Boot Security

and Physical Attacks

Exercise

This exercise's worksheet will be up on the projector.

Start up your laptop and break into it.

We will log in as root for the remainder of the class.

Physical Attacks and Defenses

The easiest method of attack is the physical attack.

Let’s talk about attack and defense on the physical side to motivate our thinking here.

Story time: the barely-guarded laptop at Def Con…

GRUB: linux init=/bin/bash

On a GRUB enabled system:

Hit **e** to edit the boot configuration

Use arrow keys to select the kernel line

Hit **e** to edit the kernel line

Append "**`init=/bin/bash`**" to the end.

Hit enter, then **b** to boot

Why Did This Work?

Review the boot process:

BIOS/EFI/EEPROM

Bootloader (GRUB)

kernel

init / upstart / systemd

rc scripts / upstart jobs / systemd services

Understanding the attack: init replacement

Trace the process to find out what **init=/bin/bash** does:

BIOS -> GRUB -> kernel -> /bin/bash   (instead of init)

Password-protecting the GRUB prompt

1. Create an MD5-hashed password:

```
# grub-md5-crypt
```

2. Place it in `/boot/grub/grub.conf`, before first boot entry:

```
password  --md5 hashedpassword
```

3. Set permissions:

```
chmod 600 /boot/grub/grub.conf
```

Protecting against init replacement

Password-protect the bootloader prompt!

We can apply a password so that these machines will boot, but will only allow a user to modify the boot settings if they know the preset password.

Re-Try the Attack

It's important to retry the attack now, to make sure that the defense actually worked!

Are We Done?

So, we've protected against that attack.

Are we done?

Well, no. Let's look at another attack.

Still trying to get root...

So we can't control the machine's bootloader.

Whatever will we do?

Bring our OWN bootloader on a CD or USB key!

http://unetbootin.github.io/

Boot off our own media!

If we boot off our own device, we start out as root!

Then, we can mount the system's main drive and modify it at will.

1) Boot from the optical/USB drive
2) Mount the hard drive and modify it:

```
mount /dev/sda1 /mnt
echo "r00t::0:0:r00t:/:/bin/bash" >>/mnt/etc/passwd
```

What happened?

How do we defend against this?!

The attacker bypassed our bootloader, our init program and just created himself an account!

The attacker still had to use our BIOS to boot off that external device!

Defense: Deactivate USB and optical drive booting

Let's reboot the system and go into the BIOS settings.

We can tell the system not to boot off of the removable drives.

We had better remember to password protect our CMOS/EFI/UEFI.

Encrypted Root Partition

At this point, we’ve got one final move.

We can encrypt the root partition.

With that said, an attacker could install a keystroke logger on the keyboard, modify the first-stage bootloader that loads the encrypted partition, and so on.

Reference: Evil Maid Attacks

Time, Effort, Class of Attacker

The point of this exercise is several-fold.

- Increase level of expertise an attacker needs.
- Slow down the attacker. (Make this a 30 minute safe)
- Force the attacker to find/bring better tools. (Now they have to bring a screwdriver?)
- Increase the chances of catching the attack.
- Break the attack!

The Fight for Security

The bad news is that you're never guaranteed a win.

The good news is that your attacker isn't either!

Researchers are always seeking vulnerabilities and countermeasures to our defenses.

It's our job to deploy the best defenses that we have and that we can create.

We can dramatically reduce an attacker's chances.