using inbound proxy servers in your environment if you require a high level of security. A proxy server will not forward fragmented or malformed packets, so it isn't possible to launch FIN scanning or other stealth methods.

• Be aware of your own network configuration and its publicly accessible ports by launching TCP and UDP port scans along with ICMP probes against your own IP address space. It is surprising how many large companies still don't properly undertake even simple port-scanning exercises.

Vulnerability Disclosure Date

The time of disclosure of vulnerability is defined differently in the security community and industry. It is most commonly referred to as "a kind of public disclosure of security information by a certain party". Usually, vulnerability information is discussed on a mailing list or published on a security web site and results in a security advisory afterwards.

The **time of disclosure** is the first date security vulnerability is described on a channel where the disclosed information on the vulnerability has to fulfill the following requirement:

- The information is freely available to the public
- The vulnerability information is published by a trusted and independent channel/source
- The vulnerability has undergone analysis by experts such that risk rating information is included upon disclosure

The method of disclosing vulnerabilities is a topic of debate in the computer security community. Some advocate immediate full disclosure of information about vulnerabilities once they are discovered. Others argue for limiting disclosure to the users placed at greatest risk, and only releasing full details after a delay, if ever. Such delays may allow those notified to fix the problem by developing and applying patches, but may also increase the risk to those not privy to full details. This debate has a long history in security; see full disclosure and security through obscurity. More recently a new form of commercial vulnerability disclosure has taken shape, as some commercial security companies offer money for exclusive disclosures of Zero Day vulnerabilities. Those offers provide a legitimate market for the purchase and sale of vulnerability information from the security community.

From the security perspective, a free and public disclosure is only successful if the affected parties get the relevant information prior to potential hackers, if they did not the hackers could take immediate advantage of the revealed exploit. With security through obscurity the same rule applies, but this time rests on the hackers finding the vulnerability themselves, as opposed to being given the information from another source. The disadvantage here is that there are lower numbers of people with full knowledge of the vulnerability who can aid in finding similar or related scenarios.

It should be unbiased to enable a fair dissemination of security critical information. Most often a channel is considered trusted when it is a widely accepted source of security information in the industry (e.g. CERT, Security Focus, Secunia and FrSIRT). Analysis and risk rating ensure the quality of the disclosed information. The analysis must include enough details to allow a concerned user of the software to assess his individual risk or take immediate action to protect his assets.

Find Security Holes Before They Become Problems

Vulnerabilities can be classified into two major categories:

• Those related to errors made by programmers in writing the code for the software.

• Those related to misconfigurations of the software's settings that leave systems less secure than they could be (improperly secured accounts, running of unneeded services, etc.).

Vulnerability scanners can identify both types. Vulnerability assessment tools have been around for many years. They've been used by network administrators and misused by hackers to discover exploitable vulnerabilities in systems and networks of all kinds. One of the early well-known UNIX scanners is SATAN (System Administrator Tool for Analyzing Networks), later morphed into SAINT (Security Administrator's Integrated Network Tool). These names illustrate the disparate dual nature of the purposes to which such tools can be put.

In the hands of a would-be intruder, vulnerability scanners become a means of finding victims and determining those victims' weak points, like an undercover intelligence operative who infiltrates the opposition's supposedly secure location and gathers information that can be used to launch a full scale attack.

In fact, the first scanners were designed as hacking tools, but this is a case in which the bad guys' weapons have been appropriated and used to defend against them. By "fighting fire Race Conditions with fire," administrators gain a much-needed advantage. For the first time, they are able to battle intruders proactively. Once the vulnerabilities are found, we have to remove them:

Identifying And Removing Vulnerabilities

Many software tools exist that can aid in the discovery (and sometimes removal) of vulnerabilities in a computer system. Though these tools can provide an auditor with a good overview of possible vulnerabilities present, they cannot replace human judgment. Relying solely on scanners will yield false positives and a limited-scope view of the problems present in the system.

Vulnerabilities have been found in every major operating system including Windows, Mac OS, various forms of UNIX and Linux, OpenVMS, and others. The only way to reduce the chance of a vulnerability being used against a system is through constant vigilance, including careful system maintenance (e.g. applying software patches), best practices in deployment (e.g. the use of firewalls and access controls) and auditing during development and throughout the deployment lifecycle.

Proactive Security vs. Reactive Security

There are two basic methods of dealing with security breaches:

• The reactive method is passive; when a breach occurs, you respond to it, doing damage control at the same time you track down how the intruder or attacker got in and cut off that means of access so it won't happen again.

• The proactive method is active; instead of waiting for the hackers to show you where you're vulnerable, you put on your own hacker hat – in relation to your own network – and set out to find the vulnerabilities yourself, before anyone else discovers and exploits them.

The best security strategy employs both reactive and proactive mechanisms. Intrusion Detection Systems (IDS), for example, are reactive in that they detect suspicious network activity so that you can respond to it appropriately.

Vulnerability assessment scanning is a proactive tool that gives you the power to anticipate vulnerabilities and keep out attackers, instead of spending much more time and money responding to attack after attack. The goal of proactive security is to prevent attacks before they happen, thus decreasing the load on reactive mechanisms. Being proactive is more cost effective and usually easier; the difference can be illustrated by contrasting the time and cost required to clean up after vandals break into your home or office with the effort and money required to simply install better locks that will keep them out.

Despite the initial outlay for the vulnerability assessment scanners and the time spent administering it, potential return on investment is very high, in the form of time and money saved when attacks are prevented.

Vulnerabilities Causes

Password Management Flaws The computer user uses weak passwords that could be discovered by brute force. The computer user stores the password on the computer where a program can access it. Users re-use passwords between many programs and websites.

•Fundamental Operating System Design Flaws – The operating system designer chooses to enforce sub optimal policies on user/program management. For example operating systems with policies such as default permit grant every program and every user full access to the entire computer. This operating system flaw allows viruses and malware to execute commands on behalf of the administrator.

•Software Bugs – The programmer leaves an exploitable bug in a software program. The software bug may allow an attacker to misuse an application through (for example) bypassing access control checks or executing commands on the system hosting the application. Also the programmer's failure to check the size of data buffers, which can then be overflowed, causing corruption of the stack or heap areas of memory (including causing the computer to execute code provided by the attacker).

•Unchecked User Input – The program assumes that all user input is safe. Programs that do not check user input can allow unintended direct execution of commands or SQL statements (known as Buffer overflows, SQL injection or other non-validated inputs).

Biggest impact on the organization would be if vulnerabilities were found in core devices on the network (routers, firewalls) as follows:

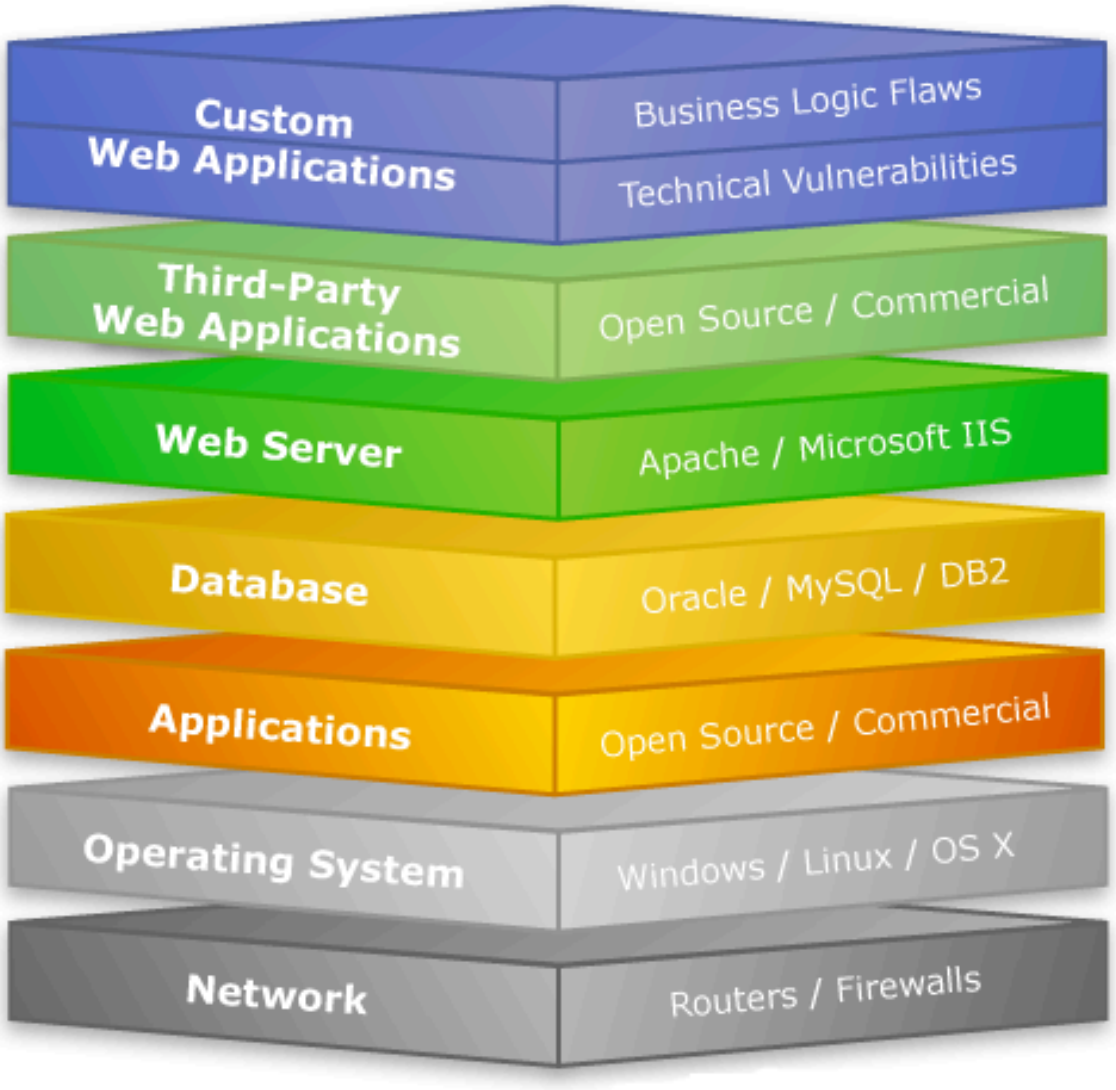

Figure 7: Vulnerabilities with biggest impact

DIY Vulnerability Assessment

If you do credit card transactions online, you're most likely PCI compliant or working on getting there. In either case, it is much better to resolve compliancy issues in an on-going basis, rather than stare at a truckload of problems as the auditor walks into your office. While writing and reviewing policies and procedures is a big part of reaching your goal, being aware of the vulnerabilities in your environment and understanding how to

remediate them is just as important. For most small businesses, vulnerability assessments sound like a lot of work and time that you just don't have. What if you could have a complete understanding of all vulnerabilities in your network and a fairly basic resolution for each, outlined in a single report within a couple of hours? Sound Good? What if I also told you the tool that can make this happen is currently free and doesn't require an IT genius to run it? Sounding better?

It isn't very pretty and it's not always right, but it can give you some valuable insight into your environment. Tenable's Nessus vulnerability scanner is one of the most widely used tools in professional vulnerability assessments today. In its default configuration, all you need to do is provide the tool with a range of IP addresses and click go. It'll then compare its database of known vulnerabilities against the responses it receives from your network devices, gathering as much information as possible without killing your network or servers, usually. It does have some very dangerous plug-ins that are disabled by default, and you can throttle down the amount of bandwidth it uses to keep the network noise levels to a minimum. The best part about Nessus is that it's (currently) free, very well documented and used by over 75,000 organizations worldwide so you know you're dealing with trustworthy product. I urge you to take a look Tenable's enterprise offerings as well. You might just be surprised at how easy it is to perform a basic Do-It-Yourself vulnerability assessment! Related links:

- Tenable's Nessus www.nessus.org
- Tenable Network Security www.tenablesecurity.com

Summary

Network based vulnerability assessment tools and host based vulnerability assessment tools are extremely useful tools in determining what vulnerabilities might exist on a particular device in the network. However, these tools are not useful if the vulnerability knowledge base is not kept current. Also, when using these tools, they can only take a snapshot of what the systems are at a particular point of time. System administrators will continually update code on the target systems and will continuously add/delete services and configure the system. All found vulnerabilities should be promptly patched, especially critical ones.

References

1. http://en.wikipedia.org/wiki/Vulnerability_assessment
2. www.pcisecuritystandards.org
3. www.darknet.org.uk/2006/04/penetration-testing-vs-vulnerability-assessment/
4. http://en.wikipedia.org/wiki/Gordon_Lyon
5. http://www.nmap.org
6. http://en.wikipedia.org/wiki/FireWire
7. http://www.scmagazineus.com/HP-WebInspect-77/Review/2365/
8. http://www.cenzic.com/products/cenzic-hailstormPro/
9. http://en.wikipedia.org/wiki/Social_engineering_(computer_security)
10. http://en.wikipedia.org/wiki/Defense_in_Depth_(computing)
11. http://www.sectools.org
12. http://en.wikipedia.org/wiki/White_hat_(computer_security)
13. http://en.wikipedia.org/wiki/Grey_hat
14. http://www.spamhaus.org
15. http://www.spamcop.net
16. http://en.wikipedia.org/wiki/Stateful_firewall