Tightening the Net: Examining and Demonstrating Commonly Available Network Security Tools

GRADUATE PROJECT

Submitted to the Faculty of the Department of Computing and Mathematical Sciences Texas A&M University-Corpus Christi Corpus Christi, Texas

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

by

Stephen E. Smith
Spring 2012

Committee Members

Dr. Mario Garcia  
Committee Chairperson

Dr. David Thomas  
Committee Member

Dr. John Fernandez  
Committee Member
Network Security tools must be developed and improved rapidly to defend against increasingly sophisticated and determined attackers. This project examines the rationale and methodology behind “Defense in Depth” network security systems, the design and functioning of these systems, and provides examples of network security tools that can be configured and used. It also contains information on network security tools, their classifications (e.g. traffic analysis, vulnerability assessment, etc.) from a review of the literature. The project details freely available network security tools and how they can be used to improve network security. For a number of tools discussed, the project features tutorial videos on their installation and use. The project concludes by emphasizing the need for network security professionals to keep themselves abreast of new developments, in this constantly changing field, regarding both threats and defenses to those threats. It also suggests areas for future work.
# TABLE OF CONTENTS

ABSTRACT ........................................................................................................... ii

TABLE OF CONTENTS ...................................................................................... iii

LIST OF FIGURES ............................................................................................... v

1. BACKGROUND AND RATIONALE ................................................................. 1
   1.1 Defense in Depth ....................................................................................... 5
       1.1.1 Baseline Analysis/Vulnerability Assessment ...................................... 8
       1.1.2 Minimization ...................................................................................... 9
       1.1.3 Patch Management .......................................................................... 10
       1.1.4 Isolation of Services ....................................................................... 11
       1.1.5 Authentication ................................................................................. 12
       1.1.6 Accountability .................................................................................. 12
       1.1.7 Controlling Network Traffic ......................................................... 12

2. NARRATIVE: NETWORK SECURITY TOOLS ............................................ 15
   2.1 Categorization of Tools .......................................................................... 16
       2.11 Vulnerability Scanners ................................................................. 16
       2.12 Packet Sniffers ................................................................................. 17
       2.13 Intrusion Detection and Prevention Systems ..................................... 19
       2.14 Vulnerability Exploit Tools ............................................................. 20
       2.15 Packet Crafting Tools ........................................................................ 20
       2.16 Password Crackers ........................................................................... 21
       2.17 Web Vulnerability Scanners ............................................................ 22
       2.18 Port Scanners ..................................................................................... 22
3. SYSTEM DESIGN: COLLECTING TOOLS FOR ANALYSIS

3.1 Selected Tools

3.11 Vulnerability Scanners

3.12 Packet Sniffers

3.13 Intrusion Detection Systems (IDS)

3.14 Vulnerability Exploit Tools

3.15 Packet Crafting Tools

3.16 Password Cracking Tools

3.17 Web Vulnerability Scanners

3.2 Learning the Tools

4. TOOL USE AND A DISCUSSION OF METRICS

4.1 Using the Tools

5. CONCLUSION AND FUTURE WORK

BIBLIOGRAPHY
LIST OF FIGURES

Figure 1.1: Plaque at Dartmouth College, honoring George Stibitz......................... 1
Figure 1.2: The CIA Triad .................................................................................................. 3
Figure 1.3: Defense in Depth key elements: People, Technology, and Operations......... 5
Figure 1.4: Example of Layered Defenses in a Defense in Depth Strategy ............... 7
Figure 1.5: Illustration of a Defense in Depth scheme. ............................................... 7
Figure 1.6: Opening Options Screen of Microsoft Baseline Security Analyzer.......... 9
Figure 1.7: A Firewall incorporated into a Gateway Router (Gouda & Liu,2005) ....... 13
Figure 2.1: Nessus Vulnerability Policies Page ............................................................. 17
Figure 2.2: Wireshark Options Screen ......................................................................... 18
Figure 3.1: Nmap Command Line Interface ............................................................... 25
Figure 3.2: Zenmap GUI Quick Scan Results .............................................................. 26
Figure 3.3: MBSA Options ......................................................................................... 28
Figure 3.4: Nessus Plug Ins Screen ........................................................................... 28
Figure 4.1: MBSA Vulnerability Report ..................................................................... 32
Figure 4.2: Nessus Scan and Security Software ....................................................... 33
1. BACKGROUND AND RATIONALE

In September of 1940, Dr. George Stibitz of Bell Laboratories, while attending a meeting of the American Mathematical Society, used a teletype link to transmit mathematical problems to his electric relay calculator (the world’s first electronic digital computer) and receive the machine’s calculated results. In so doing, he had demonstrated that the computer could be used from more than one location via telephone lines, as commemorated with the plaque shown in Figure 1.1 below (“George Stibitz,” 1995).

Figure 1.1: Plaque at McNutt Hall, Dartmouth College, New Hampshire, commemorating the successful remote operation of a computer in Manhattan via telephone lines by Bell Labs Scientist, George Stibitz

In those early days few could have imagined the widespread availability and sheer ubiquity and utility of computer networks in the various designs and configurations that exist today, let alone the modern day Internet. However, as is so often discovered the
hard way, valuable tools of great benefit can be misused by those with evil intent, causing great harm. Unfortunately, computer networks can claim no exemption from this maxim.

The same network that enables, for example, a corporate office in Seattle to rapidly exchange information with a subsidiary office in Des Moines, also provides a window of vulnerability through which unscrupulous individuals may be able to attack hosts on the network, the network itself, and to gain access to sensitive, confidential, or private data.

There has been a significant body of work at Texas A&M University—Corpus Christi on issues involving network security. Sometimes this work overlaps into the fields of computer forensics and what has come to be known as network forensics—the recovery of evidence of network intrusions—and understandably so. Reddi, for example, discusses the very real problem of “anti-forensics” or, the effort expended and tools used by computer criminals to delete any evidence that may be used against them legally (both criminally and civilly), and even to hide the fact than any sort of intrusion took place at all (Reddi, 2009).

Other recent work has looked at specific subcategories of network sites, and risks that may be inherent to those subcategories and users who connect with them. For example, Kancheti examines the growing security concerns over social networks and the risks that are most specific to those sites (Kancheti, 2010).

Finally, in recent work at TAMU—CC, there have been a number of projects dealing with detecting and preventing intrusions into networks by unauthorized personnel. One project examined two popular freeware Intrusion Detection tools, Snort
and Sax2 (Sunke, 2008), and another considered the use of Artificial Intelligence (AI) in intrusion detection (Bandela, 2010).

Network security professionals often use the “CIA triad” (unrelated to the United States Central Intelligence Agency) to illustrate three major objectives that an effective network security implementation must provide. These objectives are Confidentiality (preventing disclosure of communications between sender and receiver whether intentional or unintentional), Integrity (ensuring the accuracy and consistency of information as it moves through all parts of the network), and Availability (making sure that all who are authorized to access network resources are able to do so reliably and without undue delay). The CIA triad is illustrated graphically in Fig. 1.2 below. (Weaver, 2007, p. 26).

![Figure 1.2: The CIA Triad](image)

Network security is an integral part of the work of the Carnegie Mellon University Software Engineering Institute’s CERT® program and its work with the U.S. Department of Defense to create survivable systems. The CERT program has assisted in creating a network of Computer Security Incident Response Teams (CSIRTs) that is
national in scope, and it remains a valuable resource for anyone interested in system hardening and incident response (“Survivable Systems Analysis,” n.d.).

The field of Network Security seeks to provide reliable methods with which to thwart attacks on computer networks and prevent access to the data contained therein by unauthorized users or processes. Network security measures must be monitored for effectiveness against ever changing types and avenues of attacks. It could truly be said that the only thing about network attacks that never changes is that attacks never stop changing.

The implementation of good network security measures requires combining the functionality of a number of different security applications such as anti-virus programs, firewalls (both hardware and software), intrusion prevention systems, and (when those fail) intrusion detection systems. Configuring such methods, applications, and hardware requires not only a detailed knowledge of the network, its design, and layout, but constant monitoring of the network and the security tools used to protect it (Vigna, Valeur, Zhou, & Kemmerer, 2002). Moreover, any network security plan must take into account the all-important human element. One of the most effective ways to accomplish all of these objectives is by utilizing the concept of Defense in Depth.
1.1 Defense in Depth

The National Security Agency (NSA) describes Defense in Depth as the utilization of multiple layers of network protection that, taken together, greatly reduce the likelihood of an attack on a network, so protected, succeeding in its objective. According to the NSA (“Defense in Depth,” n.d.), the Defense in Depth strategy requires focusing on what it describes as the three key elements of network security and information assurance: people, technology, and operations.

![Defense in Depth key elements: People, Technology, and Operations](image)

Figure 1.3: Defense in Depth key elements: People, Technology, and Operations

As Figure 1.4 illustrates, all three elements must be addressed to provide effective protection to the network. The best people and technology available will not be effective in implementing a Defense in Depth (DiD) strategy, if unsafe operations are allowed to occur. No less problematic is the situation wherein technology and operations are maintained securely, but people are careless with their passwords, workstation access, or even the physical security of the building in which the various components of the network are housed.
The strength of the effectively implemented DiD strategy is that even if one layer of security is breached by an attacker, there will still be additional defensive layers that the attacker must penetrate before reaching his goal. Such layers can include any and all layers of security including physical access to those offices and buildings housing hosts and network components, authentication and password protection, operating system, antivirus protection, packet filtering, firewalls, demilitarized zones (DMZs), Intrusion Prevention Systems (IPS), Intrusion Detection Systems (IDS), Virtual Private Networks (VPNs), routers, access control, and many more (Weaver, 2007).

The goal of DiD is to put so many obstacles between an attacker and unfettered network access that in spite of defeating $x$ security layers, the attacker faces at least $x + 1$. The more time the attacker spends trying to defeat these layers, the more likely the attack is to be discovered, countered, and the attacker identified. Faced with the heavy investment of time to penetrate each layer and the fear of the serious state and federal criminal charges upon discovery (Breach of Computer Security, 2001), many attackers (though not all) will look for easier networks to crack. DiD can be implemented in numerous (and multiple) configurations. Figure 1.5, for example, illustrates the configuration of two layers of defense against attacks based on the category of attack. Figure 1.6 provides an illustration of numerous layers of defense between network computers and the outside world. Many of those same (or similar) defensive layers are deployed a second time between network computers and the inside world.
<table>
<thead>
<tr>
<th>Class of Attack</th>
<th>First Line of Defense</th>
<th>Second Line of Defense</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Defend the Enclave Boundaries</td>
<td>Defend the Computing Environment</td>
</tr>
<tr>
<td>Insider</td>
<td>Physical and Personnel Security</td>
<td>Authenticated Access Controls, Audit</td>
</tr>
<tr>
<td>Close-In</td>
<td>Physical and Personnel Security</td>
<td>Technical Surveillance Countermeasures</td>
</tr>
<tr>
<td>Distribution</td>
<td>Trusted Software Development and Distribution</td>
<td>Run Time Integrity Controls</td>
</tr>
</tbody>
</table>

Figure 1.4: Example of Layered Defenses in a Defense in Depth Strategy based on Class of Attack

Figure 1.5: Illustration of a Defense in Depth scheme. Notice the multiple layers of defense between the Outside World and Network Connected Computers repeated between those Computers and the Inside World ("Network Security," 2010).
Of course, the price of network security, like that of liberty, is eternal vigilance. For one thing, many network attacks are initiated and launched from outside of the United States and its legal jurisdiction. Such attackers have little fear of American jurisprudence. For another (and if one well-known cyber-criminal can be indicative of the attitudes of other like-minded computer criminals), some attackers are undeterred by the fear of legal consequences. In response to a 2003 bill toughening federal punishments for computer crimes, infamous social engineer, Kevin Mitnick, who served nearly six years in prison for computer crimes, expressed doubt that tougher laws would deter computer crimes. “The person who's carrying out the [crime] doesn't think about the consequences, and certainly doesn't think they're going to get caught… I really can't see people researching what the penalties are before [committing a computer crime]” (Leyden, 2003, http://www.theregister.co.uk/2003/10/03/us_intros_tougher_sentences/).

Moreover, high-value targets in which targeted networks contain military secrets, national and international intelligence, criminal justice data, or other extremely sensitive information must, of course, operate under the presumption that they are always and continuously under attack—because they probably are.

1.1.1 Baseline Analysis/Vulnerability Assessment

Baseline Analysis, considered by CERT to be the first step in hardening host systems, consists of determining the existing security status of the host. A number of tools, both open source and commercial, are available to determine the condition of the host from the standpoint of security. In 2003, CERT chose to use both the Microsoft Baseline Security Analyzer (illustrated in Figure 1.7) and the LANguard Network Security Scanner for its Windows systems. (Note: The author has included video
demonstrations of MBSA and a number of other tools in the project materials.) For its Red Hat Linux systems, the Nessus vulnerability scanner was utilized to determine the existing security status of those systems (May, C., Baker, M., Gabbard, D., Good, T., Grimes, G., Holmgren, M., Nolan, R., Nowak, R., Pennline, S., 2003).

![Microsoft Baseline Security Analyzer](image)

*Figure 1.6: Opening Options Screen of Microsoft Baseline Security Analyzer*

### 1.1.2 Minimization

The principle of minimization is that any given host system should have installed only those applications and operating system components that are essential to its functions. Windows workstations, for example, should not have Microsoft’s Internet Information Service (IIS) loaded on them because they have no legitimate need for IIS—its presence where not needed creates a vulnerability by creating an unnecessary avenue of attack. Windows file servers, on the other hand, have a genuine need for Microsoft’s IIS, but do not need (and should not have) Microsoft Office installed. Because a large
percentage of security incidents occur due to software vulnerabilities, the software installed should be kept to a minimum in order to lessen exploitation of those vulnerabilities (May et al., 2003). In general, a vulnerability scanner will make the user or administrator aware of what services are on the system. It is up to her to decide whether or not some of those services are necessary.

1.1.3 Patch Management

The profligacy of software vulnerabilities requires that all host system software be effectively and promptly patched against vulnerabilities discovered after its installation. Patch management presents quite a challenge, particularly in larger organizations with vast software libraries, but the consequences for failing to do so can be huge.

For example, in January 2003, the infamous Slammer worm exploited a vulnerability in Microsoft SQL Server, and exploded world-wide taking down ATMs, airlines, and even 911 call centers. Yet, Microsoft had issued a patch for the vulnerability the Slammer worm exploited months before the incident occurred (Cavusoglu, Cavusoglu, & Zhang, 2008). In fact, CERT/CC has estimated that nearly 95% of security breaches could be prevented by applying the necessary software patches as soon as possible (Dacey, 2003; Gerace & Cavusoglu, 2009).

The implementation of procedures for promptly updating host systems with patched code is essential to prevent the exploitation of software vulnerabilities to attack the hosts of any enterprise (May et al., 2003). Furthermore, software vendors and their customers should be clear on what they require from each other, and vendors have a responsibility to educate their customers on the classification and prioritization of patches to their products (Sihvonen & Jantti, 2010).
Gerace and Cavusoglu (2009) list seven critical elements of effective patch management, and number one on their list is senior executive support. Such support requires more than a simple recognition of the security risk unpatched software poses to their organizations, and must include support of the process of patch management, and seeing to it that the effort receives sufficient resources from the organization. Another critical but often overlooked element of effective patch management is the testing of patches prior to deployment. Such testing allows the network administrator to make sure the patches function as they were intended and do not create adverse effects on the organization’s systems. Their research found automated patch management systems most effective, and most popular among organizations (Gerace & Cavusoglu, 2009). Two commercially available patch management tools are Lumension (formerly PatchLink), which features support for multiple platforms, and HFNetChkPro supporting Windows systems.

1.1.4 Isolation of Services

Together with minimization, isolation of services is considered a best practice for system security. By isolating services such as email, web, ftp, etc. on separate physical host systems, the impact of an exploit on one service would limit the effect on other critical services within the organization (May et al., 2003). Additionally, isolation of services can take the form of a Java sandbox (a Virtual Machine within a Virtual Machine) in which untrusted code can be executed safely without placing the host (and through it the network) at undue risk (Butter & Aleksy, 2009).
1.1.5 Authentication

The process of verifying the credentials (such as username and password) of systems or users seeking access to the host system is called Authentication. Authentication serves as a sentry to ensure that only those explicitly permitted gain access. In fact, multi-factor authentication systems are increasing in use. Such authentication schemes require users to verify their identities with something they know (e.g. PIN number), something they have (e.g. electronic key), and something they are (e.g. facial recognition scan) (May et al., 2003), (Kumar, Choudhury, Sain, Lee, & Lee, 2011).

The U.S. National Institute of Standards and Technology (NIST), published electronic authentication guidelines that establish four levels for those who would access a government system with level one being the least stringent, and level four providing the highest practical level of authentication assurance (Burr, 2011).

1.1.6 Accountability

A fundamental tenet of the defense in depth paradigm is that of accountability, and it simply refers to the need for administrators to know who is doing what on a host. The goal of accountability is most often met by employing logging and auditing of host systems, which must be routinely monitored to be effective (May et al., 2003).

1.1.7 Controlling Network Traffic

Controlling what network traffic is permitted entry to or exit from a host system is considered a best practice, and can be implemented with host-based firewalls. Such a practice minimizes unauthorized network traffic, which can then be more easily logged
and inspected (May et al., 2003). It is likely that a number of firewalls will be used in a given network. Each host may have its own firewall and the network should have a minimum of one firewall at its Internet access point or router.

Firewalls can be roughly divided into hardware firewalls and software firewalls. In spite of this fact, firewalls don’t always refer to a specific piece of hardware or a specific software program. In fact, Weaver (Weaver, 2007) describes firewalls which include various combinations of hardware components and software packages.

A firewall functions by mapping or filtering packets (both incoming and outgoing) individually to a set of pre-defined rules to determine, for example, whether to accept or discard the packet. The manner in which such decisions are made determines whether the firewall is categorized as stateless or stateful. A stateless firewall determines which rules apply to a packet by examining only the packet itself. Stateful firewalls, on the other hand, apply rules based not only by examining each packet itself, but also by reviewing the packets the firewall has accepted previously. Stateful firewalls allow a greater degree of access control by maintaining information about communication states between the network and the Internet at large (Gouda & Liu, 2005).

Figure 1.7: A Firewall incorporated into a Gateway Router (Gouda & Liu, 2005)
In its simplest form, a router is a node connecting two or more networks. The router is designed to forward packets from one network to another. Routers also enable additional security measures through their own packet forwarding policies (Peterson, 2007). As illustrated in Fig. 1.9 above, routers often incorporate their own firewalls, and may be supplemented by additional hardware and software firewalls within a network.

Routers operate in two subsystems: the control plane and the forwarding plane. The primary function of the control plane is to build a routing table that identifies how best to reach various networks and their hosts—essentially to build a map on how to get from one host to another. The forwarding plane makes reference to a table where the router can find the destination addresses of incoming packets and forward them to the appropriate network or host (Gopel L, 2002).
2. NARRATIVE: NETWORK SECURITY TOOLS

As one might expect, the serious threats that exist to computer networks have generated a virtual cornucopia of tools to aid in securing such networks. These tools vary widely in how they work, what platform or platforms they support, what types of threats they seek to prevent and/or detect, at what layer they function, and their level of complexity in configuration.

It should be noted that taking available network security tools for a test drive is not without its difficulties. The first set of difficulties comes from (primarily) software firewalls which may treat some of these tools as malware and in some cases even block access to the websites from which they are available. The safest solution in such instances is to configure exceptions (individually) in the host’s firewall to allow the tools to be downloaded, installed, and run. A similar solution may be used for websites blocked by the firewall, but extreme caution must be used to make sure that each exception allowed is for the legitimate website for download of the tool, and not a site masquerading as such. If an exception is allowed for a spoofed site, the consequences could be disastrous.

Difficulties of a legal nature can arise if these tools are misused to gain access to a system the user does not own, have legal authority over, or have authorization from someone who has such ownership or authority. Although the author is not a lawyer, generally speaking, one can avoid legal/ethical problems by using these tools on her own network, a computer lab (only to the extent authorized by the instructor), and/or with the consent of the owner or authorized agent of a network (generally for testing the security
of that network). In the latter case, emphasis must be given to explaining to the client’s technical liaison when such testing is to begin and end, the types of testing to be conducted, and what (if any) effects the network might experience during testing.

Of course, the author assumes no responsibility for use or misuse of any of the tools or information discussed in this paper.

### 2.1 Categorization of Tools

While it might, with some justification, be argued that there are as many categories of network security tools as there are network security tools themselves, it is possible (and indeed necessary) to group them according to the primary security concern each is designed to address. It should be noted that there is much overlap among these categories, and for that reason one tool may be listed under more than one category while still others may defy categorization. Nmap, for example, is both a port scanner and a packet crafter and could likely be included in other categories as well.

### 2.11 Vulnerability Scanners

Vulnerability scanners are a major weapon in the fight to secure computer networks. According to one forecast, 90% of successful attacks will exploit software vulnerabilities (Zhitao & Shichao, 2010). Therefore, it is imperative that network administrators locate, identify, and correct such vulnerabilities as soon as possible.

Some freely available vulnerability scanners include Microsoft Baseline Security Analyzer, Nessus “Home Feed” (the free version of Nessus), and OpenVAS (Open Vulnerability Assessment System). Unfortunately, support for SARA (Security Auditor’s Research Assistant) was discontinued in 2009. Some commercial (non-free)
vulnerability scanners are Nessus (paid version), GFI LANguard, Retina, Core Impact, and ISS Internet Scanner. Figure 2.1 illustrates the Nessus GUI showing preloaded policies (http://static.tenable.com/documentation/nessus_4.4_user_guide.pdf).

![Figure 2.1: Nessus Policies Screen Showing Preloaded Policies](image)

2.12 Packet Sniffers

Ordinarily, when a NIC (Network Interface Card) receives a packet, it compares the MAC address to which the packet is directed to its own MAC address. A match results in acceptance of the packet, and if the packet is directed to a different MAC address, the NIC rejects it. However, if the NIC is set to promiscuous mode, all packets are accepted and none are filtered out. Packet sniffers set the NIC of the system on which they are running to promiscuous mode allowing all packets on the network to be examined and analyzed (Qadeer, Iqbal, Zahid, & Siddiqui, 2010).
Freely available packet sniffing tools include WireShark (formerly Ethereal), Kismet, Tcpdump, Ettercap, Dsniff, Ntop, Ngrep, and EtherApe, all of which are available for Linux, Windows, and Mac platforms. Cain & Abel and NetStumbler are available for the Windows platform only, while KisMAC supports only the Mac platform. Note that Cain & Abel is also a password cracking tool. As illustrated in Figure 2.2 above, WireShark features a link to a large number of Sample Captures that can be used in combination with tutorials to become familiar with all of the many features available in WireShark (http://wiki.wireshark.org/SampleCaptures).
2.13 Intrusion Detection and Prevention Systems

Intrusion Prevention Systems and Intrusion Detection Systems (IDPS, collectively) are so closely related that the National Institute of Technology and Standards (NIST) describes an Intrusion Protection System as an Intrusion Detection System that possesses the added ability to take measures to stop intrusions it detects (Kent & Mell, 2007).

Intrusion Detection Systems (IDS) can be broadly categorized into signature-based and anomaly-based tools. A signature is simply a set of properties that can be used to describe a type of network activity. Such properties can include IP addresses, port numbers, TCP/IP packet attributes and flags, and may even consist of a specific sequence of packets or a sequence of events. Signature-based tools depend upon a frequently updated database containing signatures of activities that are known to be malicious. Provided an attack signature is in the database, such tools will recognize the activity as an attack, and take a predetermined action (e.g. dropping the packets, logging the activity, and notifying the network administrator). In order for a signature-based tool to detect an attack, the signature database must contain that specific attack’s signature; therefore, the newest attacks cannot be detected (Weaver, 2007), (Al-Nashif, Y., Kumar, A. A., Hariri, S., Luo, Y., Szidarovsky, F., & Qu, G., 2008).

Anomaly-based tools, on the other hand, compare current network traffic against profiles of normal network use. Anomaly-based tools check network traffic against “normal” network behavior looking for significant variation that might signal an attack, and is especially valuable in detecting misuse of the network from within the network itself, such as an employee engaged in industrial espionage for a competitor (Weaver,
2007), (Al-Nashif et al., 2008). Drawbacks to anomaly-based intrusion detection systems include the difficulty in determining what is normal for a given system, and the fact that normal system behavior may vary widely depending on the date, time, and conditions (Nikolova & Jecheva, 2009).

Some of the freely available Intrusion Detection Systems include Snort, WinSnort, OSSEC HIDS, Fragrouter, BASE, and Sguil.

2.14 Vulnerability Exploit Tools

A vulnerability, as the term is used in this context, is a weakness in a computer or network system, its security procedures, implementation, or internal controls that makes possible the exploitation of that system by an attacker (“CNSS Instruction No. 4009,” 2010, p. 81). A vulnerability exploit is a piece of malware that takes advantage of this weakness to compromise and gain access to the system. In some exploits, commonly called “pivots”, an attacker is able to exploit a vulnerability to gain access to and control over one host, and use that host as a point from which to attack the rest of the network (Malhotra, Bhattacharya, & Ghosh, 2008). Metasploit Framework, Core Impact, and Canvas are three well known vulnerability exploit assessment tools. Metasploit is the only one of the three to be freely available with Core Impact and Canvas both costing in the tens of thousands of dollars.

2.15 Packet Crafting Tools

Packet crafting tools are useful in network testing as they enable the network administrator to create or manipulate packets that might be used by an attacker to test for security issues. One reason a packet crafting tool might be used by an attacker is to
create packets that evade detection by the network’s packet sampler and prevent the network from detecting a Denial of Service attack (Goldberg & Rexford, 2007). Some useful packet crafting tools include Hping2, Scapy, Nemesis, and Yersinia, in addition to Nmap, all of which are freely available. Interestingly, Nmap’s web site boasts that the tool was used by Trinity in the movie, Matrix Reloaded (http://nmap.org/movies.html).

### 2.16 Password Crackers

Password Crackers are useful to network administrators in determining the strength of user passwords. Moreover, police computer forensics investigators may need to crack a password in order to obtain valuable evidence of a crime. The two primary ways password crackers work are with dictionary attacks and brute force attacks. The “dictionary-attack” utilizes words found in a dictionary, but may also include attacks on known data types such as Social Security numbers, telephone numbers, and names. Authorized users of the network should never use passwords vulnerable to dictionary attacks as these are the easiest to crack (Weaver, 2007, pp. 99–100). The other attack is the “brute force attack” which consists of trying all possible combinations of characters within certain limits such as character set and password length (Bengtsson, 2007).

John the Ripper, Hash Suite (Demo Version), Cain and Abel, THC Hydra, Aircrack, Airsnort, Pwdump, RainbowCrack, and Brutus are among freely available password crackers. To illustrate how easily a dictionary password can be cracked, the author has included a video in the project materials in which a student demonstrates John the Ripper crack two simple Windows passwords from the hash file in no more than one second (http://www.youtube.com/watch?v=MGNWgFtrzb0).
2.17 Web Vulnerability Scanners

While, in a perfect world, web developers would use best coding practices, review their code for security holes, and perform penetration testing, their focus is more often on adding functionality and meeting time constraints. Moreover, web application developers oftentimes do not have a strong background in security (Vieira, Antunes, & Madeira, 2009). Therefore, it behooves the network administrator to subject web applications running on the network to vulnerability scans. Some useful freeware tools for such scanning include Nikto, Paros proxy, WebScarab, Whisker/libwhisker, Burpsuite, and Wikto.

2.18 Port Scanners

In addition to the aforementioned Nmap, other port scanners include Superscan and Angry IP Scanner. Superscan is a closed source Windows-only port scanner, while the Angry IP Scanner will run on Windows, Linux, or Mac platforms. Both are freely available.

2.19 Rootkit Detectors

Rootkits are especially dangerous pieces of software that allow an attacker to stealthily access and modify the root or administrator level of a system. Such access can be used by a bad guy to disable firewalls, log keystrokes, install backdoors, and commit other malicious acts undetected (Baliga, Ganapathy, & Iftode, 2010).

While all rootkits share the ability to access and control a host, not all rootkits are malware. Products such as eBlaster and Spector Pro are, in essence, rootkits that allow
parents to monitor and control their children’s computer activities (an activity with which the author—a single father—has had some experience).

Some available rootkit detectors include Sysinternals (Windows only), RKHunter, and chkrootkit. The latter two support UNIX and Linux systems only.
3. SYSTEM DESIGN: COLLECTING TOOLS FOR ANALYSIS

Some network security tools are quite expensive. For example, Core Security Technologies’ Core Impact Pro 8 (a vulnerability assessment and penetration testing tool) comes with a $30,000 per year price tag (Stephenson, 2009). While arguably a good investment for large enterprises, an examination of such commercial tools in this paper would, quite obviously, be cost prohibitive. Therefore, this evaluation will be limited to freely available tools. Where there exist both free and paid versions of a tool (as is the case with Nessus, for example), only the free version will be considered.

The selection criteria also include the utility of tools as perceived by the author from his research and the clarity of the tool’s output. Inclusion or exclusion from selection for research constitutes neither endorsement nor rejection of any tool by the author. It is hoped that this analysis will lead to further research on currently available network security tools, and, perhaps more importantly, those introduced in the future.

The author has assembled in the project materials some informational files on a number of network security tools. These files are grouped together by tool category, with the caveat that some tools may easily fit into more than one category. One of the author’s goals in assembling these files is to provide a source of information that can be used to assist professors in educating their students about network security tools and how they are used. These pages might also be used as a source of supplemental information for a network security class. It is hoped that they will be of use to students both in and out of the classroom.
Sometimes, one tool can complement another, either by expanding its functionality, or assisting the user in interpreting information output by the tool. An example of the latter includes Syhunt’s freeware front-end for Nmap (“A Freeware Front-end for Nmap,” http://www.syhunt.com/?n=Tools.NmapW). Nmap, which uses a CLI as illustrated in Figure 3.1, has also made available its own GUI interface, Zenmap as shown in Figure 3.2.

![Figure 3.1: Nmap command line interface.](image)
3.1 Selected Tools

Selected for inclusion in this paper are a number of freeware tools that are easily accessible to anyone, and for which much information is readily available. The author has endeavored to provide information on these tools both in this report and in the DVD files submitted with the project materials. Of course, there exist many excellent tools, and those covered by the author hardly scratches the surface of the excellent network security tools that exist.

3.11 Vulnerability Scanners

Because they are freely available and widely used, both Nessus “Home Feed” and Microsoft Baseline Security Analyzer were selected for inclusion.

Microsoft Baseline Security Analyzer is simple to download and install and can easily be used to scan a computer or multiple computers (via IP Address range) for
vulnerabilities. If scanning a single computer, the user is able to specify scanning by either computer name or IP address, and the tool provides checkboxes so the user can specify some or all of the checks he wants run, as shown in Figure 3.3.

Nessus (Home Feed) requires user registration and an email address to which an activation code will be sent. Once activated, the software provides quite a few options, and for users not familiar with the product, there is an abundance of documentation on the Tenable website, not to mention video tutorials. Note when installing Nessus on Windows it is set to run on startup. Due to the fact that it is a resource-intensive tool, users may prefer to start Nessus manually. Figure 3.4 illustrates the Nessus Plug In screen.

![Microsoft Baseline Security Analyzer](image)

*Figure 3.3: Microsoft Baseline Security Analyzer--Setting Options*
3.12 Packet Sniffers

Wireshark (formerly Ethereal) has been described as the best open source packet sniffer and network protocol analyzer to support both Windows and Linux platforms (Luo, Dong, & Jia, 2010).

That fact, together with its widespread use assured its inclusion in this project. Among others considered include Kismet, Tcpdump, Cain and Abel, Ettercap, Dsniff, NetStumbler, Ntop, Ngrep, and EtherApe.

3.13 Intrusion Detection Systems (IDS)

Snort is probably the most popular open source IDS and has been included. Others considered were OSSEC HIDS (an Open Source Host-based IDS), Fragroute
(actually an IDS evasion toolkit), BASE (The Basic Analysis and Security Engine), and Sguil.

### 3.14 Vulnerability Exploit Tools

A popular freely available tool located in this category is Metasploit Framework. It provides for developing, testing, and using exploit code. It has many exploits included in the package providing a framework with which new exploits can be written. This feature allows a network administrator to find weaknesses in his own network.

### 3.15 Packet Crafting Tools

Testing was conducted on Hping2. Other packet crafting tools considered were Scapy, Nemesis, and Yersinia.

### 3.16 Password Cracking Tools

Among the freeware password cracking tools are Cain and Abel (Windows platform only), John the Ripper, THC Hydra, Aircrack, Airsnort, Pwdump, RainbowCrack, and Brutus (Windows only). John the Ripper has been tested against several passwords of different configurations with results that will encourage users to select better passwords. John the Ripper and Cain and Abel were chosen.

### 3.17 Web Vulnerability Scanners

Paros proxy and WebScarab were considered because of their support for both Windows and Linux platforms.
3.2 Learning the Tools

One of the primary goals of this project is to provide information that will help students in the IT field learn about the tools available to help them secure the networks and systems they will encounter. Perhaps one of the most difficult aspects of learning about network security and the tools designed to enhance it is that the only constant in network traffic is that it is constantly fluctuating in volume and content. Rarely, if ever, is one minute of network traffic going to be exactly the same as another.

One method of enabling students to gain experience in using network security tools is through the use of recorded sample traffic. In fact, the Nessus website offers a large number of such sample files of varying sizes and degrees of complexity. Their website also features links to video tutorials on how to use the tool that are quite informative.
4. TOOL USE AND A DISCUSSION OF METRICS

Obviously, one tool cannot protect against all threats and attacks, one of the bases for the Defense in Depth paradigm. Testing one tool against another presents difficulties since networks are dynamic rather than static entities. Ideally, where comparisons are necessary, tests should be conducted under the same conditions (to the extent possible) to determine which exposes the greatest number of vulnerabilities and/or detects the most threats. It is anticipated that the use of more than one tool in a given category will detect more vulnerabilities and threats than any single tool alone.

4.1 Using the Tools

The tools featured vary in complexity and function. Perhaps nowhere is that variance more starkly illustrated than between the two vulnerability scanners featured: Microsoft Baseline Security Analyzer and Nessus Home Feed. The advantage of Microsoft’s tool is its simplicity of installation and ease of use. This makes it an ideal tool for use by anyone wishing to better secure a Windows host or network regardless of their technical savvy, a fact that would make the author quite comfortable in recommending the use of MBSA to his father. Figure 4.1 shows MBSA’s report after a vulnerability scan. Note the potential security risks detected.
Figure 4.1: Report generated by MBSA

Tenable’s Nessus, on the other hand, brings to the table a level of detail in both targeted scanning and reporting of vulnerabilities that go substantially beyond that of MBSA. This greater functionality however comes at the price of simplicity of
installation and ease of use. During a Nessus vulnerability scan on the author’s network, for example, a number of alerts were generated by Norton Internet Security indicating intrusion attempts. While those alerts were obviously a result of the Nessus scan (as can clearly be seen in Figure 4.2 below), and could have been avoided entirely by deactivating the Symantec product, they were also an indication that the firewall was performing the defensive role that it should. Nevertheless, the author would not feel comfortable recommending Nessus to his father!

Figure 4.2: Norton Internet Security Intrusion Attempt Alert triggered by Nessus scan

Perhaps of greater importance to system and network administrators is the question of how knowledgeable of network security issues are the people using the systems and networks they manage. The question becomes a double-edged sword for the
simple reason that the same knowledge that enables an employee to assist in securing the
network may also be used in attempts to defeat those same security measures. This gives
a clear illustration of the need to dedicate adequate security measures to the people
portion of “people, technology, and operations” outlined in the discussion on Defense in
Depth in Section 1 above.

Although measuring the performance of one particular tool against another in a
given category and situation is beyond the scope of this paper, the development of
metrics to measure the effectiveness of network security tools remains an important yet
daunting task. There may be as many ways to measure and quantify the severity of
vulnerabilities as there are vulnerabilities themselves. Moreover, rating the effectiveness
of a given network tool is not as simple as finding which tool exposes the largest number
of vulnerabilities. The existence of one undiscovered high risk vulnerability may pose
more danger to the system than 100 low risk vulnerabilities. Therefore, a given network
tool’s ability to locate and describe how to eliminate high-risk vulnerabilities must be
given greater weight than the number of vulnerabilities that it finds overall.

One measurement tool that has been employed is the Common Vulnerability
Scoring System (CVSS). Managed by the Forum of Incident Response and Security
Teams (FIRST), CVSS is intended to assist security teams in prioritizing system
vulnerabilities by providing a means of measuring their severity (Mel,
Scarfone, & Romanosky, 2007). CVSS assigns each vulnerability a severity value
between 0 and 10 with higher values indicating more severe vulnerabilities. These values
are derived from up to three different metric groups: the Base Metric (general
vulnerability characteristics), the Temporal Metric (representing severity changes over
time), and the Environmental Metric (context information that will vary depending on such variables as user and business environment). The Temporal and Environmental Metrics are optional—neither, either or both may be used with the Base Metric (Mell, Scarfone, & Romanosky, 2007).

Ahmed, Al-Shaer, and Khan (Ahmed, Al-Shaer, & Khan, 2008) argue that, all too often, rather than invest the time and financial resources to utilize the Temporal and Environmental Metrics, CVSS is employed using only the Base Metric. Such scores, by themselves, are limited in prioritization of vulnerabilities in actual practice. Using the two optional metrics will better reflect the actual effect a given vulnerability will have in the environment of a particular organization. They have described a method of quantifying network security that, using context information, seeks to identify and objectively quantify security risks, in part, by looking at existing vulnerabilities, historical vulnerabilities, predicted potential vulnerabilities, and the ability of the security policy to resist attack propagation in the network (Ahmed et al., 2008). Quantification of vulnerabilities and threats is a fascinating area for further research.
5. CONCLUSION AND FUTURE WORK

With the plethora of threats modern computer networks face, the rapid evolution of those threats requires an equally rapid response. But responding to an attack is not enough. When he entered the field of law enforcement over two decades ago, the author learned a valuable tenet: action is *always* faster than reaction. That tenet is as valid in the field of network security as it is in law enforcement. The answer in either case is the same: proactivity. By proactively seeking out the vulnerabilities in his *own* network before a bad guy can find and exploit them, the network administrator places himself in a much stronger position than one who operates only reactively. The tools examined in this project enable a more proactive approach to securing a network.

In the course of this project, the author examined a number of freely available network security tools with the goal of gathering instructive information from hands-on use of the tools, tutorial materials available on the Internet, and a review of the literature. This project has gathered together over 75 tutorial files (primarily video tutorials) designed to assist with learning not only how to use these tools, but also the different ways they may be used to probe, test, and secure a network.

This project contributes significantly to the study and practice of Network Security by examining and testing a number of different network security tools, and gathering tutorials describing and demonstrating their use. Providing an easily accessible collection of such tutorials contributes to the ability of students to learn more about a given tool *before* it is presented and discussed in a classroom or lab setting. Moreover, providing professors with the ability to assign a tutorial that students can view outside of
class will increase the amount of time professors have available in class or lab to discuss the tool in greater detail, since students new to the tool will be able to ask more informed questions. It is hoped that the author’s research along with network security tutorial files will assist in learning about the tools, how they function, and provide a flexible framework in which students and faculty can more easily run their own tests under circumstances that they, themselves, set.

Another significant contribution of this project is to encourage awareness on the part of both Information Assurance students and practitioners of the need to seek out network vulnerabilities before they are exploited, the many threats facing networks today, and additional tools that can be added to the arsenal of those seeking to maintain the security and integrity of their networks. Of course, anyone involved in network security must be fully cognizant of the dynamic nature of their work, and the need to stay as current as possible on the evolution of threats, as well as the implementation of new ways to protect against them.

Future work in network security can include building on the work begun in this project. Network security tools are updated frequently, and oftentimes the tutorials available for them are not. While many updates do not change the general operation of a given tool, they may add significant functionality to that tool. For that reason, updating tutorials on such tools would be quite helpful.

Related future work could include, for example, research on the use of virtual machines for teaching, learning, and testing network security tools. In this manner, researchers could seek out vulnerabilities and attempt to exploit them without putting the actual hosts or network at risk.
The maintenance and security of information held by various organizations of every type must be maintained if such organizations expect to keep the trust of the public. As certainly as network hardware and software engineers work to reduce vulnerabilities, it is equally certain that malevolent individuals will be working to find and exploit new ones. The fields of Network Security and Information Assurance must, of necessity, be hotbeds of innovation in the years to come.


Committee on National Security Systems (CNSS) Instruction No. 4009, 2010, p. 81


Gopel L, R. (2002). Separation of control and forwarding plane inside a network element. 5th IEEE International Conference on High Speed Networks and Multimedia Communication (Cat. No.02EX612) (pp. 161–166). Presented at the HSNMC ’02 - 5th International Conference on High-Speed Networks and Multimedia Communications, Jeju Island, South Korea. doi:10.1109/HSNMC.2002.1032568


Zhitaog., & Shichao Y. (2010). The design of the unified Vulnerability Scanning system based on SOA. *The 2nd International Conference on Information Science and*
Engineering (pp. 2444–2447). Presented at the 2010 2nd International Conference on Information Science and Engineering (ICISE), Hangzhou, China. doi:10.1109/ICISE.2010.5691366