

# Securing Linux

Presented by: Darren Mobley

# Introduction

- Hello, My name is Darren
- Have been supporting and developing cPanel for over 4 years.
- We'll be covering some steps to take to help protect servers from common attacks

# Main Topics

- Local security measures
- Protecting against common remote attacks
- What to do after an attack, cleanup
- Having and following a Security Policy

# Local Attacks: Passwords

Make sure existing users have decent passwords

- Crack your own users' passwords using JTR, crack
- Preferably run the crackers on a dedicated machine, not the server, due to load
- Any passwords that crack in under a few hours need to have shell access removed until the password can be changed. This should be written into TOS/AUP which is “signed” by the client.

# Local Attacks: xinetd

- Turning off unneeded daemons in xinetd
  - Check /etc/xinetd.conf
  - Check /etc/xinetd.d/\*
    - Common ones are cupsd (printing daemon)
    - nfs/statd (unless using nfs mounted FS)

# Local Attacks: Running Processes

## Find locally running processes

- Often script kiddies will launch backdoor scripts on the server using vulnerable php scripts
- Bad clients or hacked accounts will be used to launch IRC bots / bouncers
  - ``ps auxww``
  - ``lsof -n``
- Try to find processes hidden by a rootkit, such as SuckIt
  - `mpid=`sysctl kernel.pid_max | cut -d " " -f 3`; for i in `seq 1 $mpid`; do test -f /proc/$i/cmdline && (echo -n "[${i}] "; strings /proc/$i/cmdline; echo); done`

# Local Attacks: Login Access

- Setting login access definitions
  - /etc/login.defs
    - Expire passwords after PASS\_MAX\_DAYS
    - Set minimum password length to PASS\_MIN\_LEN
    - Set number of days before pass expires to send reminder with PASS\_WARN\_AGE
    - There are more options that are well documented in the default file
  - /etc/hosts.allow and /etc/hosts.deny
    - Suggest to use firewall instead as it will protect all services, not just the ones written to obey the rules set in the hosts.\* files

# Local Attacks: Shell limits

## Setting resource limits for shell accounts

- Set in `/etc/security/limits.conf`
  - Protect against fork bombs and out of control applications, scripts
  - Will want to start out very lax, make stricter after testing with current settings; as need arises
  - Example settings:
    - `@users`      `hard`   `nofile` 500
    - `@users`      `hard`   `cpu`   30
    - `@users`      `hard`   `nproc` 150
    - `@users`      `soft`   `nproc` 100
    - `@users`      `hard`   `rss`   50000
    - `@users`      `-`      `maxlogins` 3
    - `nobody`      `hard`   `nofile` 16384



# Local Attacks: Permissions

- Find all world writable files and directories
  - `find /\( -perm -a+w \) ! -type l >> world_writable.txt`
    - reveals target locations an attacker can use to store their files
    - fixing bad perms breaks some poorly written php/cgi scripts
    - leave (/var)/tmp alone, secure it with /scripts/securetmp
- Find all setuid/gid files
  - `find /\( -perm -4000 -o -perm -2000 \) -exec ls -ldb {} \; >> suid_files.txt`
    - Many files need these elevated permissions, do not “fix” without knowing exactly how it will affect the system.
      - sudo, su, mount, traceroute, etc
- Find all files with no owner/group
  - `find / -nouser -o -nogroup`

# Local Attacks: Mount options

- Use “nosuid” option when mounting /tmp and /home
- Consider “noexec” on /tmp after cPanel installation
- Use /scripts/securetmp to have /tmp be mounted nosuid,noexec on a temporary file

# Local Attacks: IDS / Basic Forensics

- **Tripwire**
  - Monitors checksums of files, reports when they have changed.
  - A good way of helping ensure files are not replaced by rootkits/trojans/etc.
  - Commercial : <http://www.tripwire.com>
  - OSS Branch: <http://sourceforge.net/projects/tripwire>
- **Chkrootkit**
  - <http://www.chkrootkit.org>
  - Scans system for common signs of rootkits, backdoors, lkm, etc.
- **Rkhunter**
  - [http://www.rootkit.nl/projects/rootkit\\_hunter.html](http://www.rootkit.nl/projects/rootkit_hunter.html)
  - Same as chkrootkit
- **Logwatch**
  - <http://www.logwatch.org>
  - Scans through logs and emails a daily report of system activity

# Remote Attacks: Bound ports

- Find out what programs are listening on what ports
  - netstat -nap
  - Backdoor scripts/irc apps are usually launched from a writable directory, /tmp or in the user's ~ directory.
  - Most will bind to a port and wait for connections, some will “call home” in an attempt to get around P/NAT firewalling

# Remote Attacks: /proc tunables

- tcp syn cookies
  - `sysctl -w net.ipv4.tcp_syncookies=1`
    - or
  - `echo 1 > /proc/sys/net/ipv4/tcp_syncookies`
    - Helps protect against SYN flood DoS attacks
  - `echo 999999 > /proc/sys/net/ipv4/neigh/eth0/locktime`
    - Exchange eth0 with primary outgoing ethernet device
    - Increases the time that ARP caches will expire
      - Not needed if setting the ARP cache statically

# Remote Attacks: ARP Poisoning

- Tools like ettercap make ARP cache poisoning trivial.
  - Enables MITM attacks
  - Allows easy hijacking of SSL and SSH sessions, along with any other sort of connections
- Many datacenters are vulnerable to this due to management difficulties of setting static routes
  - Set your own static route, assuming that the server doesn't get moved often
    - `arp -s 192.168.1.1 00:0A:EB:D8:35:46`
      - 192.168.1.1 = default gateway
      - 00:0A:EB:D8:35:46 = NIC's MAC address

# Remote Attacks: Firewalling

- ipchains/iptables
  - Suggest using APF or similar if not familiar with iptables for ease of use and quality protection
    - Be sure to enable all the ports cPanel requires:
      - <http://faq.cpanel.net/show.cgi?qa=108499296901804>
  - Always be sure to leave yourself a way back in
    - set crontab to disable firewall every 5-10 minutes while testing new rules
    - have serial console over ip available
    - call the DC and hope they don't charge extra to have a tech flush the rules

# Remote Attacks: Apache

- Most all attacks are against poorly coded webbased applications
  - php makes poor programming easy to pull off, most target scripts are written in php
  - Backdoors, shell imitation scripts, etc can be launched to give full shell access to the server, even if the account has no shell access itself
- Enable `openbase_dir` protection in WHM
  - will stop some scripts from accessing other user accounts



# Remote Attacks: Apache

- Enable suexec for perl scripts, phpsuexec for php scripts
  - Allows tracking of scripts and forces them to run as the user of the account, rather than the “nobody” user
  - Enforces sane permissions and environment, such as not running if world writable, or in a world writable directory
  - Greatly helps when tracking exploited scripts used by spammers
  - Keeps users from doing stuff like
    - `system("killall -11 httpd");`

# Remote Attacks: Apache

- Enable “safe\_mode” for php
  - Edit the relevant php.ini
    - `php -i |grep php.ini`
  - `safe_mode = On`
  - Note that `safe_mode` is removed from php6 and later
- Edit “disable\_functions” for php
  - `disable_functions = exec, shell_exec, system, passthru, popen, virtual, show_source, readfile, pclose`
- Disable
  - `enable_dl = Off`
    - disables loading modules from inside the script

# Remote Attacks: Apache

- Considerations
  - With `php_suexec` enabled, users can put a `php.ini` in the script directory and override all settings, including `safe_mode` and `disable_functions` to run commands
  - Almost all scripting languages allow access to the filesystem as part of the language, malicious use of these functions is the real problem

# Remote Attacks: Apache

- Using mod\_security
  - Can be installed in WHM in the addons section
  - Main website at <http://www.modsecurity.org/>
  - Good ruleset to use:
    - <http://www.hostmerit.com/modsec.user.conf>
  - Allows realtime analysis of web requests and can block malicious requests
  - One of the more powerful apache modules, especially where security is concerned

# General Policy

- Give users a jailshell rather than a fullfledged shell
- Have clients use sftp, scp, smtp+ssl, pop+ssl, https://site.tld/cpanel whenever possible to avoid plain text passwords
- Use SSHv2 only, as SSHv1 is decryptable on the fly.
- Change root/admin passwords frequently using a mix of upper/lowercase letters, numbers and symbols
- Constantly monitor logs

# Stay Informed

- Join mailing lists to find out about new attacks when the information is first available
  - <http://www.securityfocus.com>
  - <http://www.securityfocus.com/archive>
- Bugtraq
  - Classic List
- Incidents
  - Good for knowing about real security breaches in the wild
  - <http://secunia.com>
  - [http://secunia.com/secunia\\_security\\_advisories/](http://secunia.com/secunia_security_advisories/)
    - High volume list much like Bugtraq
  - <http://lists.netsys.com/mailman/listinfo/full-disclosure>
    - Full Disclosure list. Mostly unmoderated, one of the best sources of current security issues

# Questions ?

## Discussion ?

- I just might know, or have an opinion; ask away!