KCKB UPJS

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH

Financované Európskou úniou NextGenerationEU

PLÁN OBNOVY

MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

10. Pokročilé techniky reverznej analýzy kódu a analýzy malvéru v operačnom systéme Windows

PE Formát II.: Voliteľná hlavička a Sekcie

IMAGE_OPTIONAL_HEADER (Kľúčové polia):

- **AddressOfEntryPoint (OEP):** RVA, kde začína prvá inštrukcia programu.
  - *Detekcia packerov:* Ak OEP ukazuje do poslednej sekcie súboru alebo sekcie s názvom, ktorý nie je `.text` (napr. UPX0), je to silný indikátor, že súbor je "zbalený" (packed).
- **ImageBase:** Preferovaná virtuálna adresa načítania (pri 32bit zvyčajne 0x400000 pre exe, 0x10000000 pre dll, pri 64bit zvyčajne 0x140000000 pre exe a 0x180000000 pre dll). Zároveň je dôležité spomenúť, že kvôli ASLR (zapnutom v DllCharacteristics) sa súbory v praxi na tieto preferované adresy takmer nikdy nenačítajú.
- **DllCharacteristics:** Flags pre bezpečnostné funkcie (ASLR, DEP/NX, SEH).

Data Directories (Dátové adresáre):

- **Export Table:** Zoznam funkcií dostupných pre iné programy (typické pre DLL, ale malvér DLL často exportuje len jednu funkciu, napr. `ServiceMain`).
- **Import Address Table (IAT):**
  - Zoznam API funkcií, ktoré program potrebuje.
  - *Analýza:* IAT odhaľuje schopnosti malvéru. Ak vidíte `InternetOpenUrl` a `CreateFile`, malvér pravdepodobne sťahuje súbory. Ak vidíte `CryptDecrypt`, pravdepodobne ide o Ransomware.
  - *Runtime packing:* Packery často IAT vymažú, alebo poškodia a obnovujú ju dynamicky až pri spustení (pomocou `LoadLibrary` a `GetProcAddress`).

Sekcie (Sections):

- `.text` / `.code`: Vykonateľný kód.
- `.data`, `.rdata`: Globálne premenné a konštanty.
- `.rsrc`: Ikonky, menu, ale často aj *uložený zašifrovaný payload* malvéru.
- `.reloc`: Tabuľka relokácií (nevyhnutná pre ASLR).

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

Správa pamäte, VAD a izolácia

Virtuálna pamäť:

- Každý proces má vlastný virtuálny adresný priestor
- **Stránkovanie:** Preklad virtuálnych adries na fyzické zabezpečuje MMU (Memory Management Unit) pomocou tabuliek stránok (Page Tables).

VAD (Virtual Address Descriptor):

- VAD strom je interná štruktúra jadra (binárny vyvážený strom), ktorá popisuje alokované rozsahy pamäte procesu. **Význam pre analytika:**
  - Windows API `VirtualQuery` číta informácie práve z VAD.
  - Ak malvér použije `VirtualAlloc` na injektáž kódu, vznikne vo VAD uzol typu `Private Commit` s ochranou `PAGE_EXECUTE_READWRITE` (RWX).
  - *Detekcia:* Nástroj **Volatility** a plugin `malfind` hľadajú práve tieto anomálie – pamäťové oblasti, ktoré sú spustiteľné (RWX), ale nie sú mapované na žiadny súbor na disku (File backing = None).

Bezpečnostné mechanizmy:

- **ASLR (Address Space Layout Randomization):** Náhodné umiestnenie modulov, heapu a stacku pri každom reštarte, aby sa sťažilo zneužívanie exploitov závislých na pevných adresách.
- **DEP (Data Execution Prevention):** Hardvérový bit (NX - No-Execute), ktorý zabraňuje spusteniu kódu v dátových oblastiach (stack, heap). Malvér to obchádza pomocou ROP (Return-Oriented Programming).

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

User Mode vs. Kernel Mode a Syscalls

Ring 3 (User) vs. Ring 0 (Kernel):

- Aplikácie nemôžu priamo pristupovať k hardvéru. Musia požiadať jadro prostredníctvom systémového volania.

Cesta systémového volania (Flow):

1. Malvér zavolá `CreateFile` (v `kernel32.dll`).
2. `kernel32.dll` zavolá `NtCreateFile` (v `ntdll.dll`).
3. `ntdll.dll` naplní register EAX číslom systémovej služby (SSN - System Service Number) a vykoná inštrukciu `SYSCALL` (x64) alebo `SYSENTER` (x86).
4. Procesor prepne režim do Kernel Mode, kde obslužná rutina (`KiSystemService`) vykoná požiadavku.

Útoky a obchádzanie (Evasion):

- **API Hooking:** Antivírusy (EDR) často vkladajú "hooks" (JMP inštrukcie) do `ntdll.dll` v pamäti procesu, aby monitorovali volania.
- **Direct Syscalls:** Pokročilý malvér (napr. Loader *Dumpert*) si zistí číslo syscallu a zavolá inštrukciu `SYSCALL` priamo vo svojom kóde. Tým úplne obíde `ntdll.dll` a teda aj EDR monitoring (tzv. "Hell's Gate" technika). Súčasné EDR systémy sa adaptovali a dnes kontrolujú pamäť (tzv. syscall stubs). Preto po "Hell's Gate" vznikli moderné evolúcie ako Halo's Gate, ktoré dokážu nájsť SSN aj v prípade, že EDR už prepísalo úvodné inštrukcie v ntdll.dll (unhooking).
- **Heaven's Gate:** Technika spustenia 64-bitového kódu z 32-bitového (WoW64) procesu prepnutím segmentového selektora kódu (CS) z 0x23 na 0x33. Analytické nástroje (debuggery) zamerané na 32-bit kód sa v tomto bode často "stratia".

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

Interné štruktúry procesov (PEB & TEB)

Process Environment Block (PEB):

- Najdôležitejšia User Mode štruktúra pre správu procesu. Adresa uložená v registroch `FS:[30h]` (x86) alebo `GS:[60h]` (x64).
- **Kľúčové polia zneužívané malvérom:**
  - `BeingDebugged` (offset 0x002): Jednoduchý byte, ktorý je 1, ak je proces ladený. Primitívna anti-debug technika.
  - `Ldr` (PEB_LDR_DATA): Obsahuje tri spájané zoznamy (`InLoadOrder`, `InMemoryOrder`, `InInitializationOrder`) všetkých načítaných modulov.
  - *Technika:* Malvér prechádza tieto zoznamy, počíta hash-e názvov modulov (napr. ROR-13 hash) a hľadá `kernel32.dll`. Tým získa adresu API funkcií bez použitia `GetModuleHandle` a `GetProcAddress`, čím skryje svoju prítomnosť v IAT tabuľke.
  - `ProcessParameters`: Obsahuje celú cestu k súboru a *príkazový riadok*. Útočníci tu môžu prepísať dáta, aby nástroje ako Process Explorer zobrazovali falošné argumenty.

Thread Environment Block (TEB):

- Štruktúra pre každé vlákno. Obsahuje informácie o Thread Local Storage (TLS) a Exception Handlingu (SEH reťazec na `FS:[0]`).
- Na 64-bitovej (x64) architektúre sa SEH (Structured Exception Handling) vôbec neukladá do TEB ani na zásobník z dôvodu bezpečnosti (prevencia proti prepísaniu exception handlera na zásobníku). V x64 sa používa tzv. Table-based Exception Handling. Adresy obslužných rutín sú bezpečne uložené priamo v PE hlavičke v adresári .pdata (Exception Directory). Register GS:[0] v x64 neukazuje na SEH reťazec.

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

Techniky injektáže kódu (Process Injection) I.

Motivácia: Skrytie škodlivej aktivity pod legitímny proces (napr. `explorer.exe`, `svchost.exe`), obídenie personálnych firewallov a prístup k pamäti iných procesov.

Klasický DLL Injection:

1. Získanie handle cieľového procesu (`OpenProcess`).
2. Alokácia pamäte pre cestu k DLL (`VirtualAllocEx`).
3. Zápis cesty k DLL (`WriteProcessMemory`).
4. Spustenie vlákna (`CreateRemoteThread`), kde štartovacia adresa je `LoadLibraryA`.

- *Nevýhoda:* DLL musí byť na disku. Dobre detegovateľné pre AV.

Process Hollowing (Dutý proces):

- Technika nahradenia kódu legitímneho procesu pred jeho spustením.

1. `CreateProcess("svchost.exe", ..., CREATE_SUSPENDED)`: Proces vznikne, ale hlavné vlákno stojí.
2. `NtUnmapViewOfSection`: Odstránenie pôvodného kódu `svchost.exe` z pamäte (ak je to potrebné).
3. `VirtualAllocEx`: Alokácia novej pamäte na tej istej báze (často).
4. `WriteProcessMemory`: Rozbalenie a zápis malvéru do alokovaného priestoru (kopírovanie PE hlavičiek a sekcií).
5. `SetThreadContext`: Úprava registra EAX/RCX (Entry Point) hlavného vlákna na nový kód.
6. `ResumeThread`: Spustenie vykonávania malvéru pod menom `svchost.exe`.

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

Techniky injektáže kódu (Process Injection) II.

Pokročilé a Stealth techniky:

- **Process Doppelgänging:**
  - Využíva NTFS Transactional (TxF) API.
  - Útočník vytvorí transakciu, v nej vytvorí súbor s malvérom, vytvorí z neho sekciu v pamäti a potom transakciu *vráti späť* (Rollback).
  - Výsledok: Proces je vytvorený z obsahu pamäte, ktorý na disku nikdy fyzicky neexistoval (pretože transakcia nebola commit - potvrdená). Antivírus, ktorý skenuje súbor pri prístupe, vidí len pôvodný čistý súbor alebo nič.
- **Reflective DLL Injection:**
  - Klasická `LoadLibrary` nevie načítať DLL z pamäte (bufferu), iba z disku.
  - Reflective DLL je špeciálne upravená knižnica, ktorá exportuje vlastnú funkciu `ReflectiveLoader`.
  - Táto funkcia emuluje správanie OS loadera: sama alokuje pamäť, rieši importy, relokácie a zavolá `DllMain`.
  - Umožňuje poslať DLL cez socket a spustiť ju bez uloženia na disk.
- **Atom Bombing:**
  - Je technika na *zápis* (injektáž) škodlivého kódu do pamäte cudzieho procesu bez použitia `WriteProcessMemory`. Využíva na to globálnu tabuľku OS (Global Atom Table) a funkcie ako `GlobalAddAtom` a `NtNtQueueApcThread`.

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

Techniky perzistencie (Persistence)

Cieľ: Zabezpečiť automatické spustenie malvéru po reštarte systému alebo prihlásení používateľa.

Registry Run Keys & Startup:

- `HKCU\Software\Microsoft\Windows\CurrentVersion\Run`
- `HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce`
- *Trik:* Použitie kľúčov, ktoré sú menej známe, napr. `Winlogon\Userinit` alebo `Image File Execution Options` (IFEO) - pripojenie debuggera (malvéru) k legitímnemu programu (napr. `sethc.exe` - Sticky Keys).

WMI Event Subscription (Fileless Persistence):

- Veľmi silná technika, ktorá nezanecháva súbory ani bežné záznamy v registroch.
- **Filter:** Podmienka spustenia (napr. "Uplynulo 2 minúty od štartu" alebo "Prihlásil sa používateľ").
- **Consumer:** Čo sa má vykonať (napr. CommandLineEventConsumer spúšťajúci PowerShell skript zakódovaný v Base64).
- **Binding:** Prepojenie Filtra a Consumera.
- *Analýza:* Bežný `autoruns` túto perzistenciu nemusí vidieť, nutné použiť `Get-WmiObject` alebo špecializované nástroje.

Služby (Services) a Naplánované úlohy (Scheduled Tasks):

- Vytvorenie služby pomocou `sc create` (vyžaduje Admin práva).
- Úlohy cez `schtasks /create` – umožňujú spustiť kód pri rôznych triggroch (napr. pri nečinnosti systému).

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

Eskalácia privilégií a krádež poverení

Manipulácia s prístupovými tokenmi (Token Manipulation):

- Každý proces má priradený Access Token (identita, skupiny, privilégiá).
- **Token Stealing:** Malvér nájde proces bežiaci pod `SYSTEM` (napr. `lsass.exe` alebo `winlogon.exe`), otvorí ho, získa jeho token a "impersonuje" ho (vydáva sa zaň).

UAC Bypass (User Account Control):

- Obídenie dialógového okna "Chcete povoliť tejto aplikácii vykonať zmeny?".
- Techniky zneužívajúce auto-elevate binárky (napr. `fodhelper.exe`, `eventvwr.exe`). Tieto programy pri spustení nekontrolujú UAC, ak sa upraví špecifický kľúč v registroch (napr. `HKCU\Software\Classes\ms-settings\Shell\Open\command`), spustia príkaz útočníka s vysokými právami bez opýtania.

Krádež poverení z LSASS:

- `lsass.exe` je kritický proces spravujúci prihlasovanie.
- Nástroj **Mimikatz** dokáže extrahovať plaintext heslá (ak je povolený WDigest), NTLM hashe a Kerberos tikety. Windows 8.1 a v moderných buildoch Windows 10/11 je WDigest defaultne **zakázaný**, takže plaintext heslá v pamäti nie sú. Útočníci musia najprv upraviť registre (`UseLogonCredential` na 1), počkať na opätovné prihlásenie používateľa (alebo ho k tomu donútiť) a až potom použiť Mimikatz. Dnes sa Mimikatz primárne používa na krádež NTLM hashov (Pass-the-Hash) a Kerberos tiketov (Pass-the-Ticket).
- *Technika:* Malvér získa `SeDebugPrivilege`, otvorí `lsass.exe`, načíta jeho pamäť (`ReadProcessMemory` alebo vytvorí Minidump) a offline ju analyzuje.
- *Obrana:* Credential Guard (virtualizácia LSASS procesu) a RunAsPPL (Protected Process Light).

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

Anti-Analýza a Obfuskácia (Evasion) I.

Anti-Debugging (Detekcia analytika):

- **API metódy:** `IsDebuggerPresent()`, `CheckRemoteDebuggerPresent()`.
- **Manuálne kontroly PEB:** Priamy prístup k `PEB.BeingDebugged` (MOV EAX, FS:[30h]; CMP BYTE PTR [EAX+2], 1).
- **NtGlobalFlag:** Pri spustení z debuggeru má hodnotu 0x70 (zapnuté flagy pre heap checking).
- **Timing Attacks:** Použitie inštrukcie `RDTSC` (Read Time-Stamp Counter) na meranie počtu CPU cyklov medzi dvoma inštrukciami. Ak je rozdiel príliš veľký, znamená to, že analytik kód krokuje alebo beží v emulátore. Moderné analytické hypervízory a sandboxy dokážu inštrukciu RDTSC zachytiť (spôsobí VM-exit) a podvrhnúť malvéru falošný počet cyklov (tzv. RDTSC spoofing), čím túto anti-VM techniku neutralizujú.

Anti-VM a Anti-Sandbox:

- **Hardvérové odtlačky:** Kontrola MAC adresy (00:0C:29 pre VMware), ID disku, rozlíšenie obrazovky, pohyb myši.
- **CPUID inštrukcia:** Volanie `CPUID` s EAX=1 vracia v ECX bit indikujúci hypervízor.
- **Sleep Patching:** Sandboxy často urýchľujú funkciu `Sleep` (preskočia čakanie). Malvér zavolá `Sleep(60000)` a potom skontroluje systémový čas. Ak neubehla reálna minúta, vie, že je v sandboxe.

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

Anti-Analýza a Obfuskácia (Evasion) II.

Kódová obfuskácia (Zahmlievanie):

- **Dead Code:** Vkladanie inštrukcií, ktoré nič nerobia (NOPs, výpočty bez efektu), na zmenu signatúry.
- **Šifrovanie reťazcov:** Všetky texty (IP adresy, názvy súborov) sú zašifrované (XOR, AES) a dešifrujú sa len v momente použitia na stacku (Stack Strings).

Control Flow Flattening (CFF):

- Extrémna technika obfuskácie grafu riadenia (Control Flow Graph).
- Pôvodná hierarchická štruktúra (slučky, if-else) je rozbitá na množstvo malých blokov.
- Všetky bloky sú umiestnené v jednej obrovskej slučke s "switch" príkazom (Dispečer).
- Dispečer na základe stavovej premennej rozhoduje, kam sa skočí ďalej.
- **Výsledok:** V disassembleri (IDA/Ghidra) vyzerá graf ako plochá štruktúra bez logiky, extrémne náročné na pochopenie pre človeka.
- **Opaque Predicates:** Podmienky typu `if (5 > 3) ...`, ktoré sú vždy pravdivé, ale statický analyzátor to nevie a musí analyzovať aj falošnú vetvu, ktorá obsahuje nezmyselný kód.

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

Nástroje pre pokročilú analýzu I.

WinDbg Preview & Time Travel Debugging (TTD):

- Tradičné ladenie je invazívne a "lineárne v čase". Ak chybu preskočíte, musíte reštartovať.
- **TTD princíp:** Zaznamená sa kompletný beh procesu do súboru `.run` (všetky inštrukcie, prístupy do pamäte).
- **Výhody:**
  - Možnosť "krokovať späť" (Reverse Execution).
  - Dátové breakpointy fungujú spätne (Kto zapísal túto hodnotu do pamäte?).
  - Bezpečné: Malvér pri analýze záznamu nebeží, analyzujeme len nahrávku.
- **Data Model (dx):** Výkonný skriptovací jazyk vo WinDbg pre dopytovanie objektov (napr. `dx @$curprocess.Io.Handles.Where(h => h.Type == "File")`).

x64dbg:

- Moderný User-Mode debugger, nástupca OllyDbg.
- Pluginy ako *Scylla* (pre dumpovanie a opravu IAT pri rozbaľovaní malvéru) alebo *Graph View*.

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

Nástroje pre pokročilú analýzu II.

IDA Pro & Ghidra:

- De-facto štandardy pre disassemblovanie a dekompiláciu (preklad do pseudo-C kódu).
- **IDAPython:** Umožňuje písať skripty na automatizáciu rutinných úloh (napr. hromadné dešifrovanie reťazcov, premenovanie funkcií na základe hash-ov).

Symbolické vykonávanie (Symbolic Execution) - ANGR:

- Namiesto konkrétnych hodnôt (napr. int x = 5) používa symboly (int x = lambda).
- Analyzuje program a buduje matematické rovnice (Constraints) pre každú vetvu programu.
- **Použitie:** Nájdenie vstupu, ktorý vedie ku konkrétnemu kódu (napr. "Aké heslo musím zadať, aby som sa dostal do funkcie `Win`?").
- Vhodné na automatické riešenie CrackMe úloh a deobfuskáciu (napr. odstránenie Opaque Predicates).

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

Nástroje pre pokročilú analýzu III.

Volatility Framework 3:

- Analýza plného obrazu RAM (Memory Dump) získaného napr. cez DumpIt.
- Umožňuje vidieť stav systému v čase infekcie bez ovplyvnenia bežiaceho systému.
- **Kľúčové pluginy:**
  - `windows.pslist` / `psscan`: Porovnanie zoznamu procesov (ActiveProcessLinks vs. skenovanie pamäti) na odhalenie DKOM (Direct Kernel Object Manipulation) útokov, kde sa proces skryje, čím nie je viditeľný v zozname úloh.
  - `windows.malfind`: Vyhľadávanie VAD tagov s `PAGE_EXECUTE_READWRITE` ochranou (injektovaný kód).
  - `windows.dlllist` / `ldrmodules`: Odhalenie DLL knižníc, ktoré sú načítané, ale skryté z PEB zoznamov (Module Hollowing).
  - `windows.netscan`: Aktívne sieťové pripojenia (aj tie, ktoré už skončili, ale ostali v pamäti artefakty).

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

Prípadová štúdia: IcedID (BokBot)

- **Typ:** Bankový trojan, ktorý sa vyvinul na Initial Access Broker (poskytuje prístup pre Ransomvér skupiny).
- **Infekčný reťazec (Chain):**
  - Phishing email s prílohou Word (s makrom) alebo zašifrovaným ZIP.
  - Spustenie makra stiahne a spustí HTA / JS súbor.
  - Stiahnutie hlavného payloadu (DLL).
- **Technika: Falošný GZIP Loader.**
  - Payload sa sťahuje zo servera, ktorý sa tvári ako legitímny web.
  - Súbor má GZIP hlavičku, ale nie je to štandardný archív.
  - Malvér obsahuje vlastný dešifrovací algoritmus, ktorý preskočí hlavičku a dešifruje zvyšok.
- **Steganografia:** Ukrývanie šifrovanej konfigurácie a ďalších modulov priamo v pixeloch PNG obrázkov (stegano-payloads).
- **Injekcia:** Používa hookovanie `RtlExitUserProcess` v `svchost.exe` na udržanie perzistencie.

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

Prípadová štúdia - Emotet

- **Charakteristika:** Najznámejší a najodolnejší botnet (opakovane zničený a obnovený).
- **Polymorfizmus:** Server generuje unikátny hash (a niekedy aj štruktúru) binárky pre každú obeť alebo kampaň.
- **Anti-Analýza techniky:**
  - **Junk Code:** Extrémne množstvo zbytočných inštrukcií a slučiek.
  - **Control Flow Flattening:** Rozbitie grafu funkcie na "placku" pomocou obrovského `switch` príkazu, čím sa stráca vizuálna štruktúra logiky.
  - **Dynamické API:** Všetky API volania sú riešené cez hashovanie názvov.
- **Postup infekcie:** Dešifrovanie v pamäti -> Spustenie nového procesu -> Injekcia kódu -> Šírenie primárne cez spamové e-maily (makrá v dokumentoch) a laterálny pohyb cez brute-force SMB a krádež poverení; ako dropper pre sekundárne payloady (TrickBot, Ryuk).

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

Záver

Zhrnutie prednášky:

- Úspešná analýza Windows malvéru vyžaduje syntézu vedomostí o architektúre OS (Kernel, PE, Memory), assembleri a kreatívne myslenie.
- Obrana sa posúva od signatúr k behaviorálnej analýze (EDR) a AI.
- Útočníci reagujú zvyšovaním komplexity (Fileless, Living off the Land, VM-based obfuskácia).

Budúcnosť a trendy:

- AI v malvéri (automatická adaptácia na prostredie).
- Útoky na firmvér (UEFI rootkity) a Supply Chain Attacks.
- Automatizácia analýzy je kľúčom k zvládnutiu objemu hrozieb.

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.