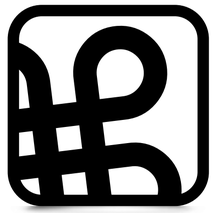


Macoun

Kernel-Hacking, Rootkits
Eine Einführung.

Thomas Tempelmann
tempelmann@gmail.com
www.tempel.org
Twitter: @tempelorg

Ablauf

- Was macht der Kernel
- Was ist ein Rootkit?
- Wie patcht man Funktionen im Kernel?
- Fragen?

Betriebssystem-Kernel

Aufgaben des Kernel

- Verwaltung des Adressraums (Speicher)
- CPU-Zuteilung (Threads)
- Dateisystem
- Zugriffsregelung (User-Verwaltung, root-User)

Kernel-Eigenschaften

- Zugriff auf allen Speicher und CPU-Zeit
- Ausgeführter Code ist unkontrollierbar
- Ein Crash killt das gesamte System

Kernel-Eigenschaften

- Darwin-Kernel-Source (*xnu*) eignet sich nur zum Studieren
- Kann nicht in OS X als Kernel eingesetzt werden (Dateien fehlen)

Kernel-Entwicklung

- Erweiterungen als KEXT laden
- KEXTs benötigen besonderes Codesigning
- Debugging per printf() und OSReportWithBacktrace()
- Häufige Crashes - VM benutzen!
- Remote-Debugging, anyone?

Rootkits

Was ist ein Rootkit?

- Hebelt Einschränkungen des Betriebssystems aus
- Läuft (teilweise) im Kernel
- Versteckt sich vor Entdeckung
- Oft Malware

Ziele von Rootkits

- Kopierschutz
- Überwachung des Benutzers
- Missbrauch des Computers
- Datenspionage

Rootkits entdecken

- Kernel-Extension (KEXT) auflisten mit *kextstat*
- Programmdateien im Dateisystem
- Netzwerkkommunikation (*Little Snitch*)
- Threads
- ...

Rootkits verstecken

- kextstat: *sLoadedKexts* in OSKext.cpp manipulieren
- Dateien: Filterung in *getdirentries(...)*-Hook
- Little Snitch: Socket-Filterliste temporär manipulieren

The Flying Circus

The Flying Circus

- Phrack #69: *fG! - Revisiting Mac OS X Kernel Rootkits*
- Autor: "Reverser" *(*http://reverse.put.as*)*
- Enthält Rootkit-Source für 10.6-10.8, uuencoded

The Flying Circus

- Findet Kernel-Symbole und -Referenzen
- Findet sysent-Tabelle
- Enthält Disassembler
- Ansätze zum Verstecken des Rootkits
- Unvollständig, hat z.T. Probleme auf >= 10.9

sysent-Tabelle

- Sprungbrett vom User-Space in den Kernel (via INT-Instruktion)
- BSD-Funktionen (exit, open, read, fork, chmod, ...)
- xnu: bsd/sys/sysent.h - Strukturen
- bsd/kern/syscalls.master - Listet alle Funktionen
- init_sysent.c (wird durchs xnu-Makefile generiert)

Wir fixen den Kernel

Ein Bug im Kernel

- http://research.swtch.com/macpprof oder http://goo.gl/bBecGy
- Falsch: *psignal(p, SIGPROF)*
- Richtig: *psignal_uthread(current_thread(), SIGPROF)*

Demo

Weiterführende Literatur

Bücher

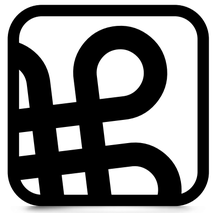
- Amit Singh: OS X Internals - alt aber immer noch gut
- Charlie Miller: The Mac Hacker's Handbook - veraltet
- Jonathan Levin: Mac OS X and iOS Internals

Online

- http://phrack.org/papers/revisiting-mac-os-x-kernel-rootkits.html
- http://reverse.put.as
- https://github.com/gdbinit/hydra (Prozessüberwachung)

Fragen?

Vielen Dank



Macoun