Nour Sharabash

BA559, April 24th 2008

NETWORK SECURITY POLICY AND ASSESSMENT

Effective security policy is an imperative for every IT-conscious, modern enterprise. From the commercial vantage point, good security policy can be a business enabler. A company stands to gain significantly from first mover and early mover advantages if they can harness online supply chains or add customer value through IT. These incentives drive organizations' ambitions to connect their partners, staff, customers, and suppliers directly online to their enterprise network. On the other hand, poor security policy implemented in the rush to be first-to-market can often does result in code maintenance nightmares, recurring security issues, potential contingencies to the firm and real realized losses, etc. The list goes on. This has repercussions for management both financially and intangibly through damaged reputations. If organizations want to effectively harvest, the financial rewards associated with IT they need to develop adequate security policies and low-level IT controls to mitigate and cope with the risks of IT vulnerability. The objective of this report is to supply the reader with a general high-level overview of network security concepts and a more in-depth look into the field of network security assessment, and penetration testing.

SECTION ONE:

BASIC SECURITY POLICY

Securing the network perimeter is crucial for preventing unauthorized access. Access control is a primary objective of network-level security policy. A network assailant must only identify one unsecure system in the network periphery in order to gain access to the network. For a network administrator, the objective is much more difficult: the network administrator must effectively ensure the non-vulnerability of all systems on its network perimeter. A single access point is like an open door. For this reason, firewalls are the essential network defense, the essence of security policy.

There are two dimensions to security policy. Firewalls encompass both. These dimensions are access and control. Access allows users to use services through permitting TCP the network infrastructure--controls deny access. Thus the two functions are opposite, and yet the same. When it comes to security policy access controls need to be well defined, and this is the essential role of the firewall. There are two basic models for peripheral access and control to the network. The first is to deny access to everything except what you allow. The second is to permit access to everything except what you deny. Security administrators rarely ever take the second approach.

POLICY ENFORCEMENT

Firewalls are genuine machines. They do what they we program them to do. And if we give them good instruction, they do their jobs well. There are without a doubt limitations to firewalls and we cannot rely on them as an end-all be-all solution to vulnerability management. The threats and vulnerabilities that IT yields far extend beyond the scope of network periphery.

The other dimension to policy enforcement is users. Unfortunately, users are not so easily controllable as machines. Assailants can manipulate users with social-engineering techniques – essentially psychological cons – that can lead to utter chaos. Sometimes users are simply inept and choose weak passwords. What is more they apply the same password universally at several varying authentication points. A hacker cracks one and now he’s effectively infiltrated the entire enterprise.

BACKDOORS

Security policy is relatively ineffective against backdoors because the backdoor already exists—too little, too late—peripheral defenses can do nothing. One best addresses security issues like these with the services of a network assessment professional. The fact is, a firewall is never the end-all-be-all of peripheral network defense. Hackers can also leverage the use of modems, a long forgotten legacy from less than a decade ago, to breach a firewall. Wireless is a completely new assortment of problems all by itself. Packet injection is a very real technique whereby data packets a hacker intercepts, manipulates, than resends the data to the destination host. To do this topic justice, you must first understand TCP/IP protocol. I will discuss this topic in this report.

SECTION TWO:

UNDERSTANDING THE HACKER MENTALITY

Assailants fall into one of two categories: the opportunist or the focused assailant. Opportunistic attackers scour large Internet address spaces for vulnerable systems. Focused attackers select their pre-determined target with a specific goal in mind. Opportunists (aka “script kiddies”) pose a real, continuous threat. These individuals leverage auto-rooting tools and probing scripts to scan subsequently exploit vulnerabilities on poorly defended networks. They install out-of-the-box root kits and backdoors on the host and effectively turn the weakly secured systems into zombie-drones. These zombies come together to form bot-nets – distributed cloud grid architectures, much like Seti@Home – from which the exploit launches even more attacks, expanding their empire exponentially and launching malicious denial of service attacks on other target hosts—that is, until a vendor releases a patch. Much of the time these root kits go undetected for years. They just wait there in the Windows kernel, waiting to heed the call of their creator.

Focused attackers take a more careful and systematic approach to hacking. Their goals are clear, and they are patient. Focused attackers exhaustively probe every point of entry into a target network, port scanning every IP address and assessing each network service in depth. Even if this determined attacker cannot compromise the target network on his first attempt, he is aware of the weaknesses and the network structure. This allows the focused attacker to return later when a relevant exploit is available.

Networks at risk are the larger ones. Many entry points multiply the potential for compromise, and magnify the risk proportionately. This is the essential dilemma of security management—the attacker must only find a single point of entry. The defender must close them all.

SECTION THREE:

TCP/IP, PACKET-SEQUENCING, AND INTERNET PROTOCOL

Internet Protocol version 4 (IPv4) is the protocol of the Internet. From a network security standpoint, this is the essential subject matter of a security assessment. IPv6 is the next version of IPv4. It offers an address space that is 128bits as opposed to the 32-bit address space of IPv4. What this essentially means is that every electronic device that can have an Internet Protocol identifier address will have one. In addition, since IP is the preliminary basis upon which reconnaissance and subsequently vulnerability scanning and exploitation occur, the implication is that network security is not going to go away anytime soon. Eventually, the entire Internet will migrate across to IPv6.

NETWORK SECURITY ASSESSMENT

Security vendors offer a number of assessment services branded in a variety of ways. Vulnerability scanning uses automated systems with require minimal hands-on security experience. Make no mistake, these tools are very effective, but they are only tools—they do not provide the strategy needed—proper security policies and controls—that an enterprise must adopt if it is to receive true quality assurance.

Network security assessment combines the automation of vulnerability scanning with hands-on probing of the client network and critical vulnerabilities. Network security assessment is a step-up from vulnerability scanning and is essentially the next step in the process to what eventually becomes full-blown penetration testing. Consultants typically perform these services then provide a written report for their clients detailing network shortcomings and suggesting best-practice guidance.

Full-blown penetration testing involves multiple attack strategies: from wireless packet sniffing and packet injection, to social engineering, operating system compromise, password compromise, etc. Penetration testers are hackers in the true sense. Their objectives are to probe and exploit vulnerabilities in a network while leaving steps untraced. After all this dubious activity, a penetration tester essentially puts on the business-suits and goes back to being a consultant, providing a fully qualified-report and advising the client about the depth of security problems at the enterprise.

SECURITY ASSESSMENT METHODOLOGY

The methodology behind security assessment is roughly analogous to the sequence of procedures performed in the service-levels described just prior. There are four essential steps to network security assessment:

- Reconnaissance: Identifying the target IP network and mapping its IP network structure
- Network scanning and vulnerability probing uses automatic open-source tools such as Nmap and Nessus
- Hands-on vulnerability testing and lower-level probing
- Exploiting vulnerabilities and bypassing intrusion detection systems.

STEP 1: RECONNAISSANCE

Reconnaissance procedures are a basic element in every hacker and security professional toolbox. Assailants and specialists both use open-source tools to make this a more efficient and effective process. These parties will query public WHOIS databases, DNS name servers, newsgroups, and even search engines such as Google to gain as much information about the enterprise and its network as it can. Even a WHOIS lookup on a domain at least yields a single person's contact information and an address. With this one could essentially go dumpster-diving for tossed out equipment, possibly get lucky and find a hard-drive. That may be far-fetched but you get the point. WHOIS and DNS related tools (whois, nslookup, traceroute, etc) help the assessor map out the network by device and IP. Web crawling gains valueable contact information that comes in hand for social engineering. Often time website administrators leave troubleshooting guides nested deep within their websites that delineates valuable information such as mail servers or other services on the network, an indication of the encryption algorithm the system users need to install to get to the network (e.g. Kerberos or Active Directory), and the list goes on. Reconnaissance is the foundation. The primary objective is to gather as much information as possible without leaving a trace. This is easy for the assessor so long as the assessor does not yet jump the gun and probe the network. The objective is to gain publically accessible information. Key pieces of information the assessor gathers through reconnaissance include IP network blocks, internal IP addresses, the target's internal DNS structure (hostnames, domain names, machine names, etc.) and details on physical locations.

STEP 3: VULNERABILITY SCANNING

Vulnerability scanning is the next step in the process. Again, this process is the same for the security professional and the hacker. An assailant uses open-source tools such as Nmap and Nessus to perform bulk automated scanning of IP ranges to identify potentially vulnerable network services. Without a doubt, this kind of activity would typically set off alarms on the networks intrusion detection systems, but the assailant is generally not that stupid. An assailant has several options: they can proxy their traffic through an intermediary so that the true culprit essentially hides behind a curtain. Of course, these public proxies do not keep logs, and that is why they work. They can also "spoof" or fake their IP address. In any case, there are multiple methods of covering your tracks, but covering your tracks is critical at this stage since you are in-fact poking every single orifice of the network's body.

Nmap is a port scanner that can iterate over very large networks while performing low-level ICMP, TCP, and UDP analysis. These are all components of the IP protocol. Nmap is convenient as it comes with steal scanning functionality. Nmap is also open-source, which means it undergoes continuous improvement from an actively motivated developer base and quality assurance is a matter of opening up the source code to see it yourself.

Nessus is also an automated probing tool but with added sophistication. It can test specific network services such as Apache web server, Microsoft IIS, Oracle, and MySQL to name a few, and provides rich feedback reports on identified vulnerabilities. The security industry seems to agree that Nessus is the most comprehensive assessment tool available for this purpose. Nessus stays current with vulnerabilities through automatic “plug-in” downloads, so it always scans for the latest vulnerabilities (much as a virus scanner typically functions).

STEP 3: HANDS-ON VULNERABILITY TESTING

Many mailing lists and RSS feeds on the Internet allow one to subscribe and stay current with exploits and “proof-of-concept” exploitation techniques. Here are three URL’s for reference: www.astalavista.com , www.hackinthebox.org , www.sectools.org .

STEP 4: EXPLOITING THE VULNERABILITIES

One would think exploitation would be the difficult part, but it really is not. The entire process is quite surprisingly simple as long as one understands how TCP/IP and packet-switching works, familiar with a UNIX environment, and have a little programming and helpfully some SQL experience, then one should have little trouble climbing the hacking learning curve.

Regardless, exploitation is the final stage. After having mapped the system, assessed its weak points, experimented with the vulnerability a little bit, it is time to put it to the test. Surprisingly, all it takes is a well-developed and supported open-source framework to be well on your way to exploiting your favorite corporate or academic network. Metasploit is one such framework.

The Metasploit Framework (http://www.metasploit.org) is an open source platform with widespread community support. The developers wrote Metasploit in the Ruby programming language and the application goes so far as to leverage Ruby to allow users to develop, test, and even use exploits code. What’s more, it is very fun. I have my bias, as Ruby is my language of choice). The framework comes with a Ruby console so you can work with the application in conjunction with the language for advanced scripting and exploitation that is more powerful. It also comes with a very cool web-interface that you can point and click your way around to download, develop, and deploy the latest exploits. Purportedly, the project began as a game, evolved into one of the most powerful, and respected penetration-testing toolkit, exploit development toolkit, and vulnerability research platform available. Moreover, it is open source.

CONCLUSION

Vulnerability management for risk mitigation will never achieve perfection when it comes to network security. People make errors, errors result in unexpected bugs and bus become full-blown system compromising exploits. Such is the circle of software development. In this paper, I attempted to provide the reader with an overview of the concept and methodologies behind network security assessment in moderately low-level detail at some points and moderately high-level detail at others. The point that I wanted to drive home, however, was that vulnerability management and risk mitigation is not the product of security consultant services. Rather, these services only help identify underlying systemic problems: problems such as weakly communicated security policies and the organizational cultures that tend to simply ignore these policies. In my own personal opinion, I feel such policies are a step above the services provided by externally hired professional network assessments, but far weaker than a simple solution which I feel would mitigate security risks down to an afterthought. That solution, though I only hinted at it in this paper, is open-source. Not only is open-source software free, but it is also of a much higher caliber due to its very nature. Open source programmers develop projects that *care* about. Their goals are not to be first-to-market, as Microsoft was attempting when it released bug-ridden operating systems and office suites. To this day periodic security vulnerabilities plague these systems. Even if the administrator patches the vulnerabilities regularly with an automated updater, Microsoft will not tell you the truth that everyone in the hacking and security community knows: the Windows kernel and Windows applications riddle with backdoors—root kits—essentially the product of reversed-engineering on the Windows kernel. They sit silently in the kernel until its creator beckons. This is what bot-nets and zombie computers are all about. Would this problem exist in an actively used and managed open-source environment such as Linux? Never. The nature of open source is just that—open source. No reverse engineering, the code is plain to see. Open source mitigates the risk usually before it manifests. This occurs through contributing developers that improve or patch the code instantly upon notice, and distributions that dole out the latest application versions on a daily basis. Therefore, while the turnaround time for a company like Microsoft or another proprietary vendor to release a patch can sometimes and often does exceed 300-some odd days, the open-source community takes care of its own in a matter of days if not hours.

WORKS CITED

[1] Aaron Bayles et. al., Penetration Tester’s Open Source Toolkit. Burlington, MA : Syngress Press, 2007.

[2] Jon Erickson, Hacking: The Art of Exploitation., San Francisco : No Starch Press, 2008.

[3] Rick Lehtinen, Deborah Russell, and G.T. Gangemi Sr., Computer Security Basics, Second Edition. Sebastopol, CA : O'Reilly Media, Inc., 2006.

[4] Justin Clark, Nitesh Dhanjani, Network Security Tools. Sebastopol, CA : O'Reilly Media, Inc., 2005.

[5] Anton Chuvakin, Cyrus Peikari, Security Warrior. Sebastopol, CA : O’Reilly Media, Inc., 2004.

[6] Chris McNet, Network Security Assessment. Sebastopol, CA : O’Reilly Media, Inc., 2008.