Global Information Assurance Certification Paper

Security and the Small Linux Internet Service Provider: A Roadmap

By Chad J. LaFrenz

March 31, 2001
GSEC Practical Assignment Version 1.2c

Your traditional small Internet Service Provider (ISP) doesn't have a huge staff, budget, or the resources of your typical corporate IT staff. The security of a small ISP is directly related to the knowledge level of their system administrator. This is, of course, true for any system; however, small ISPs usually only have one or maybe two administrators and not a pool of them to combine experience and knowledge like bigger IT staffs. Most small ISP administrators wear multiple hats and the security hat most likely collects the most dust. Speaking from first hand experience at a small ISP, it wasn't necessarily a lack of wanting to implement security but more a matter of what to do, where to start and how far to go. This article will provide a roadmap for the small ISP administrator and provide links to useful documentation as well as freeware programs to aid various aspects of security and monitoring. There is a resource section at the end of this paper that lists all the locations for programs listed in the article as well as websites referenced. As you should have derived from the title this roadmap is specifically geared towards small ISPs using Linux as their server operating system.

WHERE TO START

Ok, great, you want to increase security at your small ISP and the saliva is starting to form in anticipation of getting your hands dirty in super secret commands. Well, we can't begin until we create a picture of what our systems look like and map out our critical areas. This was the crucial step in my quest for security as it provided me with a visual checklist of what I had done and still needed to do. I got out the pen, a large piece of paper and started mapping out my systems and networks. Sure the big IT departments have their expensive Visio program but we can accomplish the same thing for a couple of dollars. There are some freeware programs out there as well, even one that comes installed on Red Hat Linux called dia. Dia runs under X-Windows and provides some flow-charting and diagramming features. Where to download dia is listed in the Resource section. Begin with the paper outline and then put the completed map in a program of your choice, if you like.

What are we looking for as far as detail on this map? Good question. You should include all of your physical equipment and how everything connects to each other. Then take a look at each piece of equipment and list any known services on that device.

It is now time to verify the information that you have on your map. We'll do this using a freeware program called Nmap. From Brent Deterding's article we find out the following:

> "Nmap…is the commonly accepted authority in information gathering tools. It is the first tool that both an attacker and a defender reach for, for a reason. It is an extremely versatile and useful information gathering tool that yields much of the necessary information about a machine and it's possible weaknesses."[1]

Please note that you should only scan a network that is yours and that you have permission to scan. You will be using Nmap to scan all of your mapped network devices to find out what ports are open and then derive what services each device is offering from that information. The reason being, even though you might have a server that you have setup just to offer web services via port 80 there could be a slew of other default services that are enabled and running without your knowledge. You should use the Nmap information to modify your map to list all known services that a device is currently offering to the world. I made two columns for each network device. Column one lists all services that should be offered by the device. Column two lists all other services that the device offers that need to be removed or shutdown. Keep in mind that Nmap isn't the only program that you will want to use to get a list of known services, but it provides the best initial information for each device. If you are not familiar on how to use Nmap then you should read Rich Jankowski's "Scanning and Defending Networks with Nmap" listed in the Resource section.

WHAT TO DO NOW

At this point take some time and review your current security policies and procedures, if any. Keep in mind questions like these:

- Who needs access to what devices?
- What data is most sensitive?
- What are the procedures for each device currently?

If your small ISP is like the one I work for then you won't need a lot of time, as we didn't have any written security policies or procedures. If this is the case, then you should start to develop some as you plan your security implementation. SANS Security Policy Research Project explains that the need for security policies isn't just to protect information but also to protect the individuals implementing policies.[2] If you are unfamiliar with making security policies I highly recommend reading the SANS S.P.R.P. pages referenced above to get a fuller understanding.

You should now be armed with a map of your network, all known services for each device on your network, and you have your security policies and procedures burned in your head. Using this information you can now proceed in getting your hands dirty.

Servers

An ISP's servers are their lifeblood right along with their WAN connections to the Internet. Servers are also your most vulnerable devices as they run an OS that is the target of many a hacker. We are luckier today than we were in the past as new exploits are fixed or at least found as quickly as they are released. We are also blessed in the fact that most of the major Linux distributions have update utilities packaged in them that make updating your server very easy. Keeping your server services up-to-date is one of the most critical areas of server security. Most hackers are not original and have to rely on

others to provide them with easy to use utilities to make use of known exploits. Traditionally hackers use the methodology of "scan the Internet for a specific weakness, when you find it, exploit it. Most of the tools they use are automated, requiring little interaction. You launch the tool, then come back several days later to get your results."[3] Therefore, by keeping your system updated so the known vulnerabilities do not exist you are decreasing your chance for an easy break-in by leaps and bounds. A perfect example of this is the Linux 'Lion' Worm. This worm was recently discovered in the wild by SANS/GIAC. This worm is very similar to the 'Ramen' worm and basically exploits known Bind vulnerabilities. These vulnerabilities were discovered and fixes were released for them back in January 2001[4]. In essence, this worm would have caused no damage if all system administrators of vulnerable Linux servers would have updated their Bind software back in January.

Before you launch into an update campaign you will want to remove any unnecessary services and software to ensure you are not updating something that shouldn't even be on the server. First take a look at your network map and the services columns for one of your servers. Your goal is to eliminate all of the services that you are not specifically using or need. You should not only turn those services off, but remove the software as well.

How to research open ports and services has been covered by numerous sources, so we are not going to delve into the details here. Instead please read Michael H. Warfield's "Securing Linux Part 1" or SANS "Securing Linux: Step-by-Step." Both of these are listed in the Resource section. The SANS guide is $49.00 but is worth every penny, in my opinion, and should be considered a must for any Linux system administrator. No stone is left unturned in this guide and it is very detailed in its explanations. It is included in the course materials if you take the Security Essentials course through SANS.

In addition to the guides you might also want to consider Bastille Linux if you are running either a Mandrake or Red Hat distribution. Version 1.2.0 should be released by the time you are reading this which supports up to versions 7.0 and 8.0 of Red Hat and Mandrake respectively. "Bastille Linux is a Hardening Program which enhances the security of a Linux box, by configuring daemons, system settings and firewalling."[5] Bastille Linux will walk you step-by-step through tightening your Linux box and is a very educational process as they describe each step quite thoroughly.

Besides open ports and their corresponding services there are a number of other items to look for and possibly configure.

1. SSH – You should switch to using SSH instead of telnet on all your servers. SSH provides an encrypted tunnel for accessing your servers instead of the plain text that telnet uses.

2. Default Accounts – Go through all the accounts on your servers and make sure that only the accounts you need are enabled and have strong passwords. Passwords should not be based on any dictionary word and should contain alpha (upper and lower case), numeric, and special characters.

3. Web Server – If you are running a web server make sure that you have deleted any sample programs and scripts, that it is not running as root, and that you have the directory structure it accesses properly locked down. Web servers are one of the most commonly exploited services so make sure you have it configured properly. If you are using Apache there are a number of guides to help you secure it. They are available on the Apache website.

4. FTP – FTP has about as many exploits as web servers do. If you do not need anonymous FTP access to your server then disable it. If your version allows you to use CHROOT to lock users into their home directories it is highly recommended that you do so as it will prevent users from being able to browse around your file system and download files that are readable to them. If you can, disable FTP altogether and use some of the secure alternatives that are included with SSH.

5. Backups – All the security in the world won't help recover data that gets lost or destroyed either by an attacker, physical device failure or user error. Every server should be backed up every night. If you have a lot of data then start the week with a full backup and do differential backups throughout the rest of the week. Differential backups greatly decrease the amount of time a backup takes by only backing up files that have changed since the last backup. You should also keep a full backup set off-site for increased protection or at a minimum get a fireproof safe that is rated for computer media to store your backups.

6. Physical Access – Just as important as everything you have done to limit network access to your server is limiting physical access to it. The most secure server setup does little good if your server sits in an open room with a root console open.

By closing open ports and removing their corresponding services as well as paying special attention to the aforementioned six points your server should be relatively secure. You should purchase the SANS guide and use Bastille Linux to further tighten your system. You'll always have to balance security vs. usability, but this is easily achieved if you have used your Nmap map and know what services your users require and what services for which they have no need. By implementing SSH you can safe guard your passwords from being sniffed out by someone on your network and give yourself an added layer of protection by removing telnet from your system. Remember that the only way a server can totally be secure is if you turn it off and remove all the cords. So by following this article and guides you are removing vulnerabilities and lowering your server's profile so an attacker has far less to work with if they are trying to break into your system.

Routers and other Devices

An ISP's network isn't just made up of servers, obviously. There are routers, network hubs, dial-in access equipment, and the list goes on. While most of the specifics on these

devices are beyond the scope of this document we will discuss some general topics. Specifically RFC 3013 with special emphasis on Section 4.2 involving routers will be discussed.

RFC 3013 is specially geared towards service providers and its purpose is stated as "Recommended Internet Service Provider Security Services and Procedures."[6] This RFC provides some excellent router policies that every ISP should implement. Section 4 is where Network Infrastructure is covered. BGP routing, Ingress/Egress Filtering and more are discussed in this section. In short, all of the steps they cover help increase security at the router level, make it harder for a hacker to hide their tracks and use your resources for attacks [6]. RFC 3013 should be studied and implemented at the various levels as specified.

By using the network map you created you can work through all your various devices. Your servers, as discussed, will have the most services on them and take the longest to work through. Most likely your other devices will have very few, if any, services on them. This doesn't mean they are not vulnerable. Keep the following questions in mind when evaluating your other devices:

1. Does this device have a console or telnet login? If so, do you have a strong password for it?

2. Does this device support SNMP? If so, does it need to be enabled? Are you using something besides the standard Public/Private for the community and private strings?

Once you have worked through all your routers and other devices there are a couple other layers of security that a small ISP should implement.

Monitoring

If you can't track your resources, you can't tell when something is wrong or has potentially been compromised. That is the plain and simple fact of system administration. You have to have baseline snapshots of services on each device, performance of your network, and have something that gathers current snapshots. There are a slew of network monitoring tools that are available for the small ISP system administrator to use and most of them are free.

For monitoring your Linux servers, syslogd is one of your best options. Syslogd will monitor a number of things including unsuccessful login attempts, ftp transfers, and much more. It can also be setup to send its messages to not only its local log files but to a remote syslogd server as well. It is highly recommended to setup a separate Linux server that is just for remote logging. Disable everything else except syslogd and SSH if you can. This way if someone does compromise a machine and change its logs they would have to break into the syslogd server as well to fully eliminate their tracks. This is highly unlikely if you setup the syslogd machine to be extremely secure, although they can potentially try to bring down your syslogd machine via a denial of service attack. The

syslogd man page covers its setup in depth. The basics from the man page are as follows:

- /etc/services must have an entry for syslog 514/udp on all machines.
- The –r option must be used on the machine that is remote listener.
- On each machine you must specify what auth levels to send to the remote machine and its host name. Example: *.* @hostname would send all syslog messages from the local machine to the remote machine hostname.

By having a central syslogd server you will be able to make custom scripts that can sift through all the data and provide you with meaningful information. The information can be gleamed automatically via a cron job as often as you like and only has to be performed on the single syslogd server instead of each individual server. An excellent tutorial on setting up and securing a remote log server is Eric Hines' article "Complete Reference Guide to Creating a Remote Log Server." The URL to this article is listed in the Resource section.

You should also consider running the freeware program called Snort on this machine. If you haven't heard of Snort here is a brief explanation of what it is from the Snort website:

> "Snort is a lightweight network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more. Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plugin architecture. Snort has a real-time alerting capability as well, incorporating alerting mechanisms for syslog, a user specified file, a UNIX socket, or WinPopup messages to Windows clients using Samba's smbclient."[8]

By using Snort in its Intrusion Detection System (IDS) capacity you give yourself a valuable weapon for thwarting attacks as they are happening instead of reacting to them after-the-fact. With attack tools evolving at such a rapid pace and the level of difficultly in their use dropping, system administrators must take a pro-active role in protecting their servers and networks. You cannot do this without the help of an IDS.

HOW FAR TO GO

Information and computer security is one big balancing act of usability vs. vulnerability. From a small ISP standpoint, we can't firewall all services except http, ftp, and email due to our users being so diverse in their applicational use of our services. This is directly opposite of corporate users where the corporation can specifically dictate what types of uses are acceptable on their networks. So how far can a small ISP take security? I suggest you can take it farther than you think. We have talked about how a small ISP administrator's security knowledge is directly relational to the security of the ISP, but I would also like you to think about the knowledge level of your users. If your users all suddenly open the new NaStYViRUsWoRm.ExE, for example then your servers and services could be hammered. However, if you have taken time to educate your users, the impact could be slight to non-existent.

In evaluating "how far to go" at your small ISP, look at services that you can provide either for free or a small fee that can protect your users. Postini is one such service that you might offer and their URL is listed in the Resource section. They do Spam and Virus filtering for your customers. This service isn't cheap but if you can pass the cost on to your users it could even be profitable. Alternatively work out software purchases for anti-virus, Spam filtering, or personal firewall software for your users. Provide your users the tools they need for security just as you provide your own tools to protect your servers and networks. Your users are a critical part of your network and are often overlooked.

A small ISP is in a unique position to educate and affect a number of people. Traditionally small ISPs know their user bases very well as we see them at the grocery store, soccer games, etc. We do not service 53 million nameless people. Because of this personal relationship, a small ISP should do everything they can to educate their members and protect them while they are using the Internet. Providing documentation on your website about software and security risks should be a minimum. Holding quarterly security training seminars in the local high school gym would be an excellent way to raise the awareness and education level of your users and your community. Sending email alerts as soon as known threats are in the wild will serve to minimize the damage those threats do to your services and your users. When implementing security at your small ISP go beyond the hardware and software and make the education of your users a priority. This will not only benefit you when new threats are targeting your users but will also help differentiate your services from the big nameless ISPs by having that personal touch that shows you care about your users.

SUMMARY

The security road for a small ISP is a bumpy and long road. Security isn't accomplished overnight nor is it a task that you can complete and then forget about. Security is a daily investigation and learning experience. Attackers are constantly finding new ways to exploit and use other's systems to attack even more systems.

We've talked about mapping out your systems and networks and using Nmap to show what services are running. We've provided guides to delve into securing your severs and about a free tool called Bastille Linux that will even walk you step-by-step through hardening your Linux servers.

Monitoring and setting up a syslogd remote server for logging all the information from your various servers was discussed. By using a central depository for your logging information you can comb through it quicker and easier by developing automated scripts. The advantage of using an IDS was introduced as well. An IDS enables you to be pro-active to attacks instead of reacting to the damage after the attack has occurred.

Finally, the critical area of your users' education and awareness was talked about. If you have an uneducated user base then simple attacks will cause lots of headaches. An uninformed user can ultimately cause an outbreak of a virus or worm and cause thousands of dollars of damage in terms of lost productivity and hours spent fighting the

outbreak. Educational classes, security email alerts, informational web pages, and virus/Spam filtering software or services are all ways to increase the awareness and educational levels of your users.

Hopefully this roadmap has provided you with the tools and information you need to get started on implementing better security at your small ISP. It is, by no means, the only steps you should take but by following this roadmap you are well on your way to minimizing how attacks threaten your systems, networks, and users.

Resources

1. DIA – A diagramming program like Visio. Homepage: http://www.lysator.liu.se/~alla/dia
2. Nmap – A network-scanning program. Homepage: http://www.insecure.org/nmap
3. Warfield, Michael H. "Securing Linux Part 1." URL: http://www.linuxworld.com/linuxworld/lw-1999-05/lw-05-ramparts.html
4. SANS. "Securing Linux: Step-by-Step." URL: http://www.sansstore.org $49.00. This guide is included in the GIAC Security Essentials course.
5. Bastille-Linux – GUI'd scripts to tighten a Linux server's security. Homepage: http://www.bastille-linux.org
6. Snort – A lightweight intrusion detection systems. Homepage: http://www.snort.org
7. Postini – A company that does filtering of Spam and Virus for ISPs. Homepage: http://www.postini.com

References

1. Deterding, Brent. "Nmap – The Tool, It's Author and It's Implications." 13 Jul 2000. URL: http://www.sans.org/infosecFAQ/audit/nmap.htm (35 Mar 2001).
2. "Security Policy Research Project." URL: http://www.sans.org/y2k/sec_policy.htm (23 Mar 2001).
3. Honeynet Project. "Know Your Enemy." 21 Jul 2000. URL: http://project.honeynet.org/papers/enemy (24 Mar 2001).
4. Fearnow, Matt and Stearns, William. "Lion Worm." Version 0.9. 26 Mar 2001. URL: http://www.sans.org/y2k/lion.htm (26 Mar 2001).

5. "Project Info – Bastille-Linux." 24 Nov 1999. URL: http://sourceforge.net/projects/bastille-linux (28 Mar 2001).

6. Killalea, Tom. "RFC 3013: Recommended Internet Service Provider Security Services and Procedures." Nov 2000. URL: http://cph.telstra.net/ietf/rfc/rfc3013.txt (26 Mar 2001).

7. SYSKLOGD (8) Man Page, Version 1.3, 12 Oct 1998.

8. "What is Snort?" URL: http://www.snort.org/what_is_snort.html (28 Mar 2001).