NetLock RMM

The Server Is Owned - But Your Clients Aren’t

Presenter: Nico Mak

Nico Mak

NetLock RMM

29 Jahre alt

17 Jahre Entwicklungserfahrung

Reverse Engineering & Malware Analyse

Entwickelt aktuell das einzige OSS RMM

Standort Köln (Kerpen :^))

Kompromittierte Update Server?

Über 18.000 Organisationen erhielten das kompromittierte Update.

Über 2,2 Millionen Nutzer betroffen.

Über 500.000 Nutzer betroffen.

Über 36.000 Unternehmen betroffen.

Schaden beim Kunden

Viele Unternehmen erlitten erhebliche Betriebsunterbrechungen. Ein prominentes Beispiel ist die schwedische Supermarktkette Coop, die aufgrund des Kaseya Hacks rund 800 Filialen für mehrere Tage schließen musste

Auswirkungen auf das Image

Erheblicher Vertrauensverlust. Das kann zwar jedem passieren, aber ich würde den Hersteller je nach Ursache vermutlich nie wieder in Erwägung ziehen.

Schätzungen nach bis zu 300.000 Systeme weltweit infiziert.

Mein Alptraum

RAM-Only Updates

Wir stellen das neue Update auf eine sichere Art und Weise für den Update Server (einmalig) bereit.

Der Update-Server lädt das verschlüsselte, gesplittete Paket direkt in den Arbeitsspeicher – ohne Zwischenspeicherung.

Der Download-Server hält das Paket im RAM und wartet auf Anfrage vom Client.

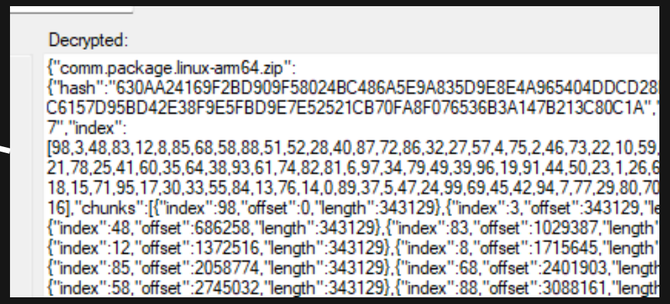
Für jedes Update werden eindeutige, verschlüsselte Metadaten erzeugt, die Informationen zur Verschlüsselung, den Chunk-Größen, der Reihenfolge sowie zur Wiederzusammensetzung enthalten. Das Updatepaket wird bei jeder Anfrage individuell verschlüsselt und fragmentiert.

Fordert die Metadaten an und lädt das verschlüsselte, gesplittete Paket direkt in den RAM herunter.

Entschlüsselt die metadata.json und hält sie im RAM vor.

Das Update-Paket wird gemäß den Metadaten entschlüsselt und rekonstruiert.

Wir führen gegebenenfalls noch weitere Prüfungen durch und starten dann das Update.



Update Server

Anwendung beim Kunden

Code Obfuscation Pt. 1

C# Binary

Vorher in VS

```
_RMM_Web_Console


Application_Paths

static string logs_dir = @"C:\ProgramData\0x101 Cyber Security'
atic string logs_dir = Path.Combine(GetBasePath(), "0x101 Cybe

atic string _private_files_devices = "devices";

atic string internal_dir = Path.Combine(GetCurrentDirectory(),
atic string internal_temp_dir = Path.Combine(GetCurrentDirecto

TART
atic string internal_package_path = Path.Combine(GetCurrentDir
atic string internal_license_info_json_path = Path.Combine(Get
ND

atic string lettuceencrypt_persistent_data_dir = Path.Combine(


tatic string GetBasePath()
```

Identifier Renaming

Variablen-, Methoden- und Klassennamen werden in zufällige, nichtssagende Namen umbenannt (a, b1, x9), um die Logik schwerer nachvollziehbar zu machen.

Control Flow Obfuscation

Der Programmablauf wird so umstrukturiert, dass er komplexer und weniger verständlich erscheint, ohne die Funktionalität zu verändern.

String Encryption

Zeichenketten werden verschlüsselt im Code abgelegt und erst zur Laufzeit entschlüsselt, um statische Analyse zu verhindern.

Code Obfuscation Pt. 2

C# Binary

Nachher in

```
42 RVA: 0x00005350 File Offset: 0x00003550
ns.NoInlining)]

MTaIs4hNg();

_Paths.internal_dir = YRt7aATUyA8E2dauRS1b.lc0TUNjBgFe(Application_Paths.jfV7brMY
9b0cd3cc-4c1c-46f5-8c54-41499dde7858}.m_fed9ef7182804d269ec34b43588d351e.m_1f97f6

_Paths.logs_dir = S7cFeRTIYuGJqudA6Unk.lc0TUNjBgFe(new string[]

tion_Paths.gdk7Q8iOGO(),
8Rybw9d5tpchI.modT8WiPGnC(1941244211 >> 1 ^ 1448573359 ^ <Module>
d3cc-4c1c-46f5-8c54-41499dde7858}.m_fed9ef7182804d269ec34b43588d351e.m_fbf75acdd9
8Rybw9d5tpchI.modT8WiPGnC(~1061363591 ^ -1793232263 ^ <Module>
d3cc-4c1c-46f5-8c54-41499dde7858}.m_fed9ef7182804d269ec34b43588d351e.m_f5bd98f905
8Rybw9d5tpchI.modT8WiPGnC(291616308 - -741392644 ^ 976642409 ^ <Module>
d3cc-4c1c-46f5-8c54-41499dde7858}.m_fed9ef7182804d269ec34b43588d351e.m_308af0e534
```

Dummy Code Insertion

Nutzlose Codeblöcke werden eingefügt, um den Code umfangreicher und schwerer lesbar zu machen.

Opaque Predicates

Logische Bedingungen, deren Ergebnis zur Laufzeit immer gleich ist, aber für den Leser komplex und unvorhersehbar erscheinen.

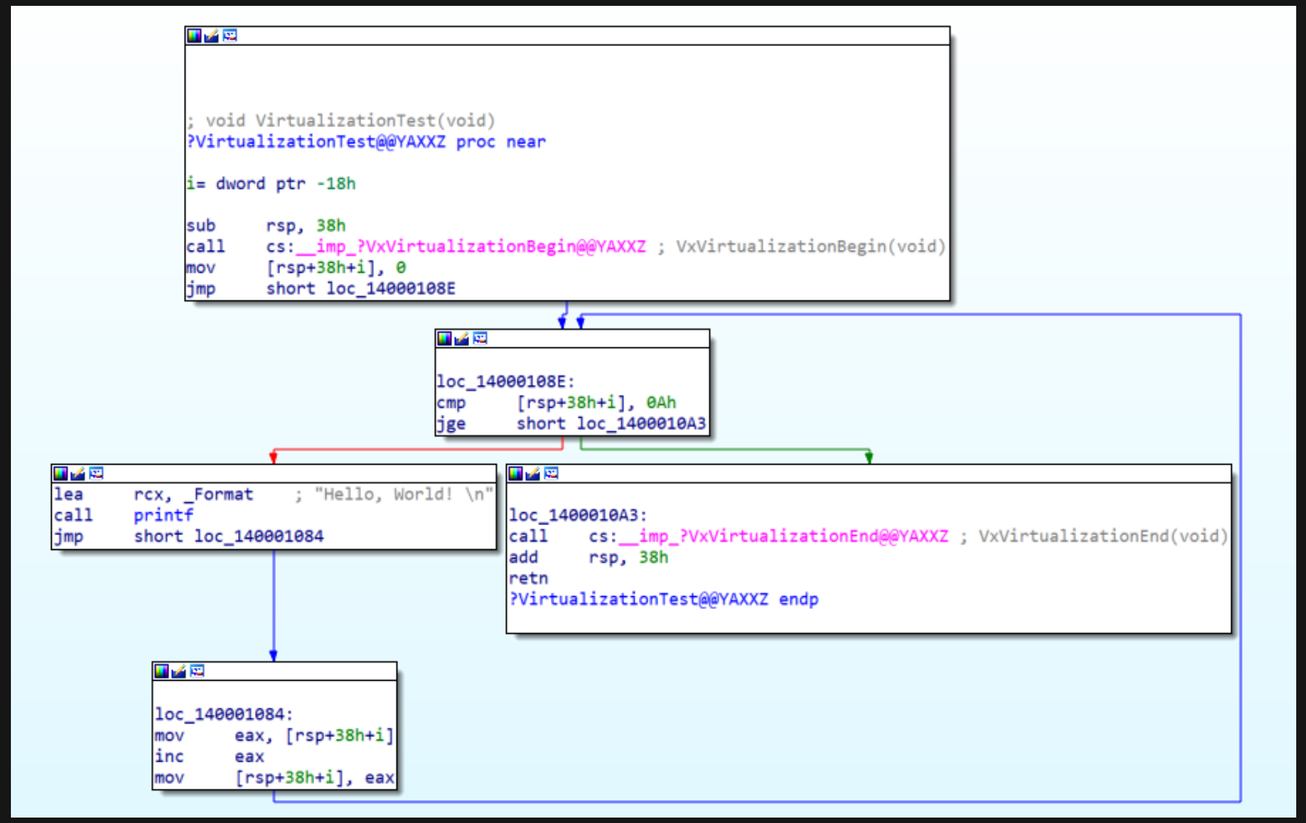
Inlining und Outlining

Methoden werden entweder vollständig in den Aufrufer integriert (Inlining) oder in kleine, schwer nachvollziehbare Teilstücke zerlegt (Outlining).

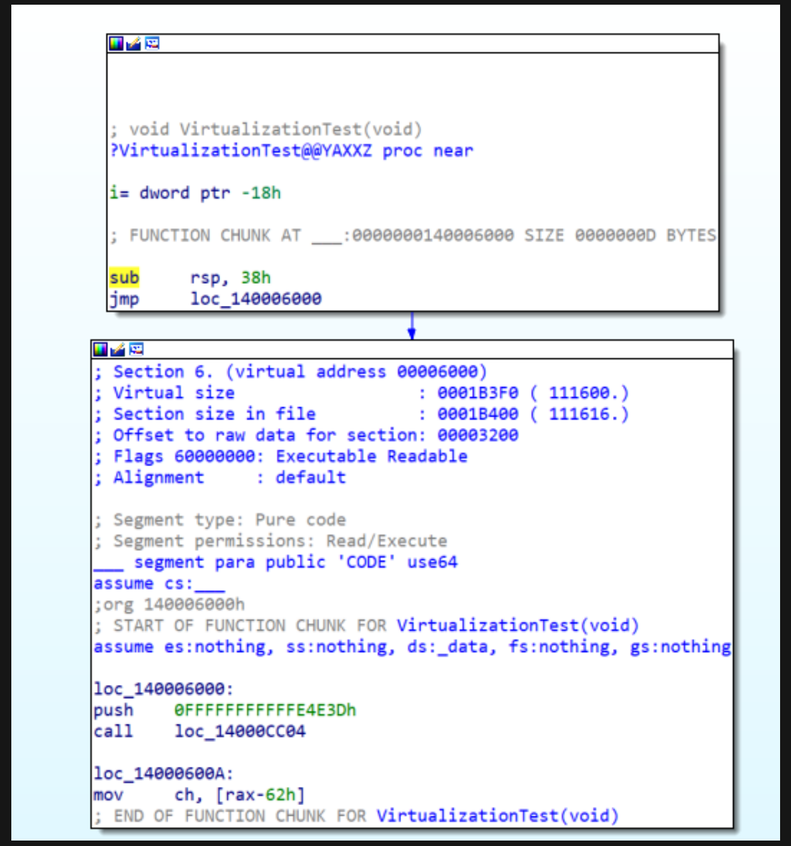
Code Virtualization Pt. 1

- Erschwert sowohl statische wie auch dynamische Analysen.
- Debugging Tools lesen nur abstrakte VM Instructions.
- Manipulationen des Prozess Speichers sehr schwer.

Vorher in IDA



Nachher in IDA



Code Virtualization Pt. 2

```
; Virtualized Anti-Tampering Mechanism

; Initialize Virtual Machine State
movi    vm_state, 0x1A2B3C4D            ; Initial VM state signature

; Integrity Check - Verify Bytecode
movi    r0, [bytecode_start]            ; Start address of the virtualized bytecode
movi    r1, [bytecode_end]              ; End address of the virtualized bytecode
hash    r2, r0, r1                      ; Generate a hash of the bytecode
cmpr    r2, 0xDEADBEEF                  ; Compare with the expected hash
jne     tamper_detected                 ; Jump if hash does not match

; Timing Check - Ensure no manipulation delays
movi    r3, [timestamp]                 ; Get current timestamp
wait    0x10                            ; Wait for 16 cycles
movi    r4, [timestamp]                 ; Get the timestamp again
subr    r4, r3                          ; Calculate the time difference
cmpr    r4, 0x10                        ; Expected time difference
jne     tamper_detected                 ; Jump if timing was manipulated

; VM State Check - Ensure virtual state integrity
movi    r5, [vm_state]                  ; Load the VM's current state
cmpr    r5, 0x1A2B3C4D                  ; Compare with the original state
jne     tamper_detected                 ; Jump if state is altered

; Main Logic (If all checks pass)
movr    r6, [protected_memory]          ; Load protected data
movr    rax, r6                         ; Store in rax for further use
jmp     main_execution                  ; Continue with main program logic

; Tampering Detected - Trigger response
label   tamper_detected
movi    rax, 0xFFFFFFFF                 ; Set error code
halt                                     ; Terminate execution
```

Integrity Check

Überprüft den Bytecode auf Manipulationen mittels Hashvergleich.

Timing Check

Erkennt Verzögerungen, die durch Debugger oder Hooking entstehen könnten.

VM State Check

Verifiziert den Zustand der virtuellen Maschine, um Manipulationen zu erkennen.

Tamper Response

Bei Erkennung eines Angriffs wird das Programm sofort terminiert. Wir könnten auch Code triggern, der uns die Manipulation mitteilt.

Wie geht das noch besser?

- Lockt die Angreifer in eine Falle.
- Überwacht bestimmte Verzeichnisse. Fake Update?

- Aktiv Network Sniffer, Prozess- und Dateisystem-Scanner, Debugger und andere Forensik-Werkzeuge erkennen.

- System isolieren: Das betroffene System sicher vom Netzwerk trennen, um weiteren Schaden zu verhindern.

- Erstellt wenn möglich automatisiert ein Abbild des Servers, um es später analysieren und den Angriff nachvollziehen zu können.

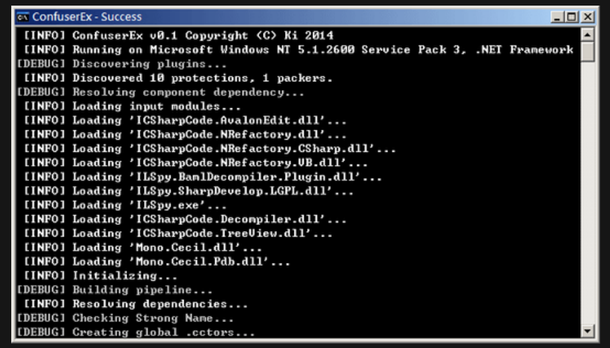
Welche Lösungen gibt es?

“Dieser Vortrag wird gesponsort von {brandName}VPN.”

(C, C++, Delphi / Pascal, Assembly (x86/x64))

.NET, VB??? (nicht so gut)

(C#, VB.NET, ASP.NET, Winforms etc.)



Open Source

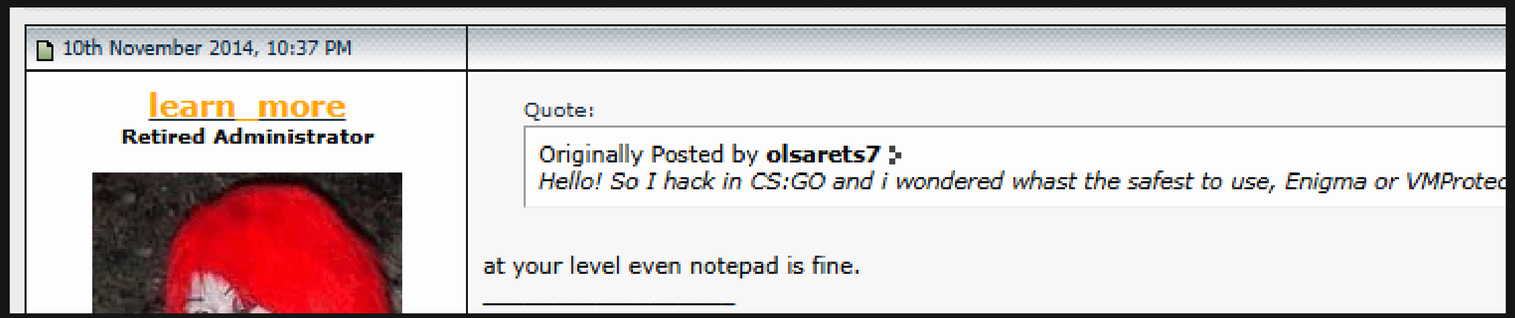
(Gar nicht mal so geil)

Schattenseiten?

- Kann die Performance massiv negativ beeinflussen.
- Es ist extrem wichtig die Implementation mit Sorgfalt durchzuführen. Schlechte Implementation = kein Schutz.

Denuvo und VMProtect angeblich schuld an hoher CPU-Last in Assassin's Creed Origins

Nach über drei Monaten haben Cracker das besonders komplizierte Kopierschutzsystem von Assassin's Creed Origins ausgehebelt.



Fragen?

NetLock RMM

NetLock RMM ist eine Open-Source-Remote-Monitoring- und Management-Software (RMM) mit dem Schwerpunkt auf Sicherheit und Transparenz.

335+
Companies already trust NetLock RMM.

170+
Users joined our Discord already.

120+
GitHub commits since release.

Worldwide Real-Time Remote Access to Your Devices: Stay Connected Anywhere

netlockrmm.com