KCKB UPJS

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH

Financované Európskou úniou NextGenerationEU

PLÁN OBNOVY

MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

12. Pokročilá reverzná analýza firmvéru jednoúčelových zariadení

Bezpečnosť vnorených systémov a špecifiká IoT

- **Internet vecí (IoT) a Priemysel 4.0:**
  - Kritická infraštruktúra (SCADA), medicínske zariadenia, Smart Home.
  - Odhad: Miliardy pripojených zariadení s minimálnym dohľadom.
- **Definícia Firmvéru v kontexte bezpečnosti:**
  - Softvér nízkej úrovne (Low-level), ktorý riadi hardvér.
  - Často distribuovaný ako monolitický **BLOB (Binary Large OBject)** bez dokumentácie.
  - Vnímaný ako "Black Box" – proprietárne ovládače a uzavretý kód.
- **Kľúčové bezpečnostné a technické výzvy:**
  - **Hardvérová heterogenita:** Rôzne architektúry (ARM, MIPS, PowerPC, SuperH) a endianita.
  - **Obmedzené zdroje:** Nedostatok miesta pre bezpečnostné mechanizmy (ASLR, Stack Canaries často chýbajú).
  - **Zraniteľnosť:** Masívne využívanie pre botnety (DDoS útoky, rezidenčné proxy siete a IoT Ransomware/Wipery).

Financované Európskou úniou
NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

Statická analýza firmvéru

- **Definícia:** Analýza štruktúry, metadát a obsahu bez samotného vykonávania inštrukcií.
- **Cieľ:** Identifikácia logických komponentov v binárnom bloku (Bootloader, Kernel, Filesystem, Konfigurácia).
- **Nástroj Binwalk:**
  - Štandardný nástroj pre analýzu firmvéru.
  - Funguje na princípe **Magic Bytes** (signatúr) – hľadá známe hlavičky súborov na rôznych offsetoch.
- **Analýza entropie (`binwalk -E`):**
  - Vizuálna reprezentácia náhodnosti dát (Shannonova entropia).
  - **Entropia ~ 0 (Nízka):** Prázdne miesto, padding (nuly), textové reťazce.
  - **Entropia ~ 0.5 - 0.8 (Stredná):** Strojový kód (opcodes procesora majú určitú štruktúru).
  - **Entropia ~ 1.0 (Vysoká):** Kompresia, šifrovanie alebo skutočne náhodné dáta.
- *Poznámka: Binwalk normalizuje Shannonovu entropiu na škálu 0 až 1. V iných nástrojoch sa stretnete so škálou 0 až 8 bitov/bajt, kde kompresia/šifrovanie dosahuje hodnoty nad 7.5.*

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

Extrakcia, "Matrioška" efekt a analýza obsahu

- **Automatizovaná extrakcia (`binwalk -Me`):**
  - Parameter `-M` (Matryoshka): Rekurzívne skenovanie extrahovaných súborov.
  - Parameter `-e` (Extract): Automatické rozbalenie nájdených artefaktov.
  - Príklad reťazca: `Firmware.bin` -> `ZIP` -> `TRX Header` -> `GZIP` -> `CPIO archive`.
- **Manuálna extrakcia (Data Carving):**
  - Keď automatika zlyhá (neznáma kompresia, posunuté hlavičky).
  - Nástroj: `dd` (Linux).
  - Príkaz: `dd if=firmware.bin of=kernel.lzma bs=1 skip=<start_offset> count=<length>`
- **Analýza reťazcov (`strings`):**
  - Rýchle vyhľadanie čitateľného textu (`-n 10` pre filtrovanie šumu).
  - Hľadanie: ciest k súborom (`/bin/sh`), IP adries, hardcoded hesiel (`admin:1234`), kľúčov API.
  - Kódovanie znakov: Nástroj strings predvolene hľadá ASCII. Ak firmvér používa Windows CE alebo iné systémy s Unicode, reťazce sú kódované v 16-bitoch. Nutné použiť strings -el (UTF-16 Little Endian) alebo strings -eb (UTF-16 Big Endian).

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

Súborové systémy v IoT: SquashFS

- **Charakteristika:** Read-only súborový systém, optimalizovaný pre embedded zariadenia. Extrémna kompresia.
- **Štruktúra Superbloku:**
  - Magické číslo: napríklad SquashFS 4.0+ používa magic bytes hsqs (0x68737173) v Little Endian formáte
  - Definuje typ kompresie (GZIP, LZMA, XZ, LZO) a verziu systému.
- **Technické špecifiká:**
  - Inody a metadáta sú tiež komprimované (na rozdiel od bežných FS).
  - **Fragment Table:** Agregácia koncových častí súborov do jedného bloku pre úsporu miesta.
- **Extrakcia a problémy:**
  - Štandard: `unsquashfs`.
  - **Vendor-Specific úpravy:** Výrobcovia (D-Link, TP-Link) často menia magické čísla alebo používajú neštandardné LZMA parametre, aby sťažili RE.
  - Riešenie: Využitie upravených nástrojov ako starší sasquatch alebo moderných forkov squashfs-tools s podporou 'vendor-specific' LZMA kompresií.

Financované Európskou úniou
NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

Súborové systémy pre Flash pamäte: JFFS2 a UBIFS

- **Kontext:** Flash pamäte (NAND/NOR) nefungujú ako bežné disky. Majú obmedzený počet zápisov a vyžadujú špeciálny manažment.
- **JFFS2 (Journalling Flash File System v2):**
  - Určený pre priamy zápis na raw flash.
  - **Log-štruktúrovaný:** Dáta sa nikdy neprepisujú na mieste, ale zapisujú sa sekvenčne na koniec logu.
  - Magické číslo uzlov: `0x1985`.
  - Extrakcia: Nástroj **Jefferson** (python skript pre rekonštrukciu FS v pamäti).
- **UBIFS (Unsorted Block Image File System):**
  - Nástupca JFFS2 pre veľkokapacitné NAND pamäte.
  - Beží nad vrstvou **UBI** (Unsorted Block Images), ktorá rieši mapovanie logických blokov na fyzické, bad blocks a wear-leveling.
  - Kľúčový koncept: **Wandering Tree** (koreňový index sa pri každom zápise presúva).
  - Extrakcia: Nástroj **ubi_reader**.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

Dynamická analýza a koncept Firmware Re-hosting

- **Limitácie statickej analýzy:**
  - Nemožnosť overiť logiku podmienok.
  - Skryté správanie (dynamicky generované kľúče, dešifrovanie konfigurácií za behu).
- **Re-hosting (Emulácia):** Proces prenesenia firmvéru z pôvodného hardvéru do virtuálneho prostredia.
- **Hlavné technické výzvy:**
  - **Chýbajúci hardvér (Hardware Abstraction Layer):** Firmvér sa pokúša komunikovať s čipmi (Wi-Fi, Switch), ktoré v emulátore neexistujú. Prístup na neznámu MMIO adresu (Memory Mapped I/O) spôsobí výnimku (Bus Error / SegFault).
  - **NVRAM závislosť:** Konfigurácia (IP, MAC, heslá) je často uložená v separátnej pamäti, ktorá nie je súčasťou obrazu systému súborov.
  - **Watchdog Timery:** Ak softvér pravidelne "neresetuje" hardvérový časovač, systém sa v emulátore reštartuje alebo zasekne.

Financované Európskou úniou
NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

Emulačné rámce: QEMU a Firmadyne

- **QEMU (Quick Emulator):** Základný emulačný nástoj.
  - **User Mode (`qemu-mips-static`):** Spustenie jednej binárky. Prekladá syscalls cieľovej architektúry na syscalls hostiteľa. Vhodné pre jednoduché binárky (`ls`, `busybox`).
  - **System Mode (`qemu-system-mips`):** Emulácia celého OS, CPU, RAM a periférií. Vyžaduje skompilované jadro a súborový systém.
- **Firmadyne:**
  - Automatizovaný systém postavený nad QEMU System Mode.
  - **Funkcionalita:**
    - Automatická extrakcia FS.
    - Použitie prekompilovaných jadier s inštrumentáciou.
    - "Network Inference" – snaha uhádnuť sieťovú konfiguráciu (VLANs, Bridge).
    - Rieši NVRAM problém pomocou knižnice `libnvram` (zachytávanie volaní).
- Moderné evolúcie Firmadyne: Nástroje ako FirmAE (zvyšuje úspešnosť emulácie pomocou arbitrážnej manipulácie s prostredím) a EMBA (komplexný automatizovaný skener).

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

Pokročilá deterministická emulácia

- **Filozofia Renode:** Zameranie na determinizmus, viditeľnosť a modularitu (na rozdiel od QEMU, ktoré cieli primárne na výkon).
- **Definícia platformy (Infraštruktúra ako kód):**
  - Súbory **.repl** (REnode PLatform): Popis hardvéru. Definuje pamäťové mapy, umiestnenie periférií a prepojenia.
  - Súbory **.resc** (REnode SCript): Skripty na načítanie binárok, nastavenie registrov a spustenie simulácie.
- **Výhody pre reverzné inžinierstvo:**
  - Možnosť "vyskladať" vlastný SoC presne podľa datasheetu (pridať UART na adresu X, Timer na adresu Y).
  - Ľahká implementácia "Stub" periférií v C# alebo Pythone (ak firmvér čaká na odpoveď od senzora, môžeme ju nasimulovať).
  - Integrácia s GDB, Wireshark a Robot Framework.
- Zatiaľ čo QEMU dominuje pri emulácii firmvérov s Linuxom (smerovače), Renode je priemyselným štandardom pre emuláciu 'RTOS' (Real-Time OS) a 'bare-metal' firmvérov (Senzory, Cortex-M, ESP32).

Financované Európskou úniou
NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

Hybridná analýza

- **Motivácia:** Niektoré hardvérové komponenty sú príliš zložité alebo neznáme na to, aby sme ich emulovali (napr. proprietárny šifrovací čip, GSM modem).
- **Riešenie: Avatar2 Framework.**
- **Princíp Orchestrácie:** Spája emulátor (QEMU) so skutočným hardvérom cez JTAG/SWD debug rozhranie.
- **Memory Forwarding (Selektívne presmerovanie):**
  - Kód a bežné inštrukcie bežia rýchlo v QEMU.
  - Keď kód siahne na špecifickú MMIO oblasť (napr. šifrovací kľúč), Avatar2 pozastaví QEMU, prepošle požiadavku do reálneho zariadenia, prečíta odpoveď a vráti ju do emulátora.
- **Výsledok:** Firmvér predpokladá, že beží na reálnom HW, ale je v emulátore. Umožňuje snapshotting a pokročilý debugging.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

Technika podvrhnutia NVRAM (Function Hooking cez LD_PRELOAD)

- **Problém:** Binárka `httpd` (web server) padá, pretože funkcia `nvram_get("lan_ipaddr")` vráti NULL (na PC nemáme NVRAM čip).
- **Riešenie: Function Hooking cez `LD_PRELOAD`.**
- **Mechanizmus:** Dynamický linker v Linuxe uprednostní symboly z knižníc uvedených v premennej prostredia `LD_PRELOAD` pred systémovými knižnicami.
- **Implementácia:**
  - Vytvoríme C kód s vlastnou implementáciou `char *nvram_get(const char *key)`.
  - Vo funkcii načítame hodnoty z lokálneho súboru (`nvram.ini`).
  - Zostavíme ako zdieľanú knižnicu: `gcc -shared -fPIC hook.c -o nvram-faker.so`.
  - Spustenie: `LD_PRELOAD=./nvram-faker.so ./usr/sbin/httpd`
- **Výsledok:** Aplikácia dostane podvrhnuté dáta (napr. heslo, IP) a úspešne sa spustí.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

Analýza IoT malvéru: Prípadová štúdia Mirai

- **Mirai Botnet (2016):** Zneužitie slabých telnet hesiel.
- **Architektúra:**
  - **Loader:** Nahráva binárku na zraniteľné zariadenie.
  - **Bot:** Hlavný payload bežiaci na zariadení.
  - **C2 (Command & Control):** Riadiaci server.
- **Kľúčové techniky v kóde (Reverse Engineering pohľad):**
  - **Obfuskácia reťazcov:** Reťazce sú zašifrované jednoduchým XOR-om (typicky jednobajtový kľúč). Funkcia `table_init` ich dešifruje do RAM pri štarte.
  - **Killer Process:** Agresívne správanie – skenuje bežiace procesy a zabíja konkurenčný malvér (Anime, Qbot) a blokuje porty 22/23 (bráni vzdialenej oprave/zásahu administrátora).
  - **Anti-Debugging:** Pravidelná kontrola `/proc/self/status` na prítomnosť `TracerPid != 0`. Ak zistí debugger, ukončí sa.
- **Evolúcia od Mirai:** Dnešný IoT malvér (napr. Mozi, Gafgyt) využíva P2P siete namiesto centrálneho C2 servera (odolnosť voči "zhodeniu" - nedostupnosti) a využíva One-Day zraniteľnosti (napr. v D-Link/Zyxel routeroch) namiesto jednoduchého telnet brute-forcingu (útok hrubou silou).

Financované Európskou úniou
NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

Detekcia hrozieb: YARA a Syscall Heuristika

- **Statická detekcia: YARA pravidlá:**
  - Definovanie vzoriek (patterns) pre klasifikáciu malvéru.
  - Hľadanie hexadecimálnych sekvencií (unikátne inštrukcie) alebo špecifických reťazcov (napr. cesty k `/bin/busybox` v kombinácii s príkazom `rm`).
  - Príklad: Pravidlo detegujúce XOR slučku Mirai.
- **Dynamická (Behaviorálna) analýza: Syscalls:**
  - Sledovanie volaní jadra (System Calls) počas behu v emulátore.
  - Malvér má špecifický "odtlačok" správania, ktorý je ťažké skryť.
  - **Vzor C2 komunikácie:** Sekvencia `socket` -> `connect` (na neznámu IP) -> `send` -> `recv`.
  - **Vzor skenovania (Syn-Scan):** Tisíce rýchlych, neblokujúcich volaní `connect()` s flagom `O_NONBLOCK` v krátkom čase.
- **Výhoda:** Heuristika funguje aj na "zbalený" (packed) alebo zmenený kód, pokiaľ robí tú istú škodlivú činnosť.

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

Záver

- **Zhrnutie workflowu:**
  - **Akvizícia:** Získanie firmvéru (web výrobcu, extrakcia z Flash čipu, zachytenie update prevádzky).
  - **Dekompozícia:** Binwalk, pochopenie súborového systému (SquashFS/UBIFS).
  - **Emulácia:** Oživenie systému cez QEMU/Renode/Avatar2 (najťažšia časť).
  - **Analýza:** Hľadanie zraniteľností alebo malvéru (Mirai).
- **Etické a právne aspekty:**
  - Reverzné inžinierstvo pre účely interoperability a bezpečnosti je v EÚ často v šedej zóne, ale pre výskum je plne akceptované.
  - Zákaz šírenia exploitov a "weaponizácie" zistení.
  - Zodpovedné zverejňovanie chýb (Responsible Disclosure) výrobcom.
- **Budúcnosť:** Rastúca úloha AI pri hľadaní chýb (Fuzzing) a nástup eBPF pre monitorovanie IoT.

Financované Európskou úniou
NextGenerationEU
PLÁN OBNOVY
MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

KCKB UPJS

UNIVERZITA
PAVLA JOZEFA ŠAFÁRIKA
V KOŠICIACH

Financované
Európskou úniou
NextGenerationEU

PLÁN OBNOVY

MINISTERSTVO
INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA
A INFORMATIZÁCIE
SLOVENSKEJ REPUBLIKY

Ďakujem za pozornosť.