Global Information Assurance Certification Paper

Douglas Ford
12 September 2003
GSEC Practical Assignment, ver 1.4b Option 1

8 SIMPLE RULES FOR SECURING YOUR INTERNAL NETWORK

TABLE OF CONTENTS

1. Abstract.

Many companies seem to focus a great amount of attention and funds on securing the perimeter of their network while forgetting that their most valuable assets are actually inside. The current focus on perimeter security can make it very hard for an attacker to get inside; however, once inside, they can roll with abandon with very little chance of getting caught. This paper will focus on eight areas that a company can look at to make their internal network just as hard and crunchy on the inside as on the outside.

2. Introduction.

Much of the recent discussion concerning the implementation of a "well designed, secure" corporate network has focused on how to secure the perimeter and the need for a good Demilitarized Zone (DMZ) to protect against all of those "evil hackers". Lots of thought has been given to the proper type of firewalls to use on the perimeter, where to place the Intrusion Detection System, making sure the public web server is patched (well, hopefully) and ensuring all mobile users are using Virtual Private Networks (VPN) to connect from their laptops. What seems to be missing in this equation is what to do about securing the internal network. During my experience in Penetration Testing and Red Teaming, I have found that any network can be hacked; all it takes is time, money and determination. However, if a customer would have followed the eight simple suggestions that I offer in this paper, the nefarious activities would have been detected much quicker, and the damage would have been mitigated much sooner. Bottom line, securing your internal network will save the company money in the long run.

The rules are simple and straightforward. Following them will aid in securing your network and mitigating the chance a hacker or adversary can do extended damage to your system. Ignoring them will only increase the chance for damage. So, without further adieu, I present the eight simple rules for securing your internal network.

1. Patch, Patch, Patch, and then Re-Patch.
2. An unenforced password policy is like gold in the bank for a hacker.
3. Hackers read your logs, you should too!
4. Multiple use servers mean multiple vulnerabilities to exploit.
5. Firewall your insides.
6. IDS's on the inside are as valuable as they are on the outside.
7. Why does your internal server need to connect to the internet?
8. Beware of whom you trust.

That is it in a nutshell. Follow those eight rules and your network will be better protected. Thank you and goodnight!

3. The "Rules" Explained.

So as not to leave you hanging on the edge of your seats, allow me to expand upon each of these rules.

Rule #1. Patch, Patch, Patch and then Re-Patch.

We all know that we should fully patch our computers before we connect it to the network. We also know that we should continually update our computers as patches come out. Following this logic, we also realize that if we add an additional service or item to the system we should repatch the computer. We all know these things; they are taught to us from day one in the computer world and every paper (including this one) and computer security class says the same thing. So why does this rule get ignored? Lack of time and available personnel is usually the problem but at what cost? All of the exploits my team used in our assessments were of known vulnerabilities. That means there was a patch available to counter every exploit we attempted, yet, we ALWAYS find computers that are not patched to exploit. The recent Blaster and Slammer worms are two good examples of exploits that could have been easily mitigated had patches been applied in a reasonable amount of time. Good security policy dictates that before patches get applied, they should be tested in a lab environment to ensure that no adverse effects result. This is a good theory; after all, you would not want your corporate payroll database server to break right before payday because of a patch. If your business is e-commerce, you certainly would not want your main public web server to crash due to a buggy patch. However, what about all of your generic workstations and non-critical servers? If your lead-time from patch release to patch deployment is 30 days, your network has been vulnerable long enough to be hacked. Every business has to make a cost/risk assessment[1] as to what to do, but allow me to be the heretic in the group and say patch now! Once a patch is released, wait a two or three days, check the newsgroups and if no one is complaining about the patch breaking their computers, patch! Usually the worst thing that can happen is that you have to uninstall the patch. You do make regular backups?

Another time to patch that many administrators forget about is after a re-build. Sure, the System Administrator (SA) knows all about the latest patch when it first comes out, but what about three or six months down the road? Keep a running baseline tally of the patches and service packs to be applied on every system and check them off one by one when you do a rebuild.

Make use of the automated tools that are out there such as Hot Fix Net Check by Shavlik[2]. Briefly, when you run this tool on a network or computer, it goes out to

[1] Krutz and Vines, p.16-26
[2] Shavlik Technologies, 2003

Microsoft and gets the latest list of patches that should be applied to your system and then checks to see if they have been applied. It is not a perfect tool, but it is much better than nothing. You should also use this tool or something like it to run regularly scheduled scans of your network to ensure all computers are in compliance.

Rule #2. An unenforced password policy is like gold in the bank for a hacker

By now, almost all, if not all organizations, have some kind of password policy that states something to the effect of "passwords will be at least eight characters in length, will be a combination of upper case and lower case letters, numbers and special characters"[3] etc. and so on. So why do Pen-Testers, Vulnerability Assessors and Auditors still find poor passwords being used? One reason is that there is no enforcement of this policy. It is just a paper tiger. When was the last time your Information Security or Internal IT Auditing personnel actually tested your accounts (both user and service) for strong passwords? The big legal disclaimer is to ensure you get approval from your legal department before attempting any password cracking, however, once the legal issues are settled, it is a good idea to regularly attempt to crack your organizations passwords using a commercial tool such as LophtCrack [4]or something similar. Run the password cracker, note the passwords that do not follow the proper policy and then lock out the accounts of those in violation. Send an email to those guilty individuals with a copy of the password policy attached and require them to sign a copy of the policy before unlocking their account so they can create a new password. This may sound draconian to some, but if your management supported the password policy in the first place they should support enforcement of the policy as well.

One thing to keep in mind when doing password cracking is to do the cracking on an OFFLINE system and then DO NOT STORE the cracked passwords on a computer. Numerous times during Pen-Tests, my team has found that some diligent System Administrator has indeed tested the corporate password policy, only to leave the cracked passwords in a file on their own computer!

Allow me to present two corollaries to this rule of password enforcement.

Corollary 1: System Administrators should use one time passwords.

Why should SA's use one time passwords you ask? Well, simply because, the System Administrator has the most privileges of anyone in a network so their passwords are the most coveted by hackers. Since the disclosure of the SA's passwords pose the greatest risk for damage, using a one time password mechanism such as SecurID by RSA[5] or some other similar system would help mitigate this risk. With one time passwords, if a hacker did actually intercept the

[3] SANS Institute, p. 2
[4] @stake Inc. 2003
[5] RSA Security Inc. 2003

SA's password token, it would only be good for that one session and of no use in future attacks. A better solution might actually be to require ALL users to go to one time passwords, however, in practice this does not usually work well and can become cost prohibitive. Requiring SA's to use one time passwords provides a good balance between cost and risk.

Corollary 2: Hackers love service accounts.

What are service accounts? Service accounts are simply, those accounts which do not have users associated with them. Examples of service accounts are the SMSCliToknAcct&, the SQLAgent, NAVadmin among others. Why are service accounts important? Many service accounts are created automatically when an application is installed. These accounts are assigned default passwords which many administrators never change and more importantly, many service accounts have either system or domain administrator level privileges. When the regularly scheduled password change comes around every 90 days (or whenever your local policy dictates), these accounts are often left untouched. A hacker who can crack the password of a service account, not only inherits the privileges of that account, they often can also use that password for many months to come. Besides the fact that service accounts often have elevated privilege levels, a hacker who uses a service account can often remain undetected as they roam throughout the network. After all, how many administrators, if they actually do read logs, would notice whether activity by a service account is unusual or just normal activity.

Rule #3. Hackers read logs; you should too!

Since reading logs was mentioned in the last rule, allow me to expand upon this. During many Pen-Test's and Red Teams that my group has conducted we had purposely not removed our activities from system logs to test the awareness levels of local system administrators. Unfortunately, the vast majority of the time, no one ever looked at the logs to discover our presence.

The first problem with logs is that to many overworked administrators, logs are mysterious files of hieroglyphics. Reading logs often is usually the best way to unravel the mysteries that are locked inside these strange files and gain an understanding of what is normal and what constitutes an anomaly. In addition, Microsoft has seen fit to design logged events in such a way that Egyptologists would be given a run for their money deciphering them. Fortunately, there are some tools that make this job easier. Log Parser[6] from Microsoft is one such tool that uses a command line tool. This tool can also be automated using scripts. There are also many third party tools. Some are commercial versions, which cost thousands of dollars, and some are freeware. One such freeware tool is

[6] Microsoft Corp., 2003

LogIDS[7] by Adam Richard. These tools and others like them can help decipher logs into something that an administrator can actually understand.

Another problem that administrators must be aware of concerning logs is that hackers can easily modify or remove entries. A quick search on Google™ using the string "wtmp log cleaner"[8] will turn up pages and pages of tools that a hacker can use to clean various Unix logs. As for cleaning web server logs, all a hacker needs to remove their entries in a log on an Microsoft web server is administrator or system rights a command prompt and the "find" command to remove all instances of their IP address from the W3SVC logs.

The third problem that administrators have to cope with concerning logs is the large number of logs they must read in a network. Each computer, network device and even individual application can create a log that should be read on a regular basis. There is no way an administrator can hope to read each and every log on every piece of equipment they are responsible for. Fortunately, most systems and devices (even Microsoft!) have the ability to send logs to a centralized "syslog" server[9]. An added benefit of using a centralized syslog server is that, if properly configured and locked down, it will be much more difficult for a hacker to access logs much less edit them. By using a centralized syslog server and some of the automated tools available, the administrator can easily review logs on a regular basis, recognize security alerts, perform system health analysis and save off those logs for future reference.

Rule #4. Multiple use Servers mean multiple vulnerabilities.

Time and time again, when my team has conducted a Pen-Test we have gained access to a customer's public web server only to find out that it is also a Backup Domain Controller (BDC). Web Server + Domain Controller = Game Over. Not only has my team "rooted" the customer's web server, we have also "bought" the entire domain as well. Time from start of hack to complete domain ownership, sixty seconds! I hope that previous line sinks in. Using a single server for multiple uses is contrary to Best Practices guidelines[10] and is just a bad idea. In 2002, CERT/CC reported a total of 4,129 vulnerabilities in computer software[11]. Consider for a minute the number of vulnerabilities that can be found on a multi-use server. For our example, lets look at a Corporate Web Server that is also a DNS and FTP server. To make things fair, we will not use Microsoft for our example. Our example server is running Linux/Red Hat 7.3, Apache version 1.3.23 Web Server, wu-FTP 2.6.1 and BIND 8.2.3. A quick check of the vulnerability archive at Security Focus[12] shows about 4 vulnerabilities for BIND

[7] Adam, 2003
[8] Google™, Sept 12, 2003
[9] Campi, 2003
[10] Frasier, p. 11
[11] CERT/CC, 2003
[12] Symantec Corp, 2003

v8.2.3, 18 for the Apache Web Server, and two for wu-FTP. The Red Hat site has 128 Red Hat Security Advisories (RHSA[13]) for packages associated with Red Hat 7.3. That is 152 vulnerabilities to be concerned with. Each service that is open and available to wanna-be hackers (DNS, FTP & Web in our example) is a potential avenue of compromise. Not only does the hacker have at least three services from which to attempt his exploits, he also has three ways to get an initial foothold on the server to attempt to exploit one of the 128 vulnerabilities that are associated with the Red Hat package.

Many times during customer out briefs, we have been told that the reason for using a single server for multiple uses is because of cost. "We just cannot afford to use separate servers for everything". That is just a shortsighted excuse. An entry-level server can be had for under $1000. How much is the information that can be gathered from your network worth? How much is your intellectual property worth? How much is a lawsuit for lack of due care and diligence against your company going to cost? Servers are relatively cheap, if you cannot afford to design a network that is secure, out source to a company that can. Follow the guidelines of Best Practices and use your servers a single job each.

Rule #5. Firewall your insides.

Quite often, my team has been able to compromise an External Web Server and then be able to not only map out, but also access the entire internal network. We have also more times than not, been able to compromise a host in the "human relations" section of a company and then easily connect right into the "research and development" section without any trouble. This is because although a company may have spent lots of time and money on external firewalls, there is no segregation of the internal network.

Segregating your internal network is just as important as "firewalling" your perimeter. Gartner Group[14] estimates that 70% of the security incidents that cause loss vice annoyance were commited by insiders. There is no reason that your HR department should have access to the R&D sections server, likewise, there is no reason the R&D folks should have access to the employee payroll database. A properly segregated network will cut down on the potential for inside abuse and limit the damage someone who does gain illegal entry into the network can make, since they would be limited to only a small portion of the internal network. The internal firewall (and also Intrusion Detection Systems (IDS), which will be discussed next) will also provide a finer layer of granularity when it comes to internal network events and security alerts. It is much easier to pinpoint a network event when multiple firewalls are used. Finally, using internal firewalls will allow a fine degree of Access Control Lists (ACL). Allowing FTP out of one area while blocking FTP access on another segment, or allowing SQL

[13] Red Hat Inc, Aug 2003
[14] Pescatore, p. 1

traffic into one segment while blocking it from the others are just a couple of examples of internal ACL'ing that can be used.

Rule #6. IDS's on the inside are as valuable as they are on the outside.

For all of the reasons that Firewalls on the internal network are a good idea, so are the reasons for IDS on the inside. Many times during Pen-Tests, once my team bypassed the external IDS, we were able to "run around" undetected inside a network. IDS sensors should be deployed within each segment of the network and designed to report back to a central IDS Server. This design allows an administrator to monitor one single station and have awareness of the entire network. The goal of internal firewalls is to prevent unauthorized access, the goal of the internal IDS is to alert to unauthorized access and therefore mitigate any damage before it can occur.

Rule #7. Why does your internal server need to connect to the internet?

Many times during assessments my team has found that administrators use servers as their personal workstation to do among other things, access the internet or read their e-mail. This has many problems not the least of which is that if the server is compromised when an administrator is using it, the hacker not only takes over a host which is never turned off, he also inherits the administrator access rights of the user. Internal, or any other server for that matter, should never be used as a personal workstation. In addition to this, with the threat of Cross Site Scripting[15], malicious web sites and trojan'ed emails, ACL's should be added to internal firewalls, to block by IP address, all internal servers from outbound traffic. If this cannot be accomplished due to the requirement for accessing a server from other segments, then the server should be ACL'd out at the firewall inside the outer perimeter of the network. If effective Access Control Lists are implemented, not only will the servers be denied access outside the network, but if the server attempts to access the outside, an alert would be generated and the administrator would be notified of a potential problem.

Rule #8. Beware of whom you trust!

To paraphrase William James, A chain is no stronger than its weakest link. You are a diligent network administrator, security is job one at your company, your team has spent time ensuring that the design of your network is in keeping with best practices, your IT department patches regularly and you even have a person dedicated to reading logs. You are secure! But what about that SQL server connection with the corporate field office in Tuscaloosa? Are they as diligent as you are? If they are not, then much of the hard work you have put into securing your network will be for naught. Taking advantage of trusted relationships with between other offices and organizations and your network is one of the easier ways to break into a network.

[15] Endler, p 4

Often during Pen-Tests, my team was unable to access a network through conventional means, however, using a trusted relationship with another office, we were able to penetrate the target system without notice. That SQL server that connects your accounts database with the corporate database is just one example of a trusted relationship. The Remote Access Server that your contractor uses to connect from his office to your network is another example. The list goes on and on. Anything that connects from your network to another network can be considered a trusted relationship.

Consider all the possibilities before you connect your network to another entity. Do you really need that connection? If so, how much access does that connection require? Are there firewalls and IDS's in place to not only limit access that the trusted connection has, but to alert your administrators to possible problems? Another method is to segregate the trusted connections into the DMZ. Anything that can be done to limit trusted access will go a long way in helping to ensure that your network is secure and will not be compromised by another's "weak link".

4. Conclusion.

That is it, eight simple rules for securing your internal network. Eight common Sense rules that if followed, will mitigate potential problems and will help ensure that your network does not become a victim of external hacking or internal corporate espionage.

References

Krutz, Ronald L & Vines, Russell Dean. The CISSP Prep Guide Gold Edition. Wiley. 2003. 16 - 26

Shavlik Technologies. HotFix Net Check. Sept 2003. http://hfnetchk.shavlik.com/

SANS Institute. Password Protection Policy Template. Sept 2003. 2 http://www.sans.org/resources/policies/Password_Policy.pdf

@Stake. LophtCrack (LC4). Aug 2003. http://www.atstake.com/research/lc/

RSA Security Inc. SecurID 2 Factor Authentication tokens and Smart Cards. Aug 2003. http://www.rsasecurity.com/products/

Microsoft Corp. Log Parser 2.0. Aug 2003. http://www.microsoft.com/downloads/details.aspx?displaylang=en&familyid=8cde4028-e247-45be-bab9-ac851fc166a4

Richard, Adam. SecurIT Informatique Inc. LogIDS. Aug 2003. http://iquebec.ifrance.com/securit/download.html

Google™. "wtmp log cleaner". Sept 12 2003. http://www.google.com/search?hl=en&ie=UTF-8&oe=UTF-8&q=wtmp+log+cleaner&btnG=Google+Search

Campi, Nathan. Campin Dot Net. Central Loghost Mini How-To. Aug 2003. http://www.campin.net/newlogcheck.html

Frasier, B. RFC-2196 Site Security Handbook. Internet Engineering Task Force. 1997. p. 11 http://www.ietf.org/rfc/rfc2196.txt?number=2196

CERT/CC. Reported Vulnerabilities – 2002. Carnegie Mellon University. 2003 http://www.cert.org/security-improvement/practices/p068.html

Symantec Corp. Security Focus Vulnerability Archive Database. 2003 http://www.securityfocus.com/bid/vendor/

Red Hat Inc. Red Hat Linux 7.3 Security Advisories. Aug 2003. https://rhn.redhat.com/errata/rh73-errata-security.html

Pescatore, John. "High-Profile Thefts Show Insiders Do the Most Damage". Gartner First Take. 26 November 2002. FT-18-9417. 1 http://www.sim2k.com/New/pdfs/Gartner Trust but Verify mention.pdf

Endler, David. Evolution of Cross Site Scripting Attacks Whitepaper. iDefense Inc. 2002. 1 – 25. http://www.idefense.com/XSS.html

Last Updated: May 30th, 2017

Upcoming Training

SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event