Jailbreaking

CS 161: Computer Security

Prof. David Wagner

April 27, 2016

Types of attacks

- Jailbreaking
- Rooting
- Unlocking

iOS Jailbreak

- iPhone used a protocol which allowed iTunes to write arbitrary files on iPhone's filesystem
  - Why? iTunes needs to be able to sync files, push music to your iPhone, overwrite its code with software updates

- Exploit: Write a program that pretends to be iTunes and uses that protocol to overwrite kernel on iPhone with one that doesn't enforce Apple's restrictions

iOS Jailbreak

- iPhone used a protocol which allowed iTunes to write arbitrary files on iPhone's filesystem
  - Why? iTunes needs to be able to sync files, push music to your iPhone, overwrite its code with software updates
- Exploit: Write a program that pretends to be iTunes and uses that protocol to overwrite kernel on iPhone with one that doesn't enforce Apple's restrictions
- If you were Apple, how would you fix this?

Jailbreakme.com

- Researcher finds buffer overrun in libtiff parsing code
- Someone notices that iOS browser uses libtiff, and runs as root.

- Exploit: browse to http://jailbreakme.com/ on your iPhone, they send you a malicious TIFF image that exploits buffer overrun, does code injection to run code that overwrites the iOS kernel on the filesystem with a modified kernel.

Jailbreakme.com

- Researcher finds buffer overrun in libtiff parsing code
- Someone notices that iOS browser uses libtiff, and runs as root.

- Exploit: browse to http://jailbreakme.com/ on your iPhone, they send you a malicious TIFF image that exploits buffer overrun, does code injection to run code that overwrites the iOS kernel on the filesystem with a modified kernel.

- If you were Apple, how would you fix this?

Pwnage 2.0

- Stack overflow in the iOS certificate parsing code
- Exploit: Send a malicious cert, exploit the buffer overrun, do code injection (just like Project 1), run code that overwrites kernel with modified kernel

steaks4uce

- Buffer overrun in the iOS code for handling a particular USB request

- Exploit: Send a malicious packet over USB, exploit the buffer overrun, do code injection, etc.

Defenses against jailbreaking

- Jailbreak = exploitable software vulnerability (often, buffer overrun)

- iOS architecture: all code is signed.
  - Bootloader verifies signed firmware, before starting it.
  - Firmware verifies signed kernel, before starting it.
  - Kernel verifies signed apps, before starting them.

- iOS architecture: apps are sandboxed.
  - System services run as user ‘root’
  - Apps run as user ‘mobile’, and are sandboxed (limited in what files they can write, devices they can access, …)

Example exploit chain

- Star PDF vulnerability: stack overflow in font parser, lets you get your code running as ‘mobile’.
- IOKit vulnerability: privilege escalation / sandbox escape – integer overflow in kernel code lets you become ‘root’.
- Payload: patch kernel to remove code signing checks.

Locking

- Unlock PIN checked by iOS code before letting you unlock device.

- Also, entire filesystem is encrypted using a key derived from your unlock PIN.
  - But your unlock PIN is only 4 digits long. What attacks does that enable?
  - How could you provide better protection? Discuss.

Takeaways

- Preventing owner of a device from compromising it is hard.
- Jailbreaks are just vulnerabilities.
- Unintended side effect: good guys look for vulnerabilities so they can root their phones; bad guys then use those vulnerabilities to infect people's phones with malware.