Practical Mac OS X Insecurity

Security Concepts, Problems and Exploits on your Mac

Who am I?

- Student of physics, mathematics and astronomy in Bonn
- Mac user since 1995
- I love Macs
- Mac evangelist

Intentions

- Macs have to become even more secure
- Apple needs to react appropriately
- Give an overview on Mac security

Overview

- Mac OS X Security Concepts
- Worst vulnerabilities and exploits
- Securing your Mac

Security Concepts

Security Concepts

- What's good:
- No network services are listening by default
- User is never logged in as root
- Relies on the strong Unix security and permissions model
- Malware usually needs some sort of privilege escalation exploit

Security Concepts

- What's not so good:
- Closed-source security components (e.g. Admin.framework)
- User can do almost anything without entering the password
- Insecure default settings

Vulnerabilities

Vulnerability #1

System Preferences

System Preferences

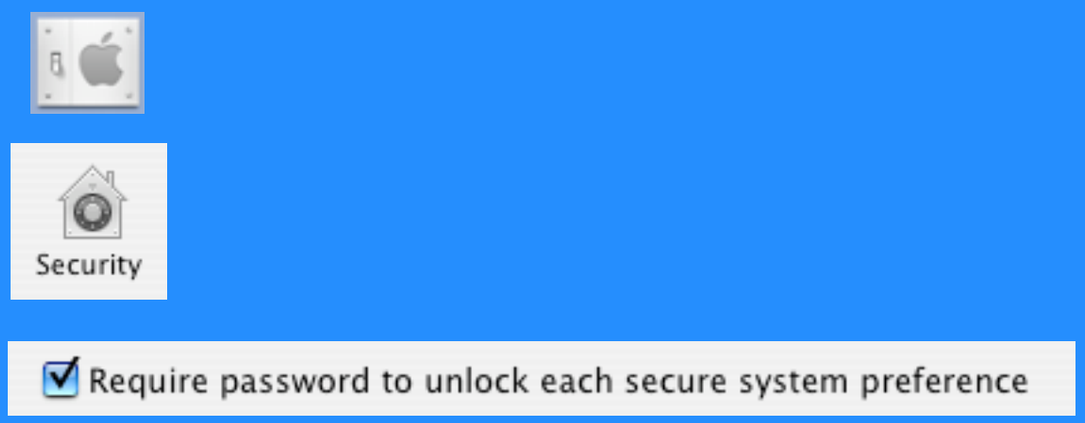
- Closed-source
- Relies on setuid root helper tool
- Many things can be done without password
- Start and stop services, change power management settings, add new users

System Preferences

- You can add users to the admin group
- You can get root within 5 seconds
- Still works in Tiger
- AppleScript Demo!

System Preferences

- How do I fix it?



System Preferences

- I reported this bug to Apple in October
- I sent the ‘exploit’ and suggested a workaround
- The answer was:

  “You can enable the "Require password to unlock each secure system preferences" checkbox, which can be found in the Security preference pane. Note: After doing so, any user will have to provide the admin name and password in order to change settings, including Accounts.”

System Preferences

- This was no question, this was a local privilege escalation vulnerability
- I wrote them again
- This time they answered:

  “Thank you for the clarification. Please know that we have forwarded this information on to the appropriate engineering team for review. “

System Preferences

- The year is now over, it is still not fixed

System Preferences

- This morning, I got mail from Apple
- We are taking security seriously
- Wait for next security update
- Don’t publish it
- Too late, it’s already in the congress proceedings

Vulnerability #2

Bad Installers and wrong Permissions

Bad Installers and wrong Permissions

- Global startup items can be put to /Library/StartupItems
- They will be executed as root

Bad Installers and wrong Permissions

- The directory /Library/StartupItems does not exist by default
- Many installers will create it with wrong permissions
- It will likely be user- or even world-writable

Bad Installers and wrong Permissions

- Execution of arbitrary code as root by putting it in /Library/StartupItems and rebooting

Bad Installers and wrong Permissions

- Disk Utility can repair permissions
- The problem is:
  Disk Utility does not repair the permissions of /Library/StartupItems
- This is a major bug

Bad Installers and wrong Permissions

- How do I fix it?

- Put something like

  ```
  chown -R root:wheel /Library/StartupItems
  chmod -R og-w /Library/StartupItems
  ```

  into `/etc/rc` or `/etc/daily` as a workaround.

Vulnerability #3

Clear Text Passwords in Swap File

Clear Text Passwords

- Apple's Security Framework does not lock passwords in physical memory by using something like mlock()
- Eventually they will be written to the swap file
- Type:
  `sudo strings /var/vm/swapfile0 |grep -A 4 -i longname`
  and you will much likely see your password in clear text

Clear Text Passwords

- Fix?
- In Panther, there is a way
  `http://andreas-s.net/osxencrypted-swap.html`
  but it makes OS X unstable
- Wait for Tiger which lets you encrypt your swap
- But: it's no clean solution, breaks performance

Vulnerability #4

Personal Filesharing Denial of Service

Personal Filesharing Denial of Service

- Personal Filesharing is the Mac equivalent to Samba on Windows
- Contains a guest account by default
- You cannot disable it in System Preferences

Personal Filesharing Denial of Service

- The guest account has write access to a drop box
  - There is no limit on the amount of written data
  - The Harddisk is the limit

Personal Filesharing Denial of Service

- The attacker can fill up all disk space with bullshit
- OS X will likely crash when it has no free swap space anymore
- Especially dangerous over gigabit ethernet
- RendezPoo Demo!

Personal Filesharing Denial of Service

- The guest account can be disabled in `/Library/Preferences/com.apple.AppleFileServer.plist`

by setting

```
<key>guestAccess</key> <true/>
```

to

```
<key>guestAccess</key> <false/>
```

Vulnerability #5

Mach Injection

Mach Injection

- Mach Kernel API
- Inject code into running Programs and execute it
- Override C-Functions at runtime

Mach Injection

- With Objective-C it gets interesting…

Intermezzo: Objective-C Categories

- Categories: Objective-C language feature
  - Add new methods to existing classes at compile time
  - Subclassing without subclassing ;-)

Intermezzo: Objective-C Categories

- Example:
  Extend NSString to be able to deal with regular expressions
  - In your App, create a Category NSString (RegExp)
  - Implement all methods you want to add
  - All NSString are now capable of regular expresions

Intermezzo: Objective-C Categories

- No subclassing and casting necessary
- Also used to split up large classes into a number of smaller files

Mach Injection

- It gets even cooler...
- We have seen what we can do with Objective-C at compile time
- On Mac OS X, you can inject Categories into running Applications at runtime
- I repeat, at runtime!

Mach Injection

- It doesn't stop there...
- The Objective-C Runtime Library lets you replace one method with another
- MethodSwizzling
- I repeat, at runtime!

Mach Injection

- What does it mean?
- You can selectively replace arbitrary methods with your own implementations in any Objective-C App at runtime
- Even possible when you don't have the source code

Mach Injection

- Consequences:
- You can patch the closed-source components in Mac OS X at runtime (e.g. the Finder or Safari)
- You can do almost anything you want with any running App

Mach Injection

- Implications on security:
  - You don’t get a free root shell
  - It makes it easy for malware to patch the system in a malicious way
  - No privileges needed

Mach Injection

- New kind of malware that works entirely with user privileges
- It will be able to patch the system in a malicious way and perhaps even hide itself
- Rootkit without root
- Pretty welcoming atmosphere for “viruses”

Mach Injection

- Implications for security are not yet explored

Vulnerability #6

Disguised Executables

Disguised Executables

- An executable can be camouflaged as any other filetype in the Finder
  - How?
  - Simply change the suffix and the symbol to e.g. match an mp3 file

Disguised Executables

- Fix?
- None known

Summary

1. System Preferences privilege escalation
2. Bad permissions
3. Clear text passwords in swap
4. Personal Filesharing DOS
5. Mach Injection
6. Disguised executables

One more thing…

The first Mac OS X Worm

Proof-of-concept, no malicious routine

The first Mac OS X Worm

Demo!

Resources

- Slides available at:
  https://21c3.annulator.de/
- Demo source code available soon