

Discovering Mac OS X
Weaknesses
and
Fixing Them with the New
Bastille OS X Port

Jay Beale

Slides v1.0 – updated at www.bastille-linux.org/dc14.pdf

Looking at OS X Security

We'll introduce Bastille soon, but let's look at OS X's default security.

- Start with the firewall.
- Wait, the firewall isn't on?!
- OK, we're at a security conference.
We've all turned ours on!

No, we haven't.

Nmap tells me otherwise.

Ask the OS X user next to you if he activated his firewall.

- About 20% of us didn't realize we had to turn the firewall on.
- Many of us expected the firewall was on by default, as in basically every other recent O/S.

However, even people who turned the firewall on got a bad firewall.

Panther has a bad firewall?

If you've got Panther, your firewall has no
UDP or ICMP blocking.

Tiger has a bad firewall.

- If you have Tiger, you don't get UDP or ICMP filtering unless you clicked on the **Advanced** tab.
- Most security professionals, including the majority of the speakers at a recent security conference, weren't clicking that Tab.
 - UDP filtering doesn't seem like an “advanced” feature!

Tiger has a bad firewall.

- If you have Tiger, you don't get UDP or ICMP filtering unless you clicked on the **Advanced** tab.
- Most security professionals, including the majority of the speakers at a recent security conference, weren't clicking that Tab.
 - UDP filtering doesn't seem like an “advanced” feature!

But it's worse than that.

Even if you do click the **Advanced** tab, the firewall doesn't do what the GUI says it will.

It's either deceptive or it reveals that the firewall configurator designer just doesn't understand security.

The GUI can't be deceptive!

Let's check the "Block UDP Traffic" box!

We get some bad rules.

Here are the highlights:

System will accept any UDP packet as long as its source port is 5353 or 67:

allow udp from any 67 to me in

allow udp from any 5353 to me in

So, we can attack any UDP-based service?

You can attack any UDP-based service, as long as you fix your source port to either 67 or 5353.

- 67 = DHCP Server's port
- 5353 = Bonjour/Zeroconf port

DEMO: Portscanning

Let's portscan an OS X target on all of its
UDP ports, fixing our source port to 5353.

DEMO: UDP Communication

Let's talk to individual UDP services that we couldn't reach without fixing our source port.

But what can we attack on UDP anyway?

What can we attack on UDP?

First, any services the user has configured to run. If he hasn't configured any, our attacker can target:

- *ntpd*
- *CUPS*
- *Bonjour*

ntpd

There have never been any vulnerabilities in ntpd, right?

Aug 29, 2005: CVE-2005-2496

NTP ntpd -u Group Permission Weakness

Mar 5, 2004: CVE 2004-0657

*NTP ntpd Date/Time Request **Remote Overflow***

Apr 4, 2001: CVE-2001-0414

*NTP ntpd readvar Variable **Remote Overflow***

CUPS

- *Common Unix Printing System*
 - *Printing systems never have vulns!*
 - *CVE 2005-2526: CUPS for OS X contains a flaw that may allow a local denial of service. The issue is triggered when CUPS receives a partial IPP request and a client terminates the connection. The printing service **will consume all available CPU resources**, and will result in loss of availability for the CUPS printing service.*
 - *There are 32 others in OSVDB...*

No exploits today?

We may not have exploits against any of these today, but the firewall holes exposing these services to the world mean every OS X machine on the network can be nailed by the guys who brought 0-day to the wireless network.

I won't put my machine on that network.

Bonjour

- *Bonjour*

- *This tells an attacker our Security Update (bundle patch) level.*
- *It also tells an attacker our hardware and machine name.*
- *Finally, it tells an attacker what programs we have Bonjour enabled: iChat, iTunes...*

UDP Blocking

The UDP blocking provided by this firewall is quite unimpressive.

We'll come back to this, but one more point is in order.

Default Allow to CUPS?

The firewall already allowed everyone to connect to CUPS.

But I told the GUI I wasn't sharing my printer!

Default Allow to Bonjour

The firewall allows anyone to talk to Bonjour.

There's not much we can say – Zeroconf isn't for people who take their computers to wireless hotspots, hotels, or other hostile networks.

So why can't I turn it off easily?

Is that all?

We'll come back to the poor UDP blocking in a bit.

Let's look at the other "Advanced" function in the Firewall configurator:

Stealth Mode

Horrible ICMP blocking

Click on the “Enable Stealth Mode” check box.

It says:

“Ensures that **any uninvited traffic** receives **no response** - not even an acknowledgement that your computer exists.”

DEMO:

Let's scan our target:

- UDP portscan reveals no change in behavior.
- ICMP scan shows that pings illicit no response. But timestamp and network mask requests sure do.

Amazingly Non-stealthy Stealth Mode!

Here's the one rule the GUI added to the firewall:

```
deny icmp from any to me in icmptypes  
8
```

So I can do anything except send a ping!

ICMP Host Discovery?

- Timestamp requests get me system time for cryptographic attacks
 - But they're also just good for system discovery, as implemented in nmap.

```
nmap -sP -PE target
```

- Netmask requests are also in nmap:

```
nmap -sP -PM target
```

Host Discovery

Remember the GUI description of Stealth Mode?

“Ensures that **any uninvited traffic** receives **no response** - not even an acknowledgement that your computer exists.”

I can get a response with two types of ICMP packets and some easy UDP packets.

Think everyone knows this?

- Every Mac-toting person I spoke to at a recent security conference, save one, hadn't created custom rules.
- Without custom rules, you get substantial weaknesses in your firewall that the GUI never leads you to expect.
- Let's look at the other rules that activating UDP blocking gave us.

DEMO: Exploring Firewall Rules

Let's take a look at the other weak firewall rules in place.

The GUI doesn't give you a good firewall.

You're going to need to make your own.

It's not much work, but the new OS X port of Bastille Linux will do it for you, helping you create only the blocking exceptions that you actually wanted.

Let's explore a few other Apple security issues.

- Bonjour
- Netinfo
- Bluetooth
- Multi-user security

DEMO: Bonjour (1/4)

If we interrogate Bonjour, we can remotely get your OS X Security Update level.

This tells the attacker what patch bundle level you're up to and whether she should spend the time to attack you or pick another target.

DEMO: Bonjour (2/4)

Anybody up for a different kind of
Wall of Sheep?

The Wall of Patchless Sheep

DEMO: Bonjour (3/4)

If we interrogate Bonjour, we can remotely get your Machine Name.

This usually tells the attacker the name of the admin user or at least gives her a good hint.

Also useful for the Wall of Patchless Sheep.

DEMO: Bonjour (4/4)

If we interrogate Bonjour, we can remotely get your Machine hardware type.

Choosing exploits for those UDP services is easier if the attacker knows exactly what hardware you're running.

Also, she can better find you in the room, take your picture, and put you on the Wall of Patchless Sheep!

DEMO: Netinfo

If we're on the system, we can pull an account listing that shows us the length of all passwords on the system.

That sure makes guessing passwords easier.

DEMO: Bluetooth

The default Bluetooth configuration is:

- Bluetooth on
- machine discoverable
- encryption off
- user auth of Bluetooth actions not always present

DEMO: Weak User Security

- DEMO: All users can see each other's files
- DEMO: autologin is on by default
- DEMO: the first user created can Trojan any application
 - A browser vuln can replace my applications – its like running as root

Show: More Weak User Security

When we run as non-admin, typing admin user and password to install software, the software still gets owned by our user!

- So our user can Trojan apps he has installed.
- This is like the old Finder flaw where app installs went in world-writable...

A Good Defense?

Let's look at how we can harden this system.

Introducing Bastille for OS X.

DEMO: Bastille on OS X.

- We can audit a system.
- We can harden it.
- We can re-audit.
- Let's talk about what Bastille is doing.

Isn't that like Bastille Linux?

- Bastille has been one of the most popular hardening and audit tools for six years.
- Bastille ships in HP-UX as part of the installer.
- Bastille is available for almost every major Linux distribution, often through automatic installation tools.
- Bastille now extends full support to OS X Tiger with a native port, available through an OS X install package.
- Does anyone want to use the Cocoa library to get a native OS X front-end?

Bastille Linux Background

- Bastille is a hardening and audit program for:
 - Red Hat, SUSE, Mandriva, Ubuntu, Gentoo, Debian Linux
 - HP-UX
 - OS X Tiger!

Bastille is both an implementation/audit tool and an educational tool.

- Each hardening item is also an audit item.
- Each hardening/audit item teaches the user about the choice he's making.
- Teaching admins and users helps them make better choices for better security.

Bastille Breaks Exploits

- Deactivating programs that would have gotten exploited breaks exploits by giving them nothing to hit.
- Configuring programs better breaks exploits because vulnerable code isn't accessible
 - This works when kernel-level containment fails you because the program never gets exploited!
- Containment configurations (like chroot jails) break exploits because the exploit expects to run programs that aren't present.

Bastille Effectiveness

- Bastille released after Red Hat 6.0 but before any exploits were discovered
- Without any foreknowledge, Bastille broke every major exploit against Red Hat.
 - All network-level ones: BIND, WU-FTPd, Sendmail+lpd
 - All Set-UID ones: dump, restore
 - All local daemon ones: gpm
 - Didn't break the ones against the man or nmh commands

Hardening Works

NSA's IAD tested working exploits against Windows after hardening with a hardening guide.

- They found 19 out of 20 exploits were broken.

Bastille Does Hardening Assessment

- Separate read-only mode to tell you what is hardened vs what is lacking
- Scores a system
 - Triage – which machines are in the best shape
 - Motivation – admins more proactively harden systems, like to get high scores, management doesn't want low scores.
- Works for skew-detection after patching
 - You can check a system against a policy file that says which items are important to your org / standard / guide

Learn about OS X Lockdown

- Want to see what Bastille does?
- You can use this talk to do it yourself if you don't dig tools.

Major OS X Steps

- Install a fully-configurable, non-deceptive firewall.
- Deactivate (optionally) Bonjour.
- Lock non-root users out of Netinfo.

DEMO: Hands-on

- Fix your firewall right now.
 - We'll remove the Major Gaping Holes to make the wireless network safer.
- Turn off Bonjour right now
 - You have to know the secret sauce – Apple doesn't want you to do this.

DEMO: Attack the System

Let's see how just these three steps majorly decrease the exposure of an OS X machine.

Hardening Bluetooth

- Bluetooth – that “other” wireless?
- MACs are **discoverable by default**.
- Basically, all Macs ship with Bluetooth.
- Turn off discoverability.
- Require pairing for everything.
- Turn on encryption where you can.

Hardening: User Account Access

- Make a normal user account so we don't run everything with a user that has admin privs
- Kill off user listing at the login screen
- Set up home dir encryption
- Turn off the “everyone can see each other's files” default stance
- Kill off autologin.
- Educate the admin on chown-ing after installations.

Hardening: Apache Web

- Rip out Apache modules to decrease available exploitable code.
- Add security-focused Apache modules, pre-compiled for OS X.
- Chroot the Apache server.
- Misc config steps.

Hardening: BIND DNS

- Chroot BIND
- Run BIND as normal user

Hardening: FTP

- Chroot users
- Restrict users who can log in

Hardening: Postfix

- Chroot components from each other
 - Breaks exploits that require interaction
 - Contains exploits that succeed

Hardening: Deactivation

- Deactivate everything we're not using
- **DEMO:** launchd restarts things we thought we turned off.
- Show how to deactivate each major way:
 - Launchd
 - Rc
 - SystemStarter

That's all folks!

- Questions with our remaining time...
- After that, I'll see you in the hallway outside!

Jay Beale

Jay Beale is a security consultant working for Intelguardians. He wrote Bastille Linux, the Center for Internet Security's first Unix Scoring Tool, columns and articles for Information Security Magazine, SecurityPortal, and SecurityFocus, as well as a number of books, including those in the Jay Beale's Open Source Security Series.