When full is only 99.9% full

SecTalks Sydney

Kaan Kay / ls

Who is this guy?

- Kaaaaaaan aka ls (ell-ess, not eye-ess)

- I was once a web developer...
- I was also twice a security consultant...
- I’m now just trying to secure hoards of data

What the frack is Secure Boot?

- Allows firmware to make trust decisions
- Based on public-key cryptography
- Attempts to provide a secure boot chain

- Windows? Pretty damn good!
- Linux? It’s… alright… you’ll see...
- OS X? I love OS X, so let’s not talk about it...

Time for a story!

- Drunken challenges are fun
- Finding a default remote zero-day is hard
- Dealing with full disk encryption is easier
- Decided to target /boot and dm-crypt because I didn’t have a hardware keylogger and for some reason people are paranoid these days?

Simplified Linux boot process

- BIOS/UEFI -> MBR -> GRUB
- GRUB: Stage 1 -> Stage 1.5 -> Stage 2
- Load initial RAM disk as root and the kernel
- Mount the real root file system
- Execute /sbin/init on the real root file system
- Remove the initial RAM disk

An obvious initrd modification

- Passphrase needs to be entered to boot
- dm-crypt uses cryptsetup to do this
- Modify the boot shell scripts
- Store the passphrase anywhere we can
- Grab it later on, mount the disk and make whatever modifications we want

Show and tell plus a demo

How to (sort of) avoid this

- Alternatives include:
  - Boot from /boot on removable media?
  - Use Secure Boot to verify ~~initrd~~ and vmlinux?
  - Throw a TPM into the mix?
- The above have their own issues too
- Hardware keyloggers still win
- Mossad will still get you (James Mickens)

Thanks! Questions?

@0x6c73
ls@moar.so