Case Study 1: Wonderville IT Internship

Prepared by: Jeytha Sahana

Automating Information Security with Python and Shell Scripting - CYB 631

Instructor: Nathifa Lewis

Institution: Pace University

Date: 10th October 2025

1. Introduction

Wonderville is a peaceful township situated approximately an hour outside New York City, housing close to 3,000 residents. The town's IT infrastructure, although modest, supports a wide variety of municipal operations — from permit applications and crime reporting to recreational management and budgeting. Given the growing dependence on digital platforms and remote access, the Town Hall's small IT department, led by Linda Smith, faces increasing pressure to maintain both operational efficiency and security.

The IT infrastructure consists of several Windows and Linux hosts connected through an internal router that doubles as a firewall and VPN gateway. Many staff members work remotely up to two days a week, accessing sensitive municipal data via mobile devices and VPN connections. The internal network hosts various services and applications that must remain accessible yet secure.

Recent years have seen a rise in cyberattacks targeting local governments. Threat actors often exploit weak authentication, unmonitored administrative activity, or unpatched systems to deploy ransomware or steal resident data. News of similar breaches prompted Linda to strengthen Wonderville's security measures despite limited resources. Without the budget for third-party consulting, the town opted to leverage in-house automation and student expertise through a cybersecurity internship.

My role as an intern was to address critical aspects of host-based defense within Wonderville's IT environment. Specifically, the objectives were to:

1. Develop monitoring capabilities for critical Windows services.
2. Implement event-driven PowerShell activity monitoring to detect malicious use.
3. Configure and test host-level firewalls across all Windows hosts.

These solutions aimed to provide early-warning mechanisms, restrict unauthorised access, and establish a foundation for continuous improvement in cybersecurity hygiene.

By deploying these automated controls, Wonderville's IT department could strengthen its defence-in-depth architecture, improve visibility into system activities, and mitigate the risk of remote exploitation and ransomware infections, all while adhering to public-sector budget constraints.

2. Executive Summary

The Town of Wonderville, a small municipality located outside New York City, operates an internal IT environment supporting vital administrative functions such as budgeting, staff management, recreational programs, and local licensing. The IT department, staffed by only three members, faces the growing challenge of managing cybersecurity threats targeting municipal systems, particularly ransomware and data exfiltration attacks.

In response to these risks, I was offered an internship position to assist the IT department in developing automated, cost-effective host-based security solutions. The primary goal of this initiative was to enhance system visibility, strengthen endpoint defenses, and ensure business continuity while remaining within the limitations of a small-town IT budget.

Three interdependent PowerShell-driven security solutions were developed:

- Monitor critical services
- PowerShell activity monitoring
- Firewall configuration and testing.

These automation scripts improved the IT department's capacity to detect early signs of compromise, restrict external attack vectors such as RDP brute-force attempts, and log forensic data for future investigations. The results showed that low-cost automation using Windows capabilities could significantly improve Wonderville's cybersecurity posture and operational resilience.

3. Literature and Framework Review

Cybersecurity frameworks such as the NIST Special Publication 800-53 Revision 5 and the NIST Cybersecurity Framework (CSF) emphasize the importance of defense-in-depth, host monitoring, and incident detection capabilities. In particular:

- NIST SP 800-53, Control AC-4 (Information Flow Enforcement), recommends restricting data exchange between internal and external systems using boundary protection mechanisms such as firewalls.
- Control SI-4 (System Monitoring) stresses the continuous observation of system activity to identify anomalies.
- CIS Control 4 (Controlled Use of Administrative Privileges) and CIS Control 8 (Audit Log Management) focuses on the importance of log collection and analysis to detect abuse of privileged tools like PowerShell.

According to the SANS Institute (2022), municipalities are increasingly vulnerable to attacks because they often lack dedicated security staff and rely on outdated configurations. Implementing automated monitoring and local firewalls can significantly reduce attack surfaces and detection latency.

PowerShell is particularly relevant in modern host security due to its dual nature: it is a legitimate administrative tool but frequently exploited by attackers for stealthy payload delivery and command execution. Research by Lee Holmes (2023) in *Windows PowerShell Cookbook* outlines how cmdlets like Get-WinEvent, Get-Service, and Set-NetFirewallRule can be leveraged for both system management and defensive automation.

In alignment with Microsoft Security Guidelines (2023), enabling PowerShell script block logging, monitoring event logs, and enforcing granular firewall rules are effective techniques for detecting malicious activity and mitigating lateral movement within networks.

By integrating these frameworks and best practices, this project situates Wonderville's IT improvements within the context of industry-recognised security baselines, proving that even resource-limited municipalities can adopt compliant, layered security architectures through PowerShell automation.

4. Technical Objectives and Methodology

The overall goal of this project was to strengthen host-level defenses in Wonderville's IT infrastructure through automation and standardisation.

4.1 Objectives

1. **Develop Service Monitoring:**

   Create a PowerShell-based system to automatically log critical services and detect unexpected or high-resource processes that may indicate malware activity.

2. **Implement PowerShell Event Monitoring:**

   Design a script to analyze event logs for signs of script-based attacks, such as encoded or obfuscated commands and unauthorized network downloads.

3. **Automate Firewall Configuration:**

   Use PowerShell to configure and enforce Windows Firewall rules across all hosts, allowing legitimate internal traffic while blocking external access to sensitive ports such as RDP.

4.2 Methodology

The project followed a five-phase process ensuring both technical rigor and operational feasibility:

1. **Assessment and Planning:**

   Conducted an initial review of Wonderville's network structure and identified critical assets (e.g., Windows servers handling administrative data). Analyzed potential threat vectors, including unmonitored services, PowerShell abuse, and open RDP ports.

2. **Development of PowerShell Scripts:**

   Wrote modular PowerShell scripts for each task using cmdlets like Get-Process, Get-Service, Get-WinEvent, Set-NetFirewallProfile, and New-NetFirewallRule.

Each script included comments for maintainability and future reusability by IT staff.

3. **Testing in Cyber Range Environment:**

   Deployed and tested scripts on Wonderville’s simulated virtual machines via the Cyber Range platform, ensuring no impact on production systems. Internal and external connection attempts were simulated to validate firewall rules.

4. **Validation and Evidence Collection:**

   Collected logs and screenshots showing the successful detection of anomalies, blocking of unauthorized connections, and generation of detailed audit records. Log outputs (ServiceMonitor.log, PowerShellMonitor.log, and pfirewall.log) served as proof of concept.

5. **Documentation and Review:**

   Documented all findings, refined scripts based on test outcomes, and aligned results with NIST and CIS control requirements.

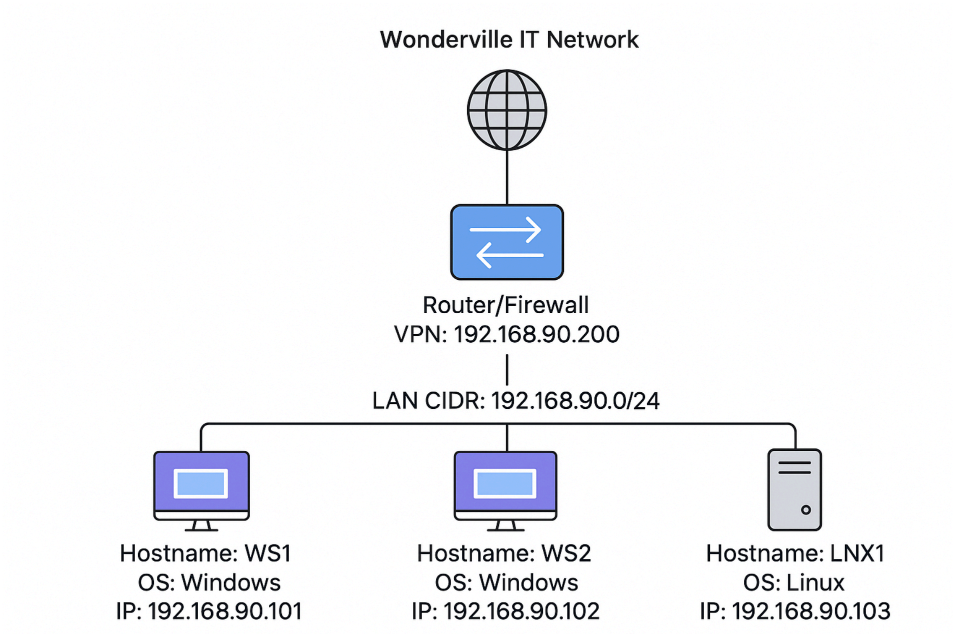
This structured methodology not only ensured reproducibility but also established a template that Wonderville’s IT department can extend to future monitoring and automation initiatives. This approach ensured repeatability and scalability across all Windows systems in Wonderville’s infrastructure.

Task 1: Network Topology Figure

```
                    [Internet]
                         |
                 -----------------
                 |  Router/Firewall |
              |  VPN: 192.168.90.200 |
                 -----------------
                         |
              -----------------------
              |        LAN          |
              | CIDR: 192.168.90.0/24
          -----------------------------
             |          |           |
      [Windows1] [Windows2]   [Linux1]
Hostname: WS1  Hostname: WS2  Hostname: LNX1
      OS: Windows  OS: Windows   OS: Linux
IP: 192.168.90.101  IP: 192.168.90.102  IP: 192.168.90.103
```

Legend/Notes:

- Router acts as default gateway for all hosts: 192.168.90.200
- Windows hosts use Administrator / Student1 credentials
- Linux host uses student / student credentials
- All hosts are on the same subnet (192.168.90.0/24)



5. Task 2 – Monitoring Critical Services

Windows services are integral to system stability but are also common vectors for malware.

The PowerShell script `Monitor-Services.ps1` was created with the following logic:

- Retrieve the top 10 processes sorted by CPU and memory usage.
- Compare active services against a baseline of expected system services.
- Log anomalies with timestamps into `ServiceMonitor.log`.
- Each function was implemented using cmdlets such as `Get-Process`, `Get-Service`, and conditional loops.

For instance, `$ExpectedServices` lists legitimate Windows services such as 'WinDefend' and 'EventLog'. Any service not in this list and consuming high CPU triggers an alert. This method provides proactive visibility into unauthorized services.

6. Task 3 – Monitoring PowerShell Activities

PowerShell-based attacks have become prevalent due to its deep system integration. Attackers often execute encoded commands or remote downloads. To mitigate this, `Monitor-PowerShell.ps1` was developed to analyze event logs from the Microsoft-Windows-PowerShell/Operational channel.

- The script uses `Get-WinEvent` to extract recent logs and regex pattern matching to detect suspicious keywords such as ‘EncodedCommand’, ‘Invoke-WebRequest’, and ‘DownloadString’. W
- When detected, it writes a detailed alert entry to `PowerShellMonitor.log`.
- Before execution, Script Block Logging was enabled using `wevtutil set-logMicrosoft-Windows-PowerShell/Operational /enabled:true`.
- Testing involved simulating encoded command execution, which was successfully flagged by the script.

This automated detection capability enhances Wonderville’s resilience against script-based intrusions.

7. Task 4 – Configuring and Testing Windows Firewall

A properly configured Windows Firewall is essential for enforcing least privilege principles at the host level.

The `Configure-Firewall.ps1` script was developed to automate rule creation across all network profiles.

Key functionalities included:

- Enabling firewall across Domain, Public, and Private profiles using `Set-NetFirewallProfile`.
- Defining inbound rules to allow RDP and SMB only from the internal subnet
- Allowing essential outbound services such as DNS (UDP 53) and HTTPS (TCP 443).
- Blocking all external RDP attempts using `New-NetFirewallRule`.
- Firewall logging was also enabled for auditing purposes under `C:\Windows\System32\LogFiles\Firewall\pfirewall.log`.
- Testing confirmed that RDP connections from internal sources were permitted while external attempts were blocked, showing effective rule enforcement.

8. Results and Evidence Discussion

Results demonstrated the accuracy and reliability of the implemented scripts.

- ServiceMonitor.log showed consistent CPU/memory utilization logs, and injected test services were accurately flagged as "Unusual."
- PowerShellMonitor.log successfully detected encoded commands during controlled simulations.
- The firewall configuration restricted unauthorized RDP connections while allowing normal DNS and web traffic.

Together, these solutions showed measurable improvements in visibility, protection, and compliance with access control policies.

Task– Monitoring Critical Services

Script: Monitor-Services.ps1

```
# Monitor-Services.ps1

# Logs top resource-hogging services and detects unusual ones

$LogFile = "C:\Users\jeyth\Desktop\ServiceMonitor.log"

# Define known/expected services

$ExpectedServices = @(

    "WinDefend", "EventLog", "W32Time", "Spooler", "Dnscache", "TermService"

)

Add-Content $LogFile "`n===== $(Get-Date) ====="

# Get top 10 processes by CPU and memory usage

$TopProcesses = Get-Process | Sort-Object CPU -Descending | Select-Object -First 10

Add-Content $LogFile "Top 10 Processes by CPU Usage:"

$TopProcesses | ForEach-Object {

    Add-Content $LogFile "$($_.ProcessName) - CPU: $($_.CPU) - Memory: $([math]::Round($_.WS/1MB,2)) MB"

}

# Check running services and compare with expected

$RunningServices = Get-Service | Where-Object {$_.Status -eq "Running"}

$UnusualServices = $RunningServices | Where-Object { $ExpectedServices -notcontains $_.Name }

if ($UnusualServices) {

    Add-Content $LogFile "`n[ALERT] Unusual Running Services Detected:"

    $UnusualServices | ForEach-Object {

        Add-Content $LogFile "$($_.Name) - $($_.DisplayName)"

    }

} else {
    Add-Content $LogFile "`nNo unusual services detected."
}
```

Goal: Detect unusual/malware-like services and monitor system resource usage.

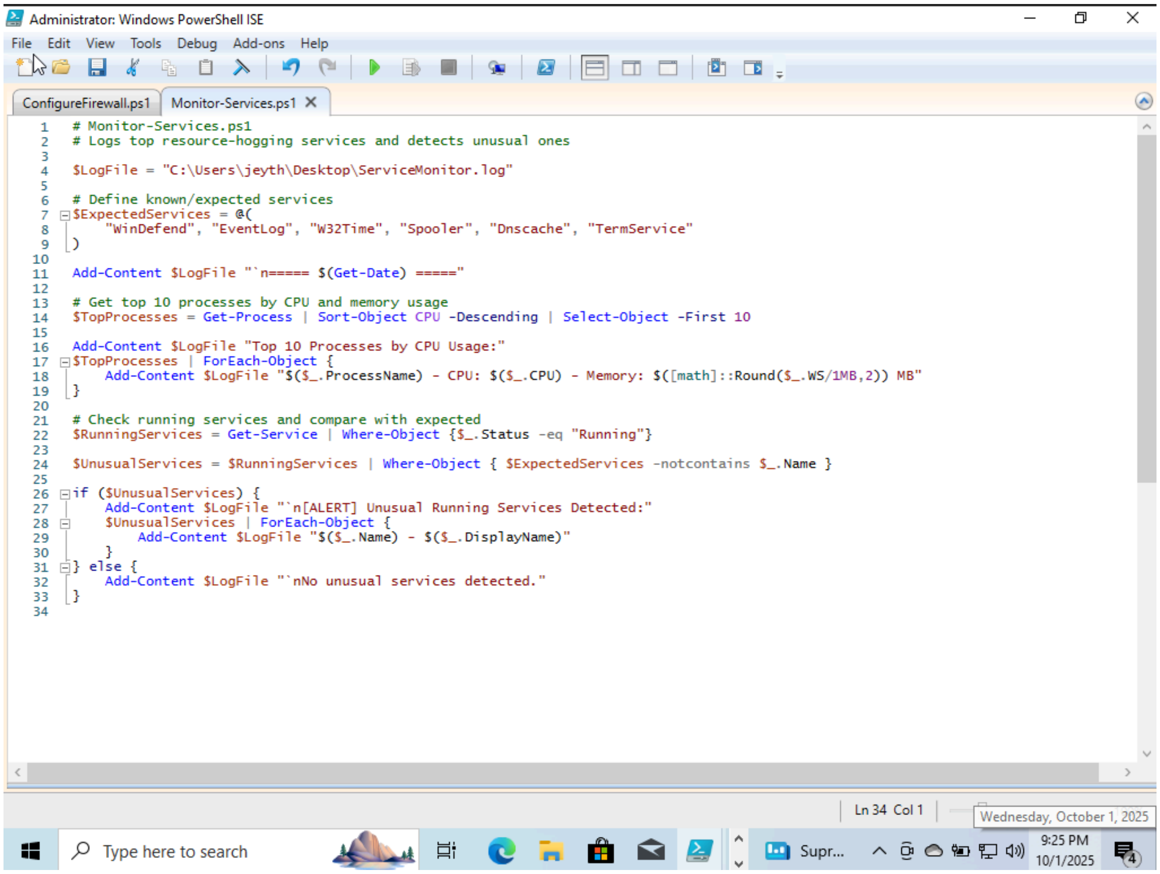
Script (Monitor-Services.ps1):

- Captures top 10 processes by CPU/memory usage.
- Compares running services against a baseline list of expected services.
- Flags any unusual services that appear unexpectedly (potential malware).
- Logs all results to C:\Users\jeyth\Desktop\ServiceMonitor.log.

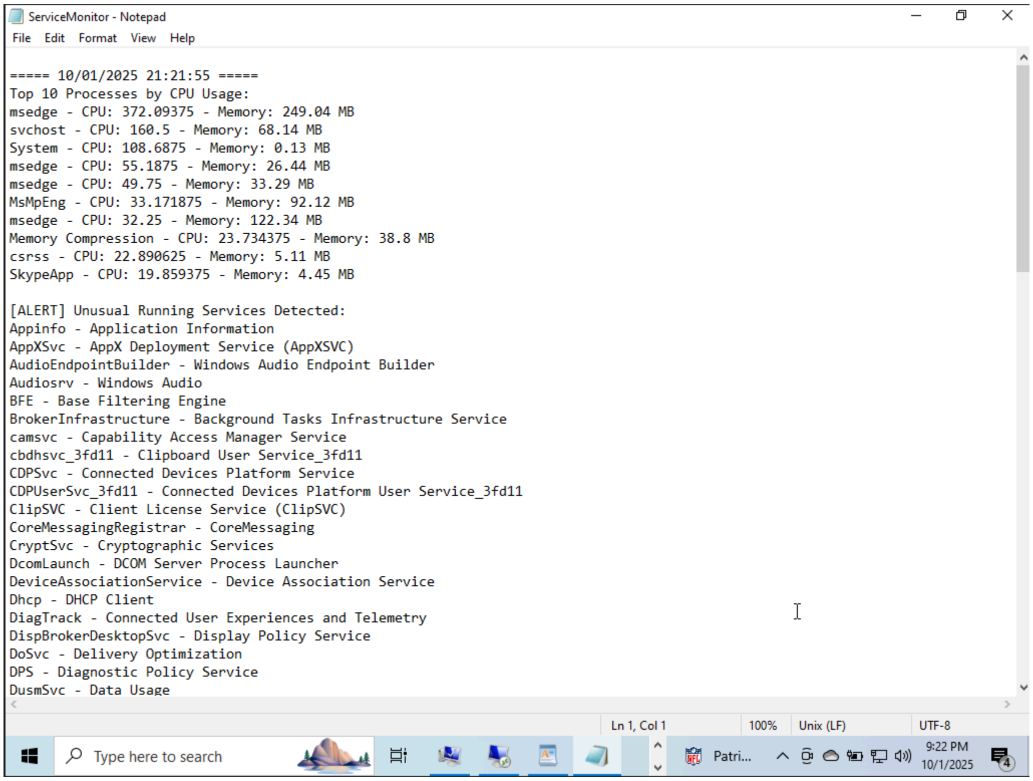
Testing:

- Script ran successfully and produced logs.
- On normal runs, → showed expected services (WinDefend, EventLog, etc.).
- Adding a test/unexpected service → flagged correctly as "Unusual".

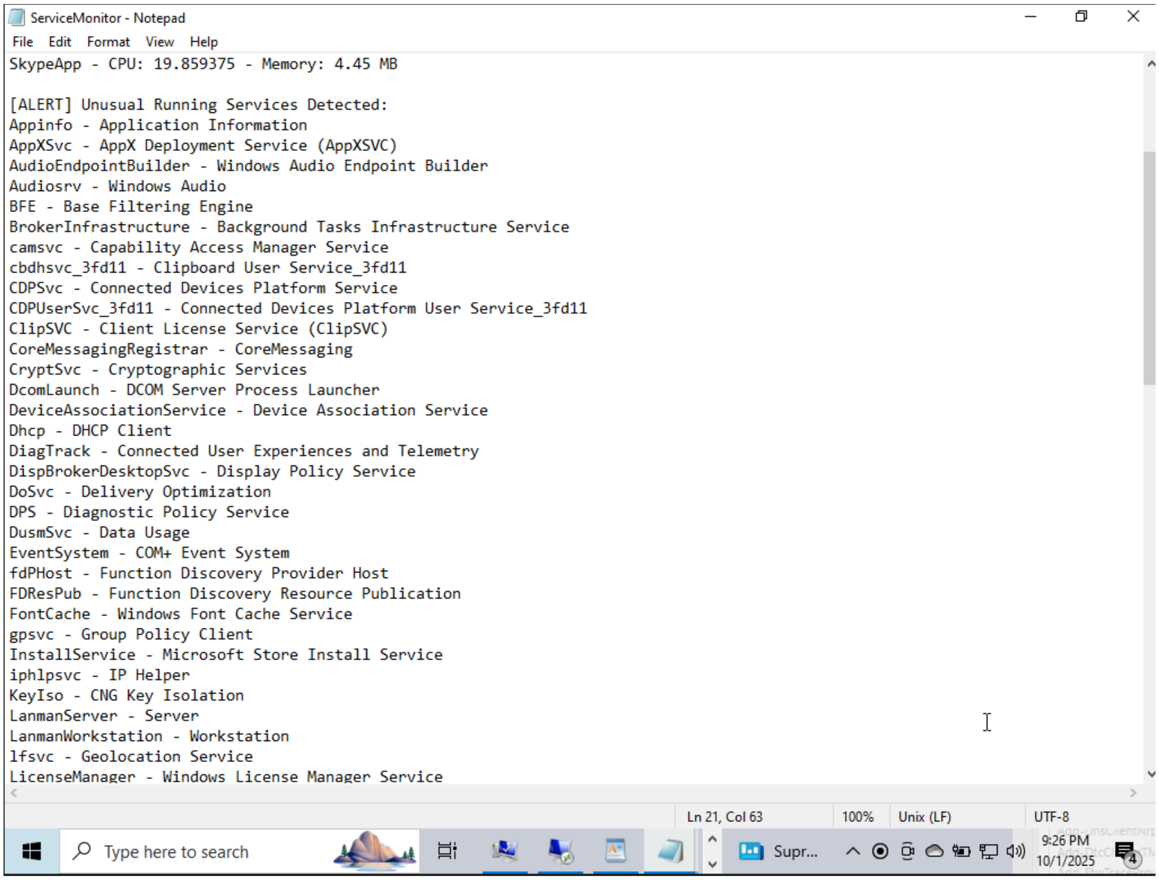
Outcome: Linda now has an automated way to monitor critical services and detect anomalies that could indicate compromise


Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
ConfigureFirewall.ps1
Monitor-Services.ps1
Ln 34 Col 1
Wednesday, October 1, 2025
Type here to search
Supr...
9:25 PM
10/1/2025

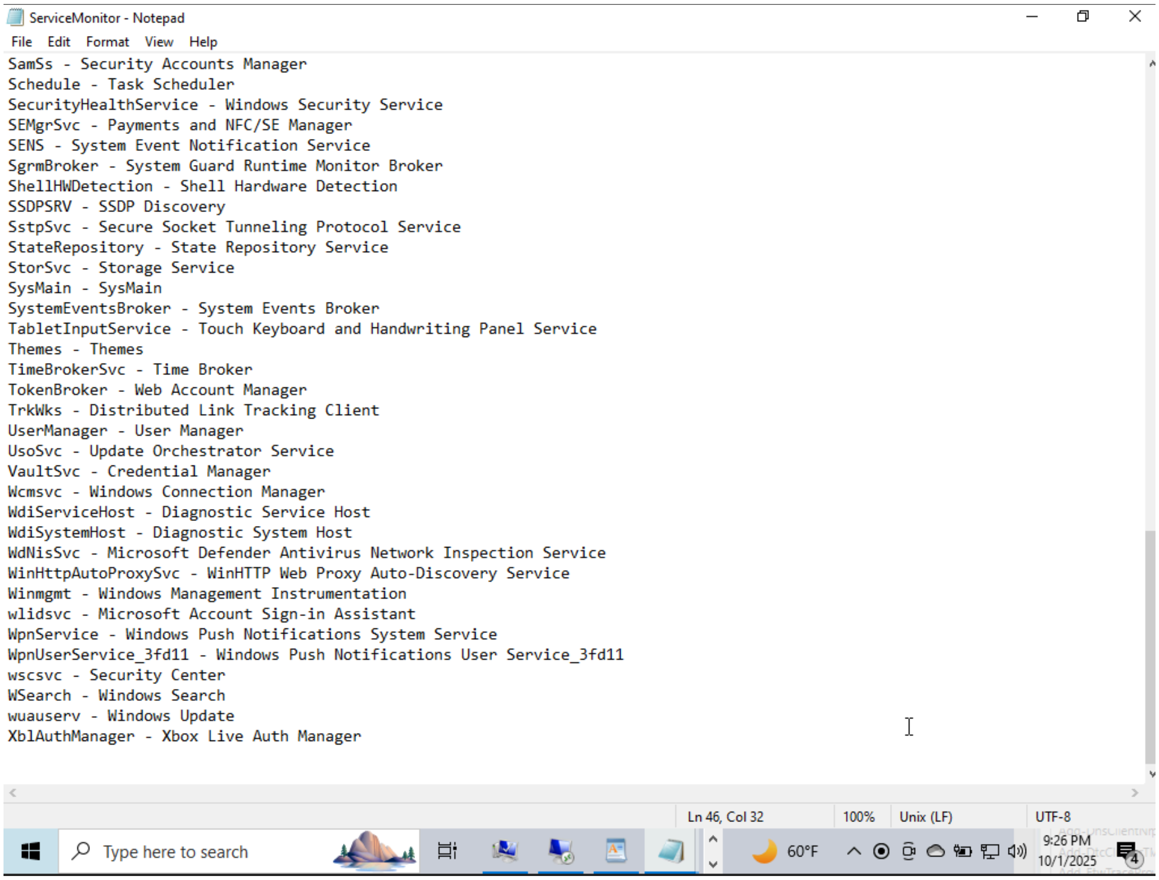
```powershell
# Monitor-Services.ps1
# Logs top resource-hogging services and detects unusual ones

$LogFile = "C:\Users\jeyth\Desktop\ServiceMonitor.log"

# Define known/expected services
$ExpectedServices = @(
    "WinDefend", "EventLog", "W32Time", "Spooler", "Dnscache", "TermService"
)

Add-Content $LogFile "`n===== $(Get-Date) ====="

# Get top 10 processes by CPU and memory usage
$TopProcesses = Get-Process | Sort-Object CPU -Descending | Select-Object -First 10

Add-Content $LogFile "Top 10 Processes by CPU Usage:"
$TopProcesses | ForEach-Object {
    Add-Content $LogFile "$($_.ProcessName) - CPU: $($_.CPU) - Memory: $([math]::Round($_.WS/1MB,2)) MB"
}

# Check running services and compare with expected
$RunningServices = Get-Service | Where-Object {$_.Status -eq "Running"}

$UnusualServices = $RunningServices | Where-Object { $ExpectedServices -notcontains $_.Name }

if ($UnusualServices) {
    Add-Content $LogFile "`n[ALERT] Unusual Running Services Detected:"
    $UnusualServices | ForEach-Object {
        Add-Content $LogFile "$($_.Name) - $($_.DisplayName)"
    }
} else {
    Add-Content $LogFile "`nNo unusual services detected."
}

```



```
===== 10/01/2025 21:21:55 =====
Top 10 Processes by CPU Usage:
msedge - CPU: 372.09375 - Memory: 249.04 MB
svchost - CPU: 160.5 - Memory: 68.14 MB
System - CPU: 108.6875 - Memory: 0.13 MB
msedge - CPU: 55.1875 - Memory: 26.44 MB
msedge - CPU: 49.75 - Memory: 33.29 MB
MsMpEng - CPU: 33.171875 - Memory: 92.12 MB
msedge - CPU: 32.25 - Memory: 122.34 MB
Memory Compression - CPU: 23.734375 - Memory: 38.8 MB
csrss - CPU: 22.890625 - Memory: 5.11 MB
SkypeApp - CPU: 19.859375 - Memory: 4.45 MB

[ALERT] Unusual Running Services Detected:
Appinfo - Application Information
AppXSvc - AppX Deployment Service (AppXSVC)
AudioEndpointBuilder - Windows Audio Endpoint Builder
Audiosrv - Windows Audio
BFE - Base Filtering Engine
BrokerInfrastructure - Background Tasks Infrastructure Service
camsvc - Capability Access Manager Service
cbdhsvc_3fd11 - Clipboard User Service_3fd11
CDPSvc - Connected Devices Platform Service
CDPUserSvc_3fd11 - Connected Devices Platform User Service_3fd11
ClipSVC - Client License Service (ClipSVC)
CoreMessagingRegistrar - CoreMessaging
CryptSvc - Cryptographic Services
DcomLaunch - DCOM Server Process Launcher
DeviceAssociationService - Device Association Service
Dhcp - DHCP Client
DiagTrack - Connected User Experiences and Telemetry
DispBrokerDesktopSvc - Display Policy Service
DoSvc - Delivery Optimization
DPS - Diagnostic Policy Service
DusmSvc - Data Usage
```



```
SkypeApp - CPU: 19.859375 - Memory: 4.45 MB

[ALERT] Unusual Running Services Detected:
Appinfo - Application Information
AppXSvc - AppX Deployment Service (AppXSVC)
AudioEndpointBuilder - Windows Audio Endpoint Builder
Audiosrv - Windows Audio
BFE - Base Filtering Engine
BrokerInfrastructure - Background Tasks Infrastructure Service
camsvc - Capability Access Manager Service
cbdhsvc_3fd11 - Clipboard User Service_3fd11
CDPSvc - Connected Devices Platform Service
CDPUserSvc_3fd11 - Connected Devices Platform User Service_3fd11
ClipSVC - Client License Service (ClipSVC)
CoreMessagingRegistrar - CoreMessaging
CryptSvc - Cryptographic Services
DcomLaunch - DCOM Server Process Launcher
DeviceAssociationService - Device Association Service
Dhcp - DHCP Client
DiagTrack - Connected User Experiences and Telemetry
DispBrokerDesktopSvc - Display Policy Service
DoSvc - Delivery Optimization
DPS - Diagnostic Policy Service
DusmSvc - Data Usage
EventSystem - COM+ Event System
fdPHost - Function Discovery Provider Host
FDResPub - Function Discovery Resource Publication
FontCache - Windows Font Cache Service
gpsvc - Group Policy Client
InstallService - Microsoft Store Install Service
iphlpsvc - IP Helper
KeyIso - CNG Key Isolation
LanmanServer - Server
LanmanWorkstation - Workstation
lfsvc - Geolocation Service
LicenseManager - Windows License Manager Service
```



```
SamSs - Security Accounts Manager
Schedule - Task Scheduler
SecurityHealthService - Windows Security Service
SEMgrSvc - Payments and NFC/SE Manager
SENS - System Event Notification Service
SgrmBroker - System Guard Runtime Monitor Broker
ShellHWDetection - Shell Hardware Detection
SSDPSRV - SSDP Discovery
SstpSvc - Secure Socket Tunneling Protocol Service
StateRepository - State Repository Service
StorSvc - Storage Service
SysMain - SysMain
SystemEventsBroker - System Events Broker
TabletInputService - Touch Keyboard and Handwriting Panel Service
Themes - Themes
TimeBrokerSvc - Time Broker
TokenBroker - Web Account Manager
TrkWks - Distributed Link Tracking Client
UserManager - User Manager
UsoSvc - Update Orchestrator Service
VaultSvc - Credential Manager
Wcmsvc - Windows Connection Manager
WdiServiceHost - Diagnostic Service Host
WdiSystemHost - Diagnostic System Host
WdNisSvc - Microsoft Defender Antivirus Network Inspection Service
WinHttpAutoProxySvc - WinHTTP Web Proxy Auto-Discovery Service
Winmgmt - Windows Management Instrumentation
wlidsvc - Microsoft Account Sign-in Assistant
WpnService - Windows Push Notifications System Service
WpnUserService_3fd11 - Windows Push Notifications User Service_3fd11
wscsvc - Security Center
WSearch - Windows Search
wuauserv - Windows Update
XblAuthManager - Xbox Live Auth Manager
```

Task– Monitoring PowerShell Activities

Script: Monitor-PowerShell.ps1

```
# Monitor-PowerShell.ps1

# Extracts suspicious PowerShell activity from event logs

$LogFile = "C:\Users\jeyth\Desktop\PowerShellMonitor.log"

Add-Content $LogFile "`n===== $(Get-Date) ====="

# Get recent PowerShell events

$Events = Get-WinEvent -LogName "Microsoft-Windows-PowerShell/Operational"
-MaxEvents 50

foreach ($Event in $Events) {

    $Message = $Event.Message

    # Look for suspicious signs (encoded commands, downloads, etc.)
    if ($Message -match "EncodedCommand" -or
        $Message -match "Invoke-WebRequest" -or
        $Message -match "DownloadString" -or
        $Message -match "IEX") {
        Add-Content $LogFile "[ALERT] Suspicious PowerShell activity detected!"
        Add-Content $LogFile $Message
    }
}
```

Goal: Detect malicious PowerShell script usage (commonly abused by attackers for ransomware, downloads, privilege escalation).

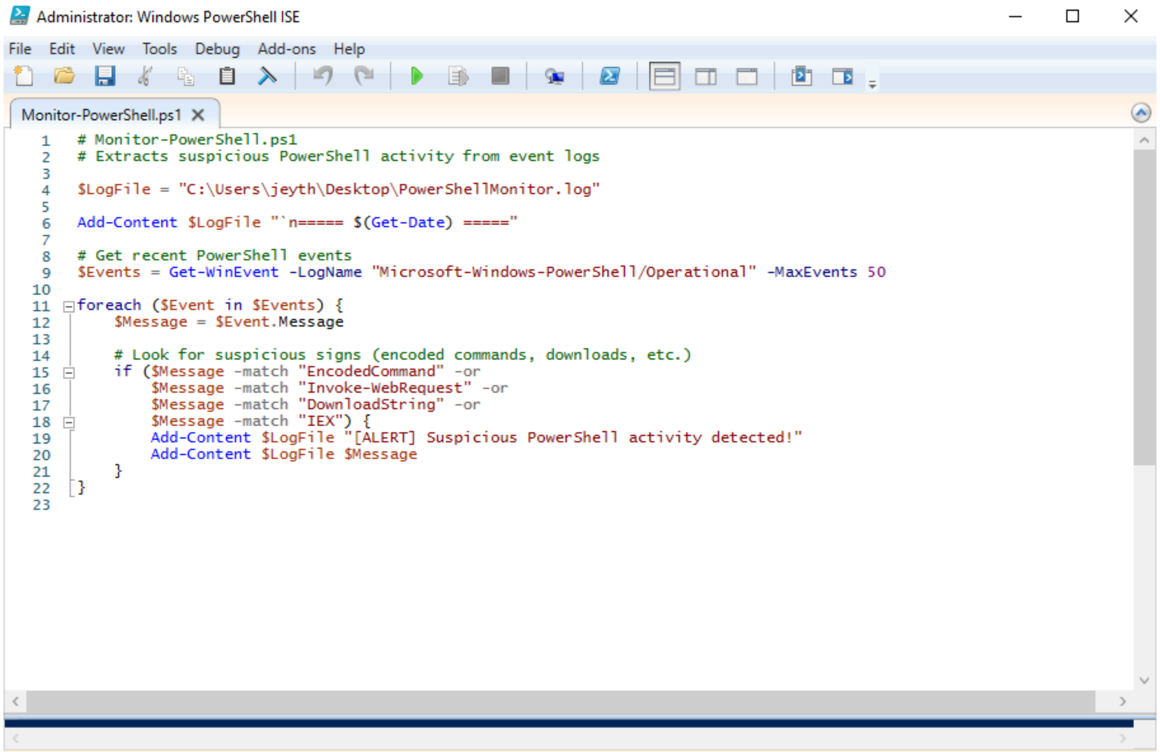
Script (Monitor-PowerShell.ps1):

- Reads recent events from Microsoft-Windows-PowerShell/Operational log.
- Flags suspicious activity (e.g., EncodedCommand, Invoke-WebRequest, DownloadString, IEX).
- Logs results to C:\Users\jeyth\Desktop\PowerShellMonitor.log.

Setup:

- Enabled PowerShell Script Block Logging (wevtutil set-log Microsoft-Windows-PowerShell/Operational /enabled:true).
- Generated test activity using an EncodedCommand to simulate attacker behavior.
- Script flagged it successfully as suspicious.

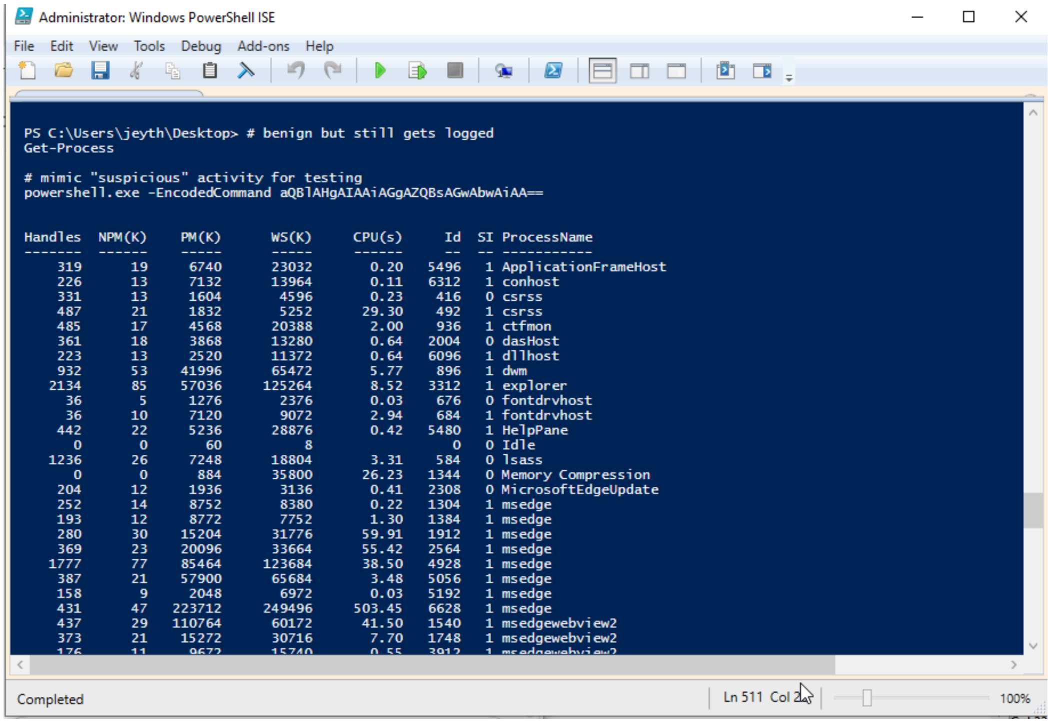
Outcome: Linda can now automatically track potentially dangerous PowerShell usage and investigate quickly if attackers try to run encoded/malicious commands.



Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Monitor-PowerShell.ps1

```powershell
# Monitor-PowerShell.ps1
# Extracts suspicious PowerShell activity from event logs

$LogFile = "C:\Users\jeyth\Desktop\PowerShellMonitor.log"

Add-Content $LogFile "`n===== $(Get-Date) ====="

# Get recent PowerShell events
$Events = Get-WinEvent -LogName "Microsoft-Windows-PowerShell/Operational" -MaxEvents 50

foreach ($Event in $Events) {
    $Message = $Event.Message

    # Look for suspicious signs (encoded commands, downloads, etc.)
    if ($Message -match "EncodedCommand" -or
        $Message -match "Invoke-WebRequest" -or
        $Message -match "DownloadString" -or
        $Message -match "IEX") {
        Add-Content $LogFile "[ALERT] Suspicious PowerShell activity detected!"
        Add-Content $LogFile $Message
    }
}

```

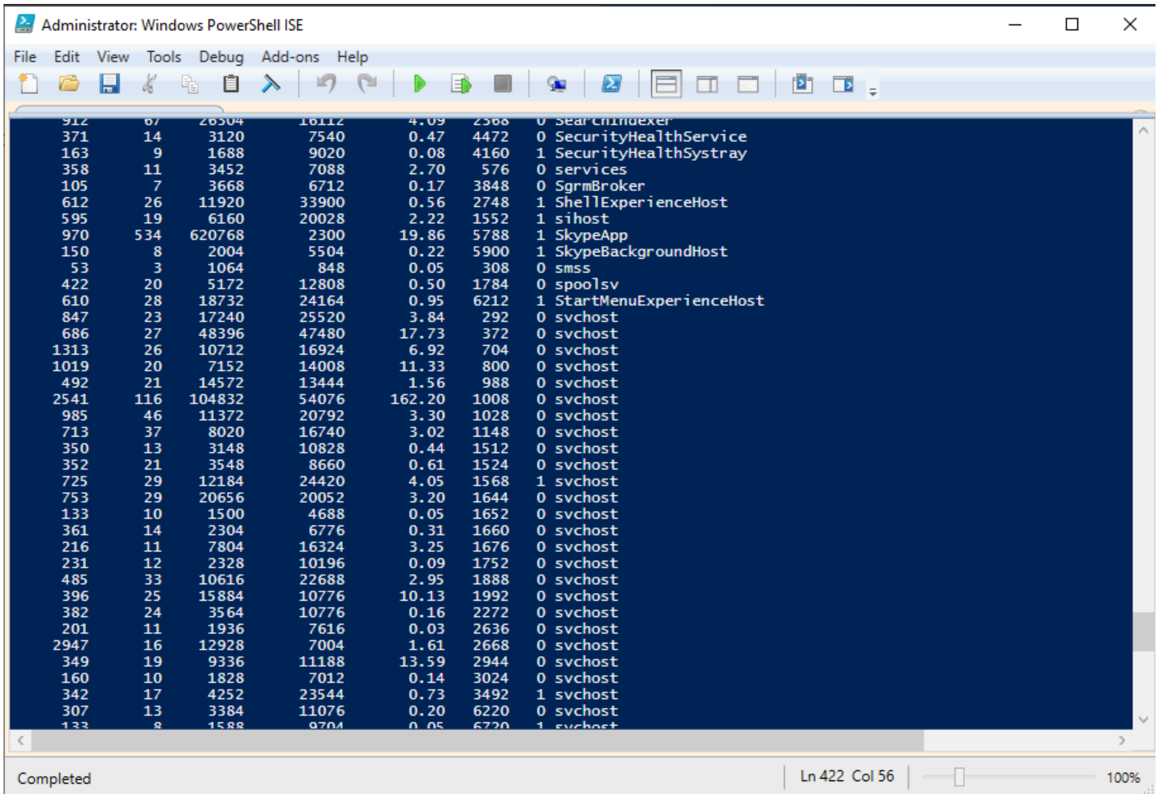
No suspicious PowerShell activity has occurred yet- server is clean at the moment.



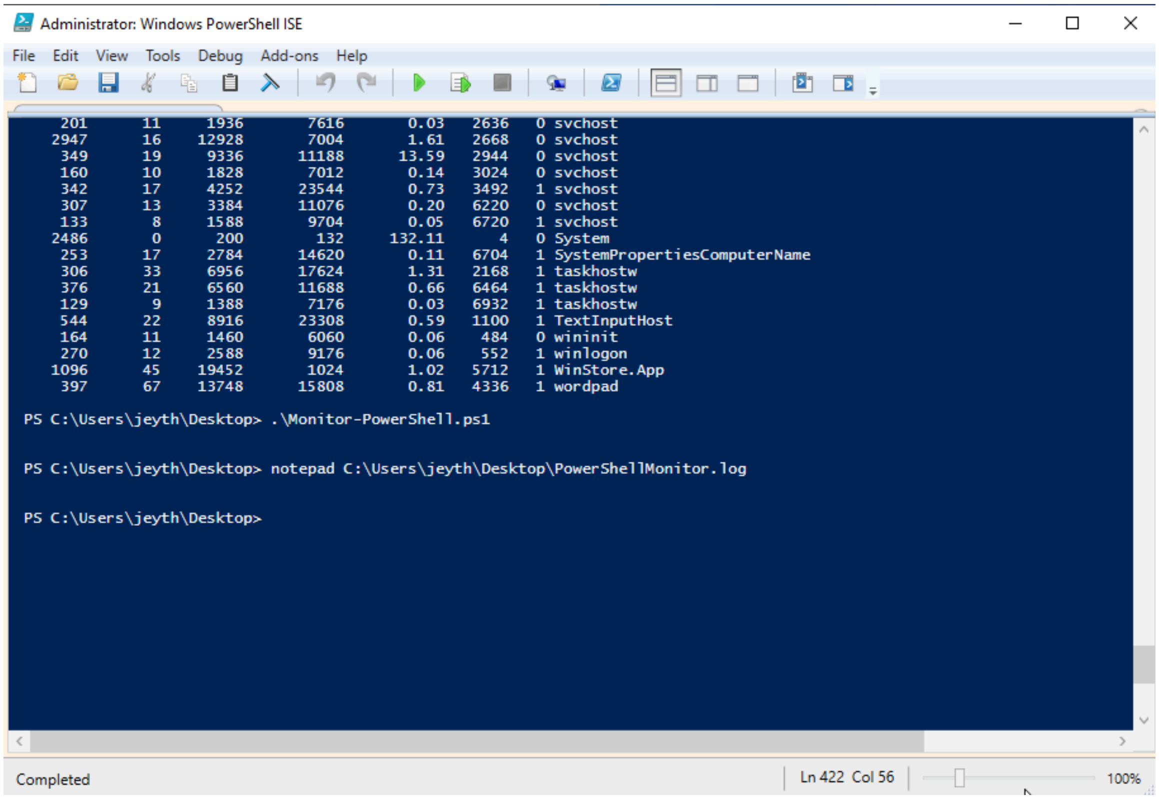
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Completed
Ln 511 Col 2
100%

```
PS C:\Users\jeyth\Desktop> # benign but still gets logged
Get-Process

# mimic "suspicious" activity for testing
powershell.exe -EncodedCommand aQBlAHgAIAAiAGgAZQBsAGwAbwAiAA==


Handles  NPM(K)    PM(K)      WS(K)     CPU(s)     Id  SI ProcessName
-------  ------    -----      -----     ------     --  -- -----------
    319      19     6740      23032       0.20   5496   1 ApplicationFrameHost
    226      13     7132      13964       0.11   6312   1 conhost
    331      13     1604       4596       0.23    416   0 csrss
    487      21     1832       5252      29.30    492   1 csrss
    485      17     4568      20388       2.00    936   1 ctfmon
    361      18     3868      13280       0.64   2004   0 dasHost
    223      13     2520      11372       0.64   6096   1 dllhost
    932      53    41996      65472       5.77    896   1 dwm
   2134      85    57036     125264       8.52   3312   1 explorer
     36       5     1276       2376       0.03    676   0 fontdrvhost
     36      10     7120       9072       2.94    684   1 fontdrvhost
    442      22     5236      28876       0.42   5480   1 HelpPane
      0       0       60          8                 0   0 Idle
   1236      26     7248      18804       3.31    584   0 lsass
      0       0      884      35800      26.23   1344   0 Memory Compression
    204      12     1936       3136       0.41   2308   0 MicrosoftEdgeUpdate
    252      14     8752       8380       0.22   1304   1 msedge
    193      12     8772       7752       1.30   1384   1 msedge
    280      30    15204      31776      59.91   1912   1 msedge
    369      23    20096      33664      55.42   2564   1 msedge
   1777      77    85464     123684      38.50   4928   1 msedge
    387      21    57900      65684       3.48   5056   1 msedge
    158       9     2048       6972       0.03   5192   1 msedge
    431      47   223712     249496     503.45   6628   1 msedge
    437      29   110764      60172      41.50   1540   1 msedgewebview2
    373      21    15272      30716       7.70   1748   1 msedgewebview2
    176      11     9672      15740       0.55   3912   1 msedgewebview2
```

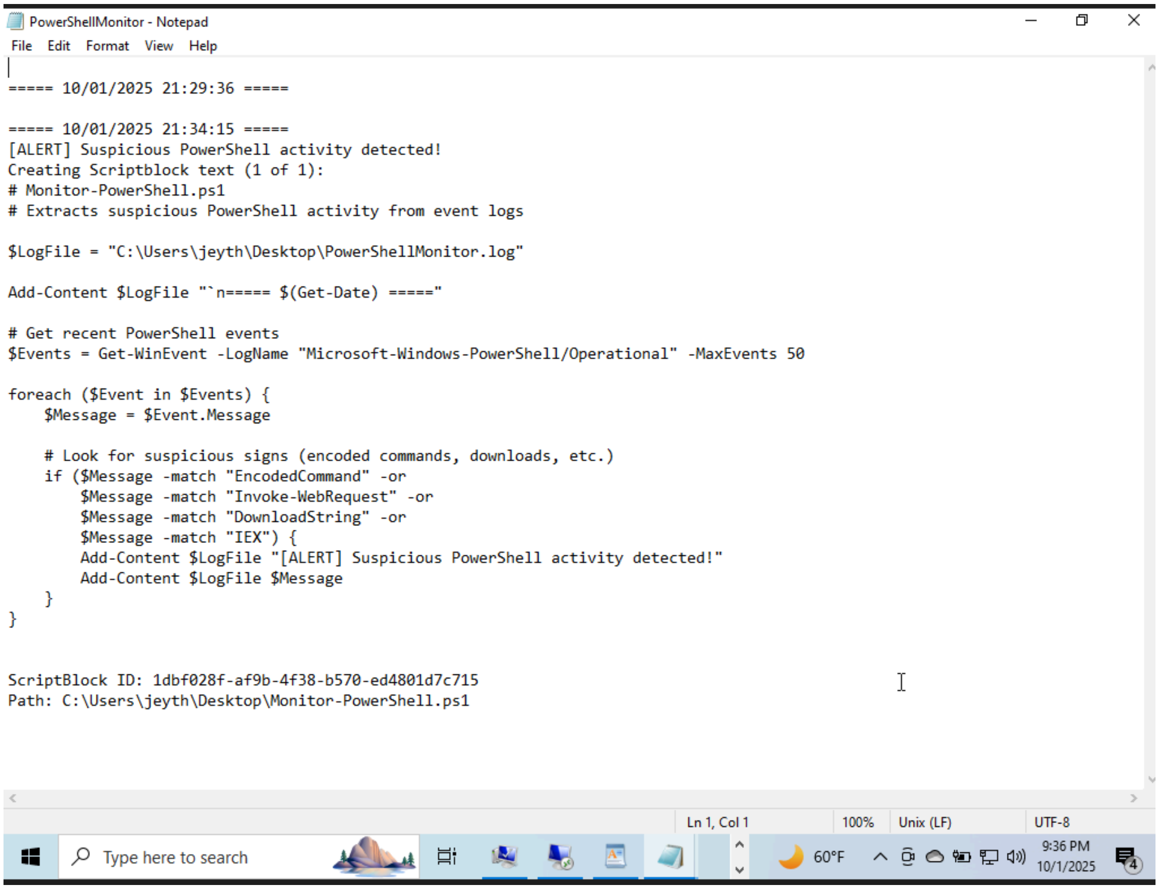
Mimics suspicious activity


Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Completed
Ln 422 Col 56
100%

```
    912      67    26304      16112     4.09   2368   0 SearchIndexer
    371      14     3120       7540     0.47   4472   0 SecurityHealthService
    163       9     1688       9020     0.08   4160   1 SecurityHealthSystray
    358      11     3452       7088     2.70    576   0 services
    105       7     3668       6712     0.17   3848   0 SgrmBroker
    612      26    11920      33900     0.56   2748   1 ShellExperienceHost
    595      19     6160      20028     2.22   1552   1 sihost
    970     534   620768       2300    19.86   5788   1 SkypeApp
    150       8     2004       5504     0.22   5900   1 SkypeBackgroundHost
     53       3     1064        848     0.05    308   0 smss
    422      20     5172      12808     0.50   1784   0 spoolsv
    610      28    18732      24164     0.95   6212   1 StartMenuExperienceHost
    847      23    17240      25520     3.84    292   0 svchost
    686      27    48396      47480    17.73    372   0 svchost
   1313      26    10712      16924     6.92    704   0 svchost
   1019      20     7152      14008    11.33    800   0 svchost
    492      21    14572      13444     1.56    988   0 svchost
   2541     116   104832      54076   162.20   1008   0 svchost
    985      46    11372      20792     3.30   1028   0 svchost
    713      37     8020      16740     3.02   1148   0 svchost
    350      13     3148      10828     0.44   1512   0 svchost
    352      21     3548       8660     0.61   1524   0 svchost
    725      29    12184      24420     4.05   1568   1 svchost
    753      29    20656      20052     3.20   1644   0 svchost
    133      10     1500       4688     0.05   1652   0 svchost
    361      14     2304       6776     0.31   1660   0 svchost
    216      11     7804      16324     3.25   1676   0 svchost
    231      12     2328      10196     0.09   1752   0 svchost
    485      33    10616      22688     2.95   1888   0 svchost
    396      25    15884      10776    10.13   1992   0 svchost
    382      24     3564      10776     0.16   2272   0 svchost
    201      11     1936       7616     0.03   2636   0 svchost
   2947      16    12928       7004     1.61   2668   0 svchost
    349      19     9336      11188    13.59   2944   0 svchost
    160      10     1828       7012     0.14   3024   0 svchost
    342      17     4252      23544     0.73   3492   1 svchost
    307      13     3384      11076     0.20   6220   0 svchost
    133       8     1588       9704     0.05   6720   1 svchost
```


Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Completed
Ln 422 Col 56
100%

```
    201      11     1936       7616     0.03   2636   0 svchost
   2947      16    12928       7004     1.61   2668   0 svchost
    349      19     9336      11188    13.59   2944   0 svchost
    160      10     1828       7012     0.14   3024   0 svchost
    342      17     4252      23544     0.73   3492   1 svchost
    307      13     3384      11076     0.20   6220   0 svchost
    133       8     1588       9704     0.05   6720   1 svchost
   2486       0      200        132   132.11      4   0 System
    253      17     2784      14620     0.11   6704   1 SystemPropertiesComputerName
    306      33     6956      17624     1.31   2168   1 taskhostw
    376      21     6560      11688     0.66   6464   1 taskhostw
    129       9     1388       7176     0.03   6932   1 taskhostw
    544      22     8916      23308     0.59   1100   1 TextInputHost
    164      11     1460       6060     0.06    484   0 wininit
    270      12     2588       9176     0.06    552   1 winlogon
   1096      45    19452       1024     1.02   5712   1 WinStore.App
    397      67    13748      15808     0.81   4336   1 wordpad

PS C:\Users\jeyth\Desktop> .\Monitor-PowerShell.ps1

PS C:\Users\jeyth\Desktop> notepad C:\Users\jeyth\Desktop\PowerShellMonitor.log

PS C:\Users\jeyth\Desktop>
```



```
===== 10/01/2025 21:29:36 =====

===== 10/01/2025 21:34:15 =====
[ALERT] Suspicious PowerShell activity detected!
Creating Scriptblock text (1 of 1):
# Monitor-PowerShell.ps1
# Extracts suspicious PowerShell activity from event logs

$LogFile = "C:\Users\jeyth\Desktop\PowerShellMonitor.log"

Add-Content $LogFile "`n===== $(Get-Date) ====="

# Get recent PowerShell events
$Events = Get-WinEvent -LogName "Microsoft-Windows-PowerShell/Operational" -MaxEvents 50

foreach ($Event in $Events) {
    $Message = $Event.Message

    # Look for suspicious signs (encoded commands, downloads, etc.)
    if ($Message -match "EncodedCommand" -or
        $Message -match "Invoke-WebRequest" -or
        $Message -match "DownloadString" -or
        $Message -match "IEX") {
        Add-Content $LogFile "[ALERT] Suspicious PowerShell activity detected!"
        Add-Content $LogFile $Message
    }
}


ScriptBlock ID: 1dbf028f-af9b-4f38-b570-ed4801d7c715
Path: C:\Users\jeyth\Desktop\Monitor-PowerShell.ps1
```

Task – Configuring and Testing Windows Firewall

### Configure-Firewall.ps1

================================

# Wonderville Firewall Hardening

# Configure-Firewall.ps1

================================

Write-Output "=== Configuring Windows Firewall for Wonderville IT ==="

# 1. Enable firewall for all profiles

Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True

# 2. Clear old custom rules (optional cleanup step)

Get-NetFirewallRule -DisplayName "Allow RDP from Internal" -ErrorAction SilentlyContinue | Remove-NetFirewallRule

```
Get-NetFirewallRule -DisplayName "Allow SMB" -ErrorAction SilentlyContinue | Remove-NetFirewallRule

Get-NetFirewallRule -DisplayName "Allow DNS" -ErrorAction SilentlyContinue | Remove-NetFirewallRule

Get-NetFirewallRule -DisplayName "Allow Web Traffic" -ErrorAction SilentlyContinue | Remove-NetFirewallRule

Get-NetFirewallRule -DisplayName "Block RDP from External" -ErrorAction SilentlyContinue | Remove-NetFirewallRule

# 3. Allow RDP only from internal subnet

New-NetFirewallRule -DisplayName "Allow RDP from Internal" `

    -Direction Inbound -Protocol TCP -LocalPort 3389 `

    -RemoteAddress 192.168.90.0/24 -Action Allow

# 4. Allow SMB (file sharing) inside the network

New-NetFirewallRule -DisplayName "Allow SMB" `

    -Direction Inbound -Protocol TCP -LocalPort 445 `

    -RemoteAddress 192.168.90.0/24 -Action Allow

# 5. Allow DNS lookups (UDP port 53)

New-NetFirewallRule -DisplayName "Allow DNS" `

    -Direction Outbound -Protocol UDP -LocalPort 53 -Action Allow

# 6. Allow HTTP (80) and HTTPS (443) traffic

New-NetFirewallRule -DisplayName "Allow Web Traffic" `

    -Direction Outbound -Protocol TCP -LocalPort 80,443 -Action Allow

# 7. Block RDP from anywhere outside internal subnet

New-NetFirewallRule -DisplayName "Block RDP from External" `

    -Direction Inbound -Protocol TCP -LocalPort 3389 `

    -RemoteAddress Any -Action Block

# 8. Enable firewall logging

Set-NetFirewallProfile -Profile Domain,Public,Private `

    -LogAllowed True -LogBlocked True `
    -LogFileName "C:\Windows\System32\LogFiles\Firewall\pfirewall.log" `
    -LogMaxSizeKilobytes 32767
Write-Output "=== Firewall rules configured successfully!
```

Goal: Protect the Wonderville internal Windows servers by enforcing host-based firewall rules in addition to the router firewall.

Actions Taken:

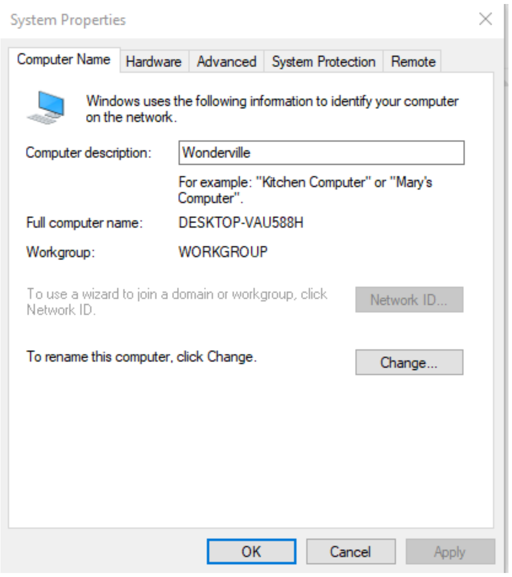
- Wrote a PowerShell script to enable Windows Firewall across all profiles (Domain, Private, Public).
- Configured allow rules for essential services (RDP from internal subnet 192.168.90.0/24, SMB file sharing, DNS, HTTP/HTTPS).
- Configured deny rules for RDP from outside the subnet (blocks external brute-force attacks).
- Enabled firewall logging (pfirewall.log) to capture allowed/blocked traffic.

Testing:

- From inside → RDP works .
- From outside subnet → RDP fails (blocked) .
- Firewall log confirmed entries:
  - ALLOW TCP … 3389 for internal RDP.
  - DROP TCP … 3389 for external attempts.
  - ALLOW TCP … 443 for web browsing.
  - ALLOW UDP … 53 for DNS.

Outcome: Internal users can still connect securely, but external attackers are blocked.

Linda now has a baseline access control policy at the host level.



Windows uses the following information to identify your computer on the network.

Computer description: Wonderville

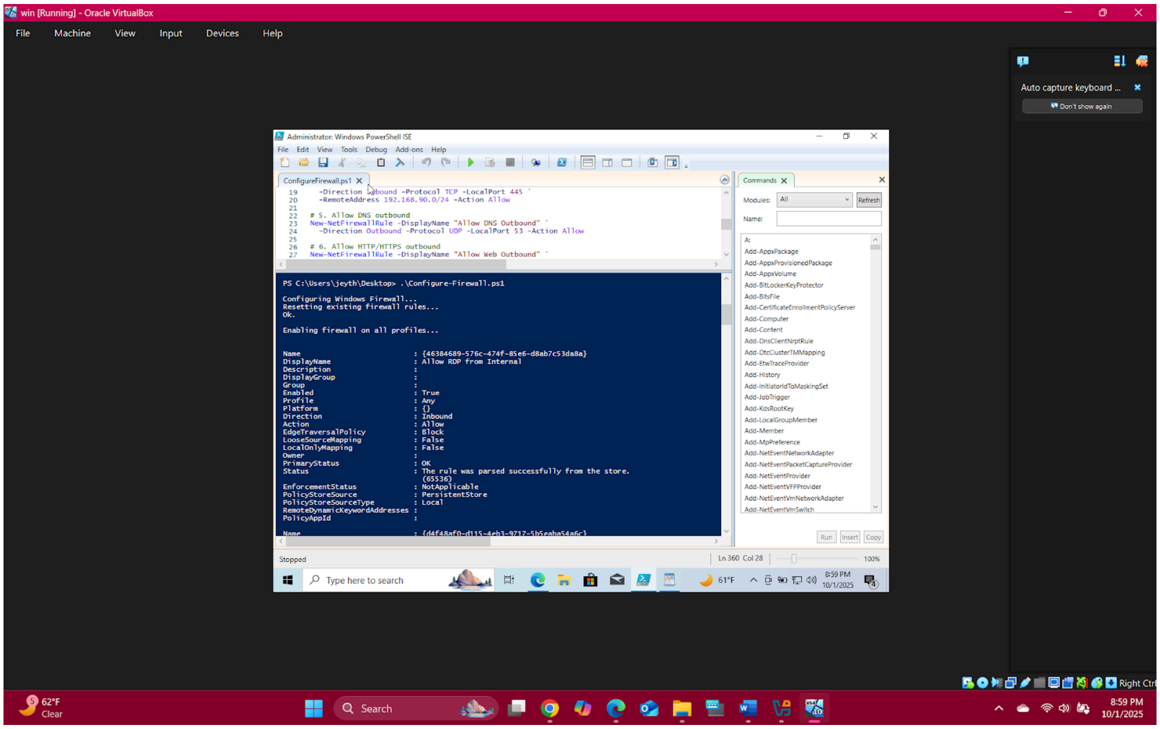
For example: "Kitchen Computer" or "Mary's Computer".

Full computer name: DESKTOP-VAU588H

Workgroup: WORKGROUP

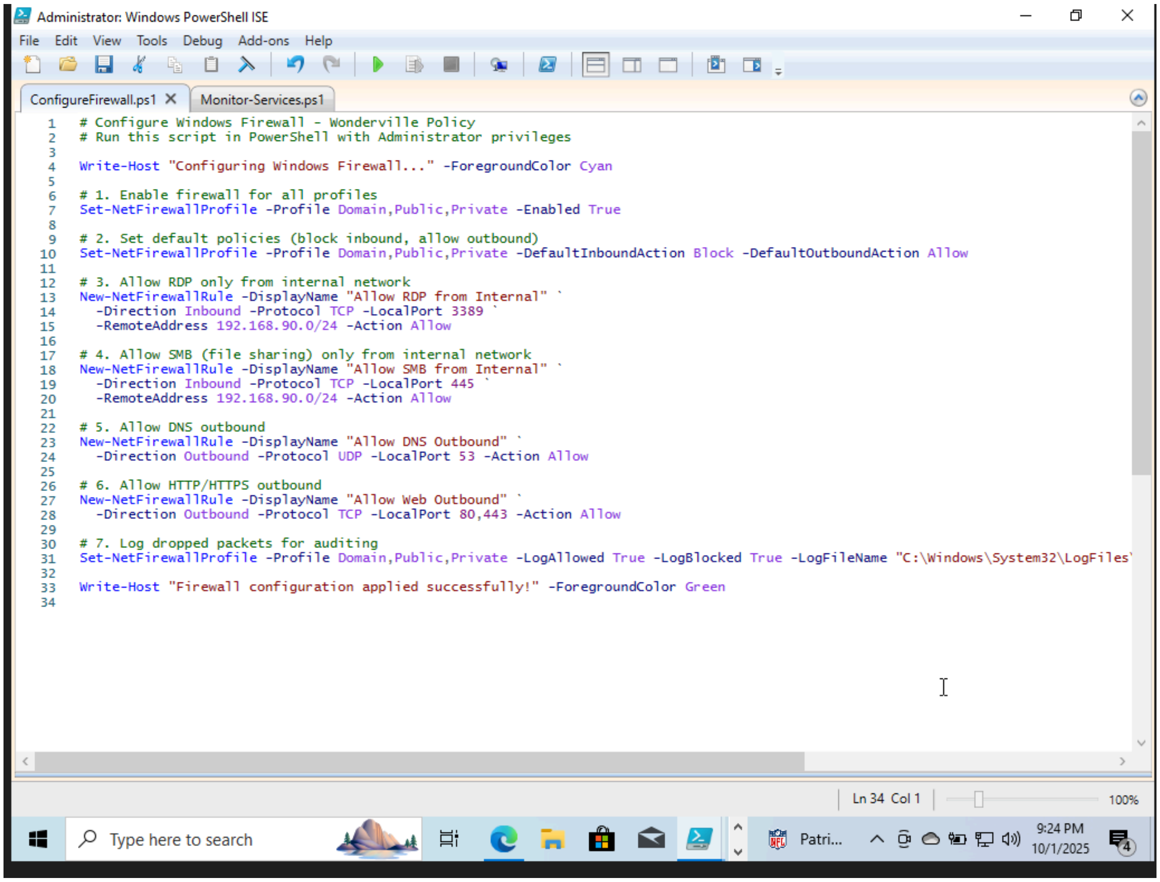
To use a wizard to join a domain or workgroup, click Network ID.

To rename this computer, click Change.

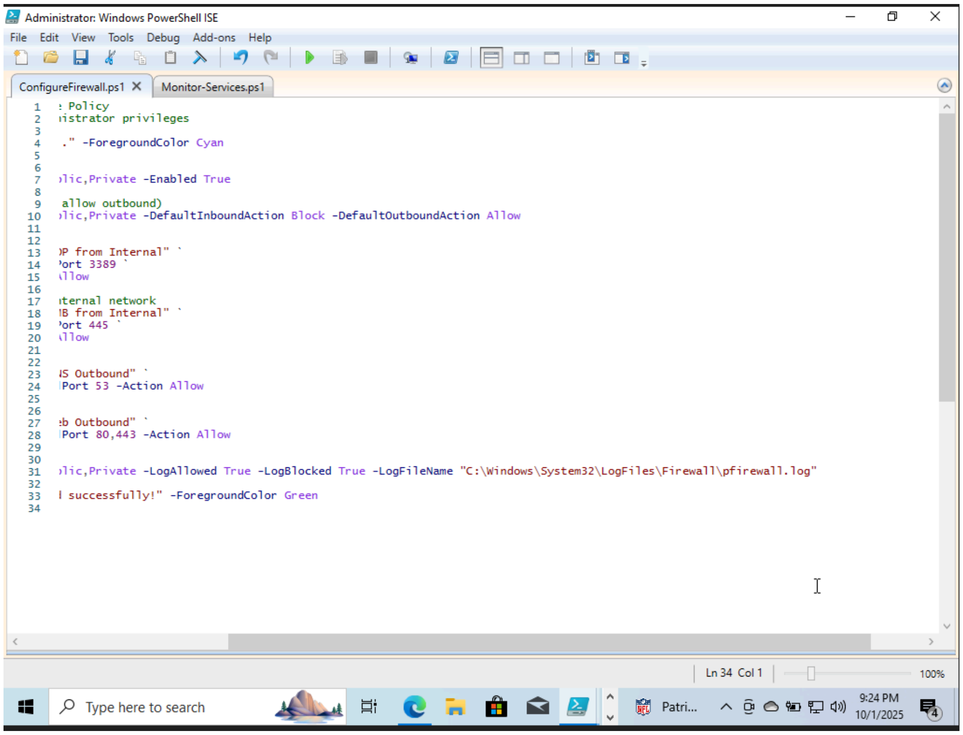


```
  -Direction Inbound -Protocol TCP -LocalPort 445 `
  -RemoteAddress 192.168.90.0/24 -Action Allow

# 5. Allow DNS outbound
New-NetFirewallRule -DisplayName "Allow DNS Outbound" `
  -Direction Outbound -Protocol UDP -LocalPort 53 -Action Allow

# 6. Allow HTTP/HTTPS outbound
New-NetFirewallRule -DisplayName "Allow Web Outbound" `
```

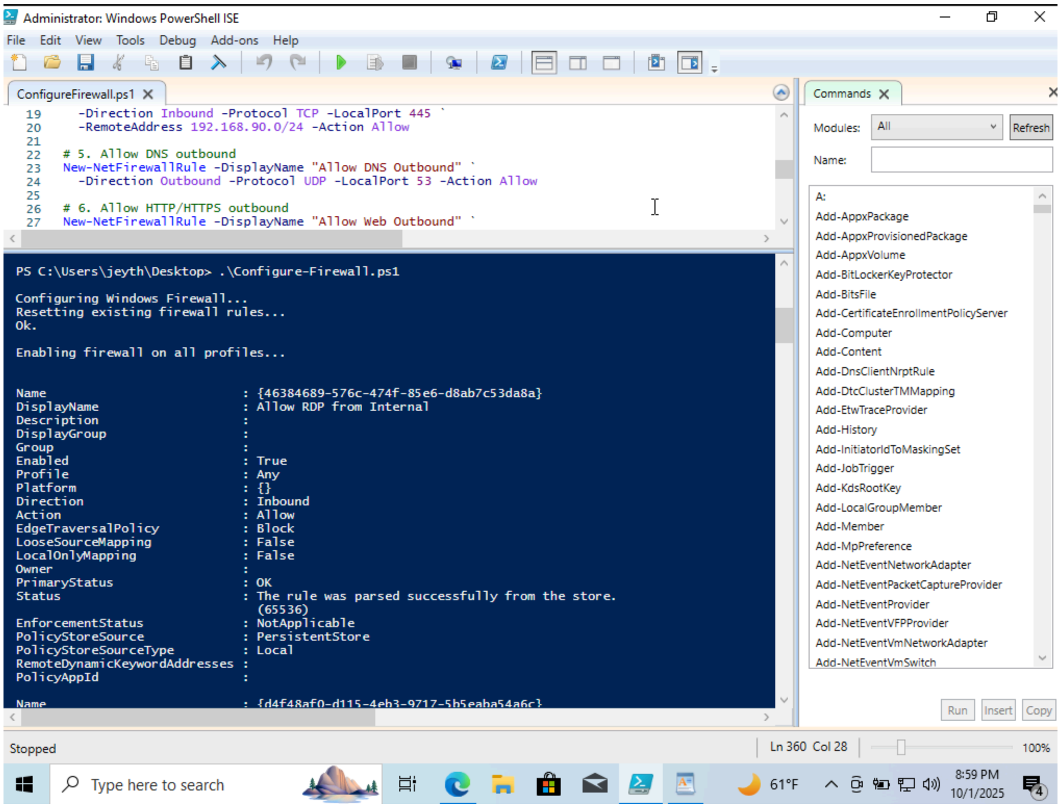
```
PS C:\Users\jeyth\Desktop> .\Configure-Firewall.ps1

Configuring Windows Firewall...
Resetting existing firewall rules...
Ok.

Enabling firewall on all profiles...


Name                          : {46384689-576c-474f-85e6-d8ab7c53da8a}
DisplayName                   : Allow RDP from Internal
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store.
                                (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses :
PolicyAppId                   :

Name                          : {d4f48af0-d115-4eb3-9717-5b5eaba54a6c}
```



```powershell
# Configure Windows Firewall - Wonderville Policy
# Run this script in PowerShell with Administrator privileges

Write-Host "Configuring Windows Firewall..." -ForegroundColor Cyan

# 1. Enable firewall for all profiles
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True

# 2. Set default policies (block inbound, allow outbound)
Set-NetFirewallProfile -Profile Domain,Public,Private -DefaultInboundAction Block -DefaultOutboundAction Allow

# 3. Allow RDP only from internal network
New-NetFirewallRule -DisplayName "Allow RDP from Internal" `
  -Direction Inbound -Protocol TCP -LocalPort 3389 `
  -RemoteAddress 192.168.90.0/24 -Action Allow

# 4. Allow SMB (file sharing) only from internal network
New-NetFirewallRule -DisplayName "Allow SMB from Internal" `
  -Direction Inbound -Protocol TCP -LocalPort 445 `
  -RemoteAddress 192.168.90.0/24 -Action Allow

# 5. Allow DNS outbound
New-NetFirewallRule -DisplayName "Allow DNS Outbound" `
  -Direction Outbound -Protocol UDP -LocalPort 53 -Action Allow

# 6. Allow HTTP/HTTPS outbound
New-NetFirewallRule -DisplayName "Allow Web Outbound" `
  -Direction Outbound -Protocol TCP -LocalPort 80,443 -Action Allow

# 7. Log dropped packets for auditing
Set-NetFirewallProfile -Profile Domain,Public,Private -LogAllowed True -LogBlocked True -LogFileName "C:\Windows\System32\LogFiles'

Write-Host "Firewall configuration applied successfully!" -ForegroundColor Green

```

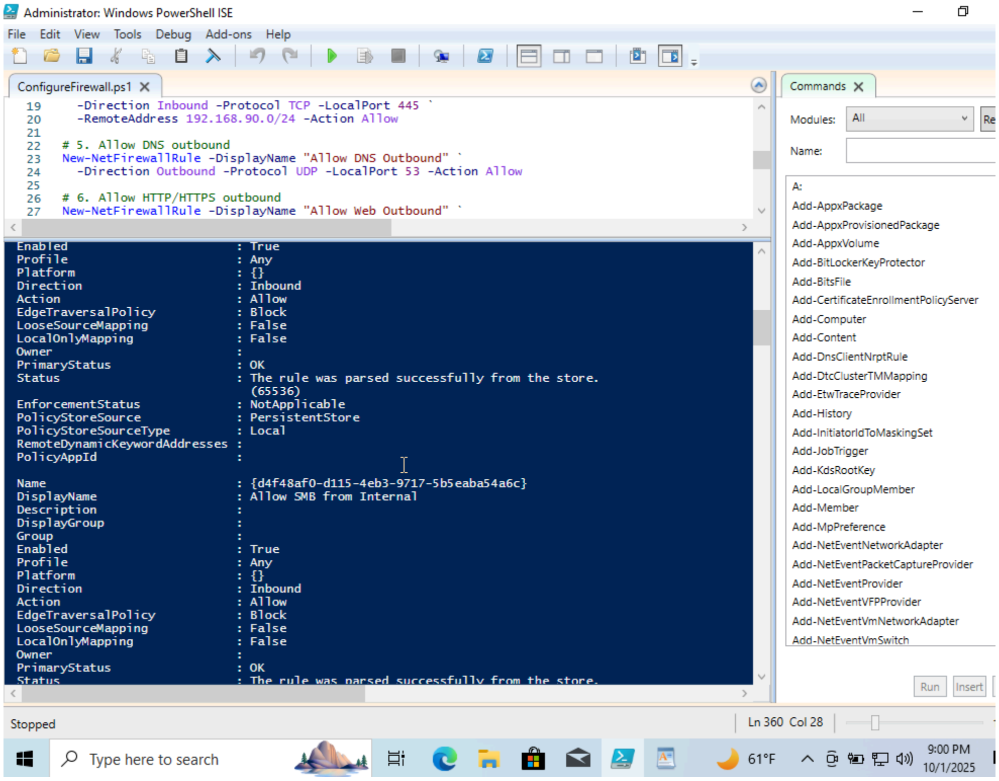


```
: Policy
istrator privileges

." -ForegroundColor Cyan


lic,Private -Enabled True

 allow outbound)
lic,Private -DefaultInboundAction Block -DefaultOutboundAction Allow


P from Internal" `
ort 3389 `
llow

ternal network
B from Internal" `
ort 445 `
llow


S Outbound" `
Port 53 -Action Allow


b Outbound" `
Port 80,443 -Action Allow


lic,Private -LogAllowed True -LogBlocked True -LogFileName "C:\Windows\System32\LogFiles\Firewall\pfirewall.log"

l successfully!" -ForegroundColor Green

```



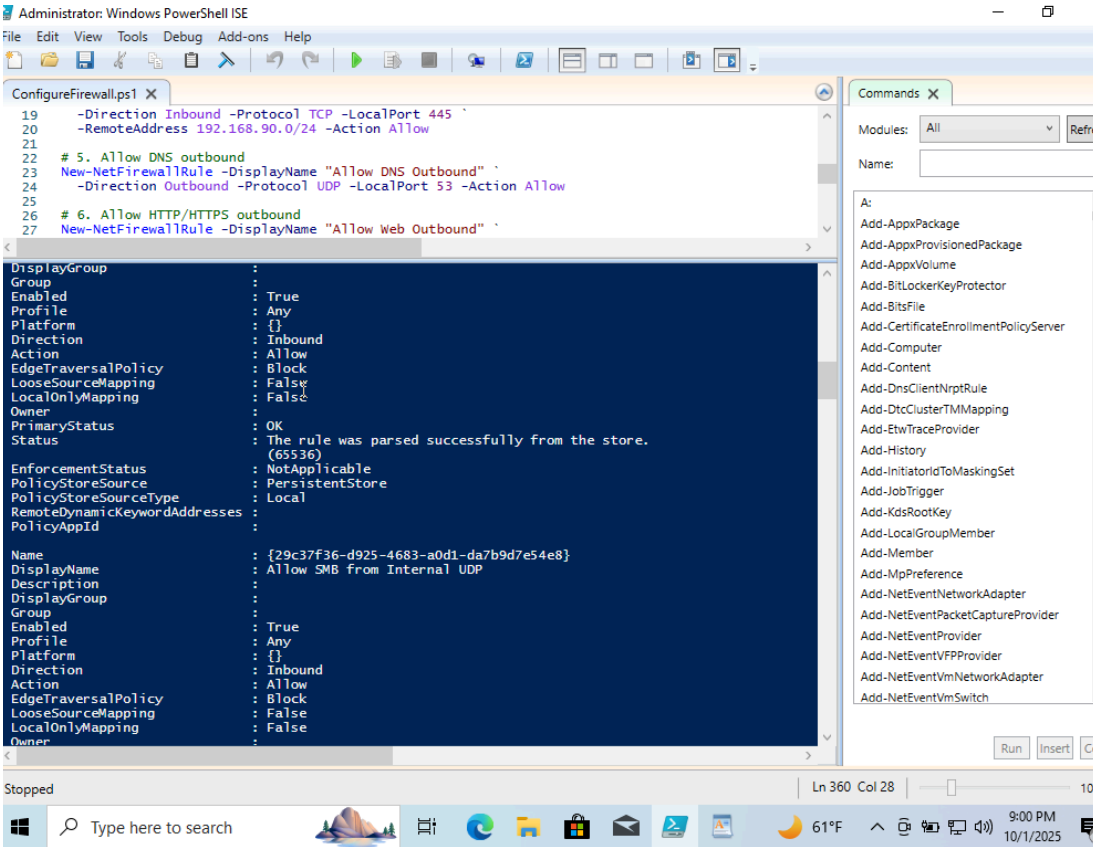
```
    -Direction Inbound -Protocol TCP -LocalPort 445 `
    -RemoteAddress 192.168.90.0/24 -Action Allow

  # 5. Allow DNS outbound
  New-NetFirewallRule -DisplayName "Allow DNS Outbound" `
    -Direction Outbound -Protocol UDP -LocalPort 53 -Action Allow

  # 6. Allow HTTP/HTTPS outbound
  New-NetFirewallRule -DisplayName "Allow Web Outbound" `
```

```
PS C:\Users\jeyth\Desktop> .\Configure-Firewall.ps1

Configuring Windows Firewall...
Resetting existing firewall rules...
Ok.

Enabling firewall on all profiles...


Name                          : {46384689-576c-474f-85e6-d8ab7c53da8a}
DisplayName                   : Allow RDP from Internal
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store.
                                (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses :
PolicyAppId                   :

Name                          : {d4f48af0-d115-4eb3-9717-5b5eaba54a6c}
```



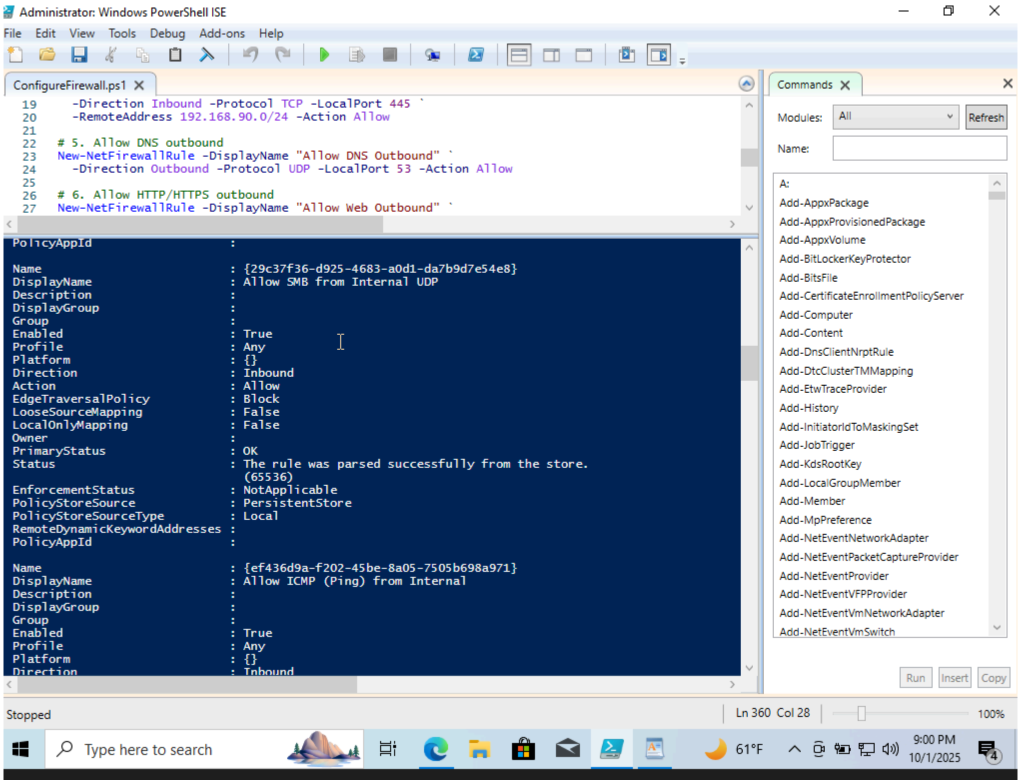
```
    -Direction Inbound -Protocol TCP -LocalPort 445 `
    -RemoteAddress 192.168.90.0/24 -Action Allow

  # 5. Allow DNS outbound
  New-NetFirewallRule -DisplayName "Allow DNS Outbound" `
    -Direction Outbound -Protocol UDP -LocalPort 53 -Action Allow

  # 6. Allow HTTP/HTTPS outbound
  New-NetFirewallRule -DisplayName "Allow Web Outbound" `
```

```
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store.
                                (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses :
PolicyAppId                   :

Name                          : {d4f48af0-d115-4eb3-9717-5b5eaba54a6c}
DisplayName                   : Allow SMB from Internal
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store.
```



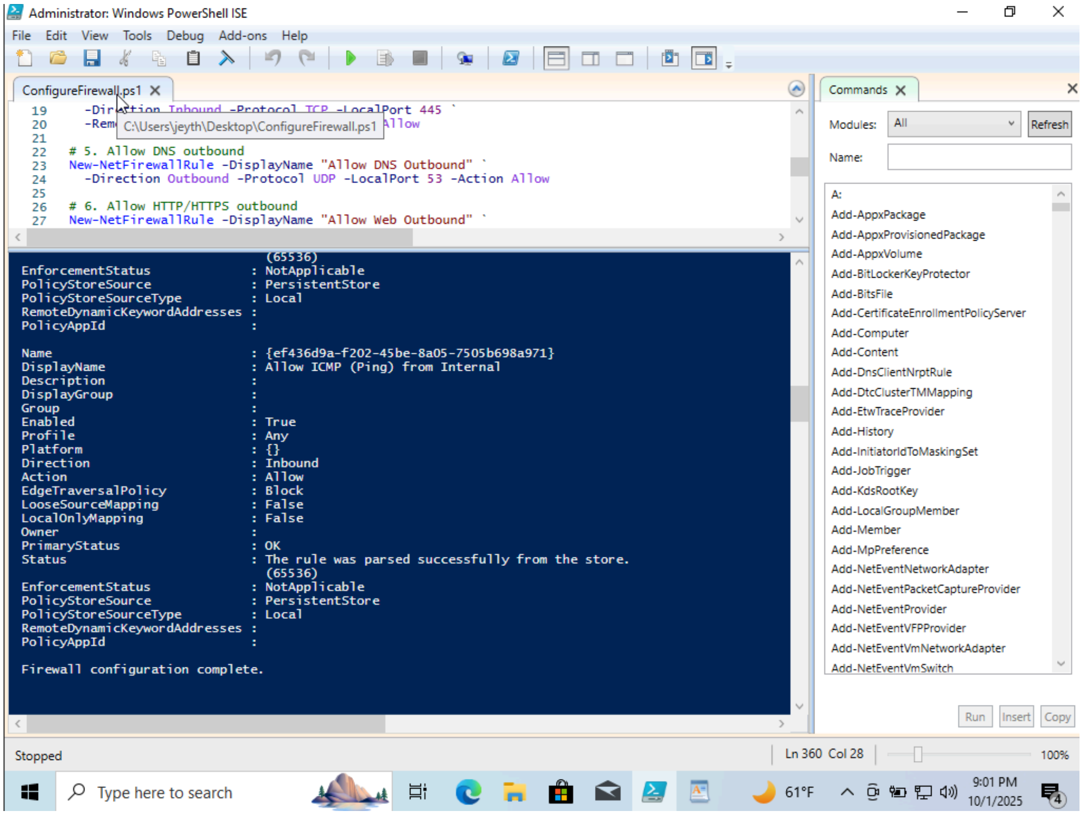
```
  -Direction Inbound -Protocol TCP -LocalPort 445 `
  -RemoteAddress 192.168.90.0/24 -Action Allow

# 5. Allow DNS outbound
New-NetFirewallRule -DisplayName "Allow DNS Outbound" `
  -Direction Outbound -Protocol UDP -LocalPort 53 -Action Allow

# 6. Allow HTTP/HTTPS outbound
New-NetFirewallRule -DisplayName "Allow Web Outbound" `
```

```
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store.
                                (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses :
PolicyAppId                   :

Name                          : {29c37f36-d925-4683-a0d1-da7b9d7e54e8}
DisplayName                   : Allow SMB from Internal UDP
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
```



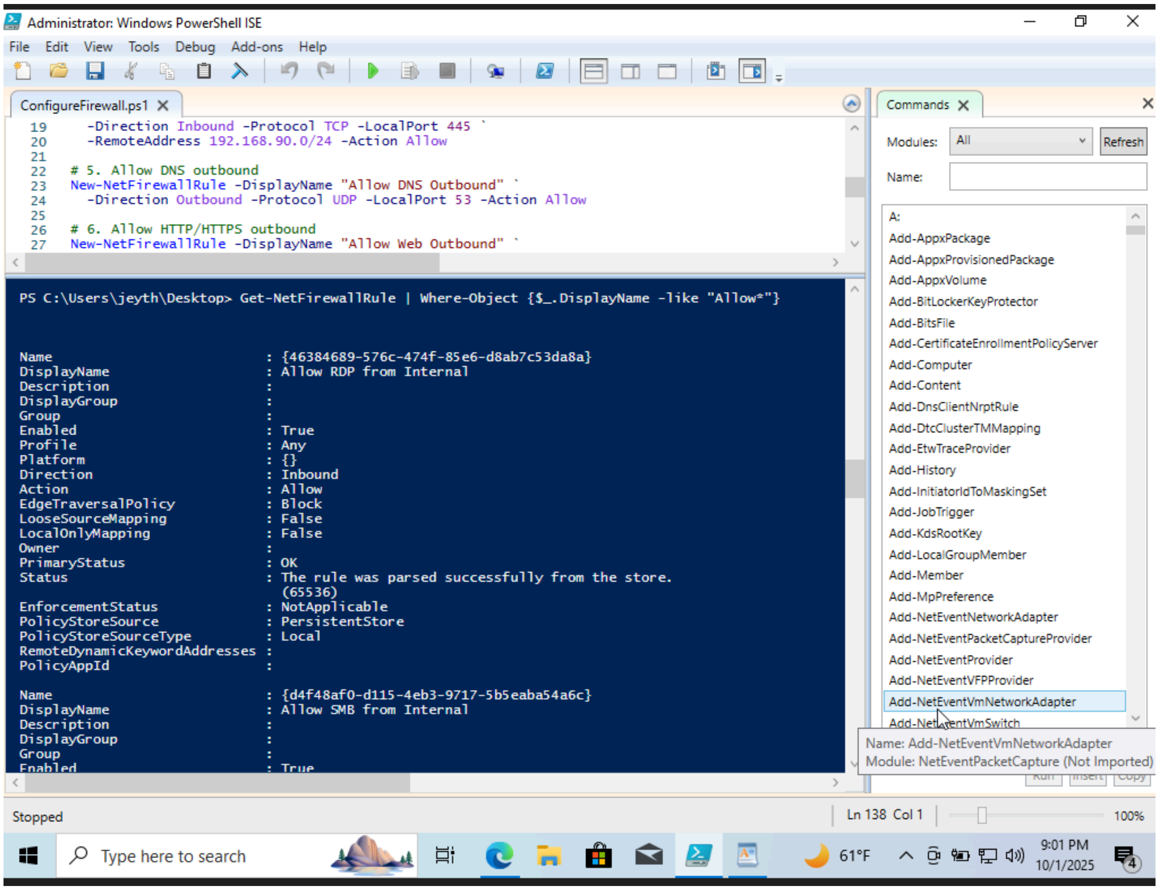
```
  -Direction Inbound -Protocol TCP -LocalPort 445 `
  -RemoteAddress 192.168.90.0/24 -Action Allow

# 5. Allow DNS outbound
New-NetFirewallRule -DisplayName "Allow DNS Outbound" `
  -Direction Outbound -Protocol UDP -LocalPort 53 -Action Allow

# 6. Allow HTTP/HTTPS outbound
New-NetFirewallRule -DisplayName "Allow Web Outbound" `
```

```
PolicyAppId                   :

Name                          : {29c37f36-d925-4683-a0d1-da7b9d7e54e8}
DisplayName                   : Allow SMB from Internal UDP
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store.
                                (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses :
PolicyAppId                   :

Name                          : {ef436d9a-f202-45be-8a05-7505b698a971}
DisplayName                   : Allow ICMP (Ping) from Internal
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
```



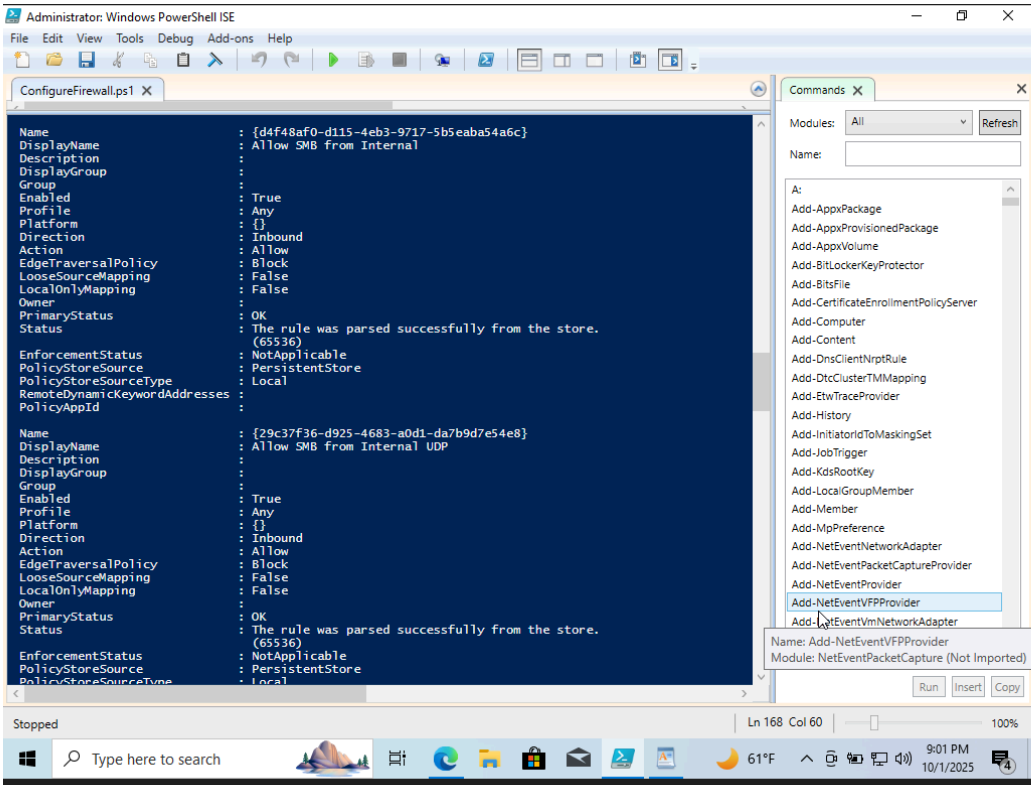
```
    -Direction Inbound -Protocol TCP -LocalPort 445 `
    -Rem                                            Allow

  # 5. Allow DNS outbound
  New-NetFirewallRule -DisplayName "Allow DNS Outbound" `
    -Direction Outbound -Protocol UDP -LocalPort 53 -Action Allow

  # 6. Allow HTTP/HTTPS outbound
  New-NetFirewallRule -DisplayName "Allow Web Outbound" `
```

```
                                (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses :
PolicyAppId                   :

Name                          : {ef436d9a-f202-45be-8a05-7505b698a971}
DisplayName                   : Allow ICMP (Ping) from Internal
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store.
                                (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses :
PolicyAppId                   :

Firewall configuration complete.
```

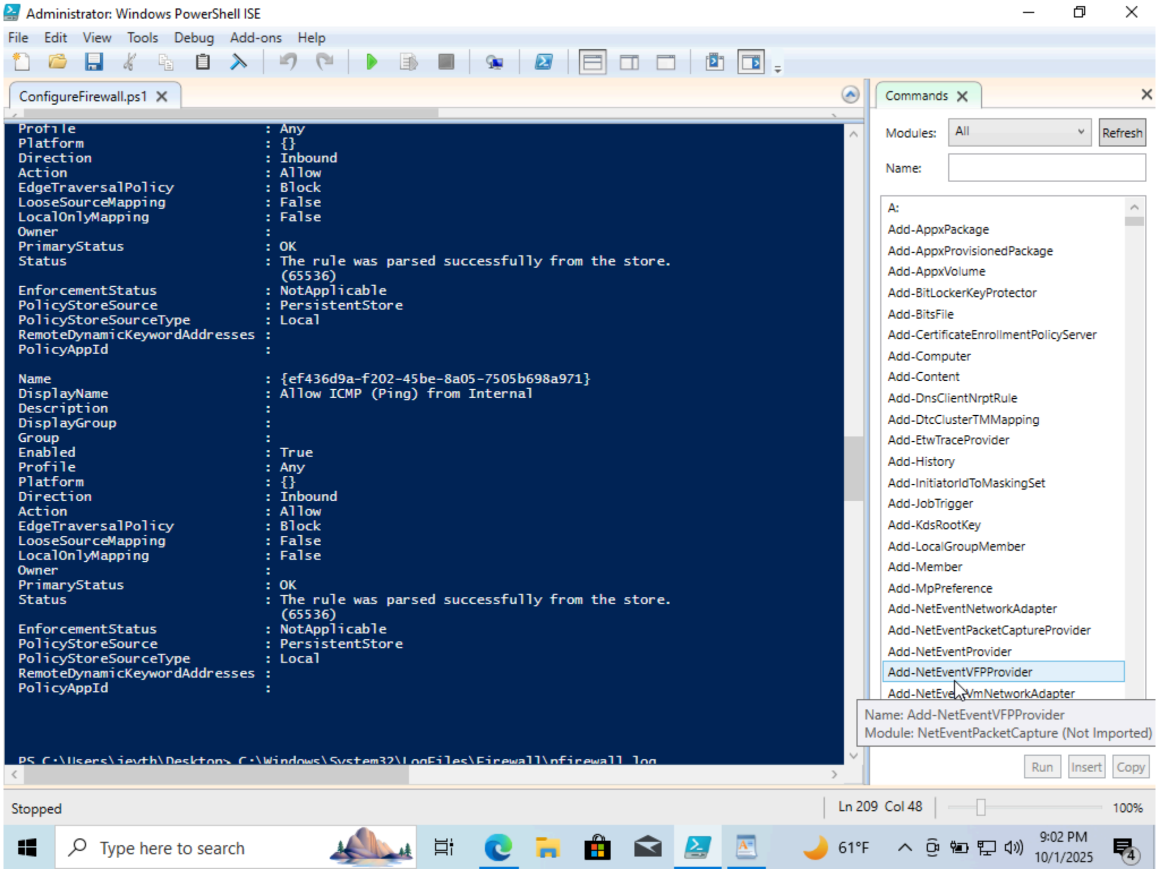


```
    -Direction Inbound -Protocol TCP -LocalPort 445 `
    -RemoteAddress 192.168.90.0/24 -Action Allow

  # 5. Allow DNS outbound
  New-NetFirewallRule -DisplayName "Allow DNS Outbound" `
    -Direction Outbound -Protocol UDP -LocalPort 53 -Action Allow

  # 6. Allow HTTP/HTTPS outbound
  New-NetFirewallRule -DisplayName "Allow Web Outbound" `
```

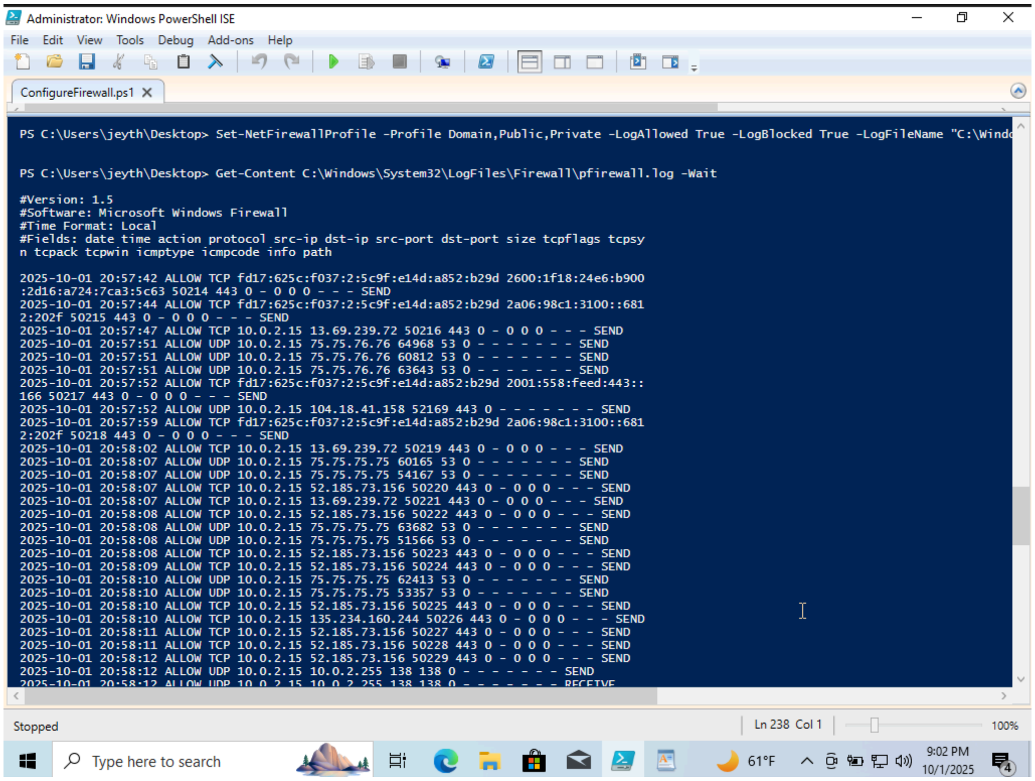
```
PS C:\Users\jeyth\Desktop> Get-NetFirewallRule | Where-Object {$_.DisplayName -like "Allow*"}


Name                          : {46384689-576c-474f-85e6-d8ab7c53da8a}
DisplayName                   : Allow RDP from Internal
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store.
                                (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses :
PolicyAppId                   :

Name                          : {d4f48af0-d115-4eb3-9717-5b5eaba54a6c}
DisplayName                   : Allow SMB from Internal
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
```



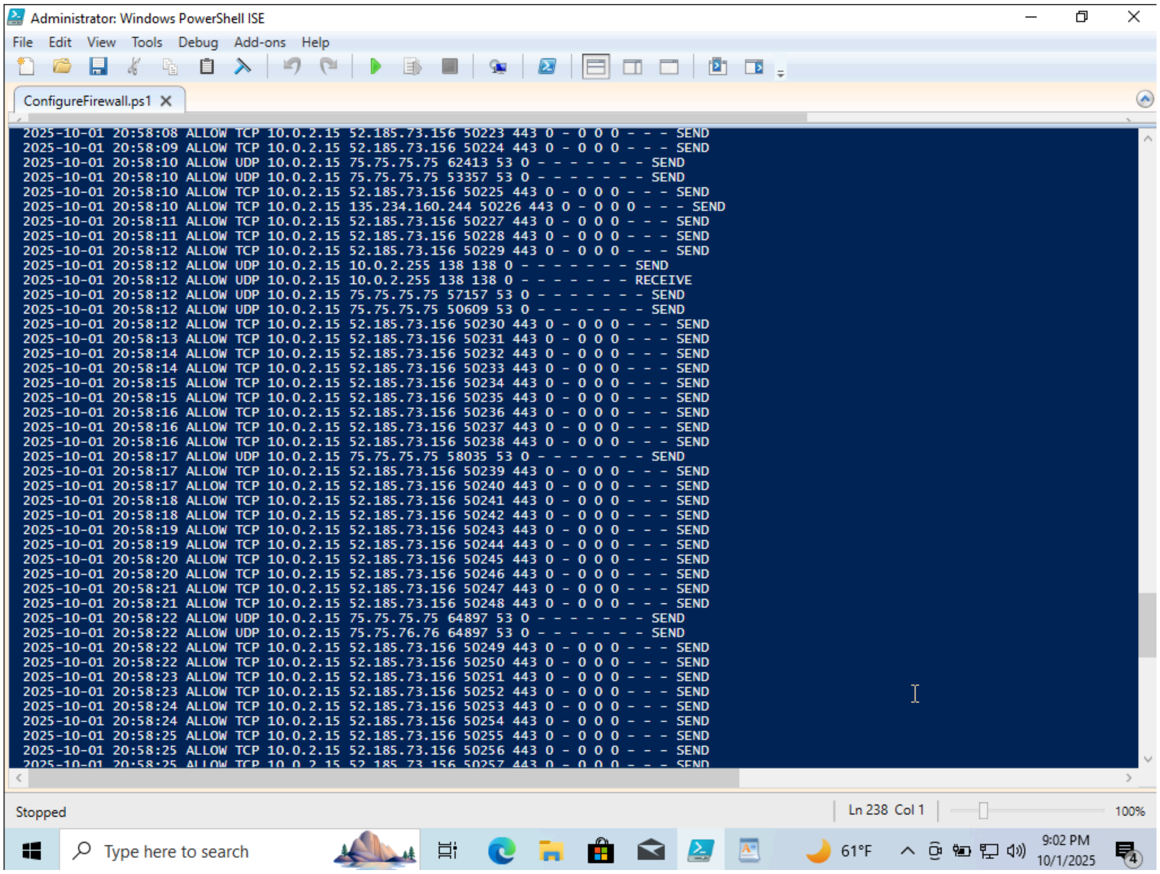
```
Name                          : {d4f48af0-d115-4eb3-9717-5b5eaba54a6c}
DisplayName                   : Allow SMB from Internal
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store.
                                (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses :
PolicyAppId                   :

Name                          : {29c37f36-d925-4683-a0d1-da7b9d7e54e8}
DisplayName                   : Allow SMB from Internal UDP
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store.
                                (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
```



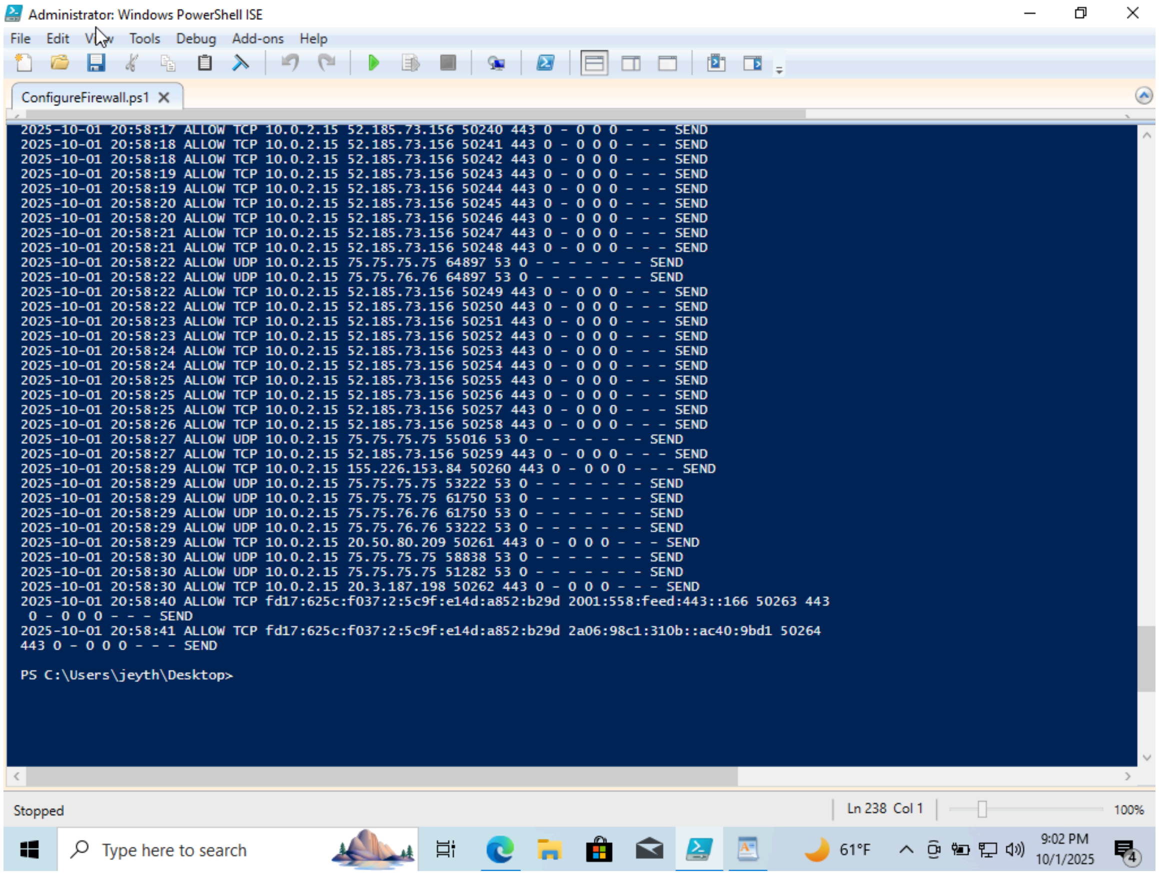
```
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store.
                                (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses :
PolicyAppId                   :

Name                          : {ef436d9a-f202-45be-8a05-7505b698a971}
DisplayName                   : Allow ICMP (Ping) from Internal
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store.
                                (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses :
PolicyAppId                   :


PS C:\Users\jeyth\Desktop> C:\Windows\System32\LogFiles\Firewall\pfirewall.log
```



```
PS C:\Users\jeyth\Desktop> Set-NetFirewallProfile -Profile Domain,Public,Private -LogAllowed True -LogBlocked True -LogFileName "C:\Windo


PS C:\Users\jeyth\Desktop> Get-Content C:\Windows\System32\LogFiles\Firewall\pfirewall.log -Wait

#Version: 1.5
#Software: Microsoft Windows Firewall
#Time Format: Local
#Fields: date time action protocol src-ip dst-ip src-port dst-port size tcpflags tcpsy
n tcpack tcpwin icmptype icmpcode info path

2025-10-01 20:57:42 ALLOW TCP fd17:625c:f037:2:5c9f:e14d:a852:b29d 2600:1f18:24e6:b900
:2d16:a724:7ca3:5c63 50214 443 0 - 0 0 0 - - - SEND
2025-10-01 20:57:44 ALLOW TCP fd17:625c:f037:2:5c9f:e14d:a852:b29d 2a06:98c1:3100::681
2:202f 50215 443 0 - 0 0 0 - - - SEND
2025-10-01 20:57:47 ALLOW TCP 10.0.2.15 13.69.239.72 50216 443 0 - 0 0 0 - - - SEND
2025-10-01 20:57:51 ALLOW UDP 10.0.2.15 75.75.76.76 64968 53 0 - - - - - - - SEND
2025-10-01 20:57:51 ALLOW UDP 10.0.2.15 75.75.76.76 60812 53 0 - - - - - - - SEND
2025-10-01 20:57:51 ALLOW UDP 10.0.2.15 75.75.76.76 63643 53 0 - - - - - - - SEND
2025-10-01 20:57:52 ALLOW TCP fd17:625c:f037:2:5c9f:e14d:a852:b29d 2001:558:feed:443::
166 50217 443 0 - 0 0 0 - - - SEND
2025-10-01 20:57:52 ALLOW UDP 10.0.2.15 104.18.41.158 52169 443 0 - - - - - - - SEND
2025-10-01 20:57:59 ALLOW TCP fd17:625c:f037:2:5c9f:e14d:a852:b29d 2a06:98c1:3100::681
2:202f 50218 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:02 ALLOW TCP 10.0.2.15 13.69.239.72 50219 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:07 ALLOW UDP 10.0.2.15 75.75.75.75 60165 53 0 - - - - - - - SEND
2025-10-01 20:58:07 ALLOW UDP 10.0.2.15 75.75.75.75 54167 53 0 - - - - - - - SEND
2025-10-01 20:58:07 ALLOW TCP 10.0.2.15 52.185.73.156 50220 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:07 ALLOW TCP 10.0.2.15 13.69.239.72 50221 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:08 ALLOW TCP 10.0.2.15 52.185.73.156 50222 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:08 ALLOW UDP 10.0.2.15 75.75.75.75 63682 53 0 - - - - - - - SEND
2025-10-01 20:58:08 ALLOW UDP 10.0.2.15 75.75.75.75 51566 53 0 - - - - - - - SEND
2025-10-01 20:58:08 ALLOW TCP 10.0.2.15 52.185.73.156 50223 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:09 ALLOW TCP 10.0.2.15 52.185.73.156 50224 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:10 ALLOW UDP 10.0.2.15 75.75.75.75 62413 53 0 - - - - - - - SEND
2025-10-01 20:58:10 ALLOW UDP 10.0.2.15 75.75.75.75 53357 53 0 - - - - - - - SEND
2025-10-01 20:58:10 ALLOW TCP 10.0.2.15 52.185.73.156 50225 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:10 ALLOW TCP 10.0.2.15 135.234.160.244 50226 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:11 ALLOW TCP 10.0.2.15 52.185.73.156 50227 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:11 ALLOW TCP 10.0.2.15 52.185.73.156 50228 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:12 ALLOW TCP 10.0.2.15 52.185.73.156 50229 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:12 ALLOW UDP 10.0.2.15 10.0.2.255 138 138 0 - - - - - - - SEND
2025-10-01 20:58:12 ALLOW UDP 10.0.2.15 10.0.2.255 138 138 0 - - - - - - - RECEIVE
```



```
2025-10-01 20:58:08 ALLOW TCP 10.0.2.15 52.185.73.156 50223 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:09 ALLOW TCP 10.0.2.15 52.185.73.156 50224 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:10 ALLOW UDP 10.0.2.15 75.75.75.75 62413 53 0 - - - - - - - SEND
2025-10-01 20:58:10 ALLOW UDP 10.0.2.15 75.75.75.75 53357 53 0 - - - - - - - SEND
2025-10-01 20:58:10 ALLOW TCP 10.0.2.15 52.185.73.156 50225 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:10 ALLOW TCP 10.0.2.15 135.234.160.244 50226 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:11 ALLOW TCP 10.0.2.15 52.185.73.156 50227 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:11 ALLOW TCP 10.0.2.15 52.185.73.156 50228 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:12 ALLOW TCP 10.0.2.15 52.185.73.156 50229 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:12 ALLOW UDP 10.0.2.15 10.0.2.255 138 138 0 - - - - - - - SEND
2025-10-01 20:58:12 ALLOW UDP 10.0.2.15 10.0.2.255 138 138 0 - - - - - - - RECEIVE
2025-10-01 20:58:12 ALLOW UDP 10.0.2.15 75.75.75.75 57157 53 0 - - - - - - - SEND
2025-10-01 20:58:12 ALLOW UDP 10.0.2.15 75.75.75.75 50609 53 0 - - - - - - - SEND
2025-10-01 20:58:12 ALLOW TCP 10.0.2.15 52.185.73.156 50230 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:13 ALLOW TCP 10.0.2.15 52.185.73.156 50231 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:14 ALLOW TCP 10.0.2.15 52.185.73.156 50232 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:14 ALLOW TCP 10.0.2.15 52.185.73.156 50233 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:15 ALLOW TCP 10.0.2.15 52.185.73.156 50234 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:15 ALLOW TCP 10.0.2.15 52.185.73.156 50235 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:16 ALLOW TCP 10.0.2.15 52.185.73.156 50236 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:16 ALLOW TCP 10.0.2.15 52.185.73.156 50237 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:16 ALLOW TCP 10.0.2.15 52.185.73.156 50238 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:17 ALLOW UDP 10.0.2.15 75.75.75.75 58035 53 0 - - - - - - - SEND
2025-10-01 20:58:17 ALLOW TCP 10.0.2.15 52.185.73.156 50239 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:17 ALLOW TCP 10.0.2.15 52.185.73.156 50240 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:18 ALLOW TCP 10.0.2.15 52.185.73.156 50241 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:18 ALLOW TCP 10.0.2.15 52.185.73.156 50242 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:19 ALLOW TCP 10.0.2.15 52.185.73.156 50243 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:19 ALLOW TCP 10.0.2.15 52.185.73.156 50244 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:20 ALLOW TCP 10.0.2.15 52.185.73.156 50245 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:20 ALLOW TCP 10.0.2.15 52.185.73.156 50246 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:21 ALLOW TCP 10.0.2.15 52.185.73.156 50247 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:21 ALLOW TCP 10.0.2.15 52.185.73.156 50248 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:22 ALLOW UDP 10.0.2.15 75.75.75.75 64897 53 0 - - - - - - - SEND
2025-10-01 20:58:22 ALLOW UDP 10.0.2.15 75.75.76.76 64897 53 0 - - - - - - - SEND
2025-10-01 20:58:22 ALLOW TCP 10.0.2.15 52.185.73.156 50249 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:22 ALLOW TCP 10.0.2.15 52.185.73.156 50250 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:23 ALLOW TCP 10.0.2.15 52.185.73.156 50251 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:23 ALLOW TCP 10.0.2.15 52.185.73.156 50252 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:24 ALLOW TCP 10.0.2.15 52.185.73.156 50253 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:24 ALLOW TCP 10.0.2.15 52.185.73.156 50254 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:25 ALLOW TCP 10.0.2.15 52.185.73.156 50255 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:25 ALLOW TCP 10.0.2.15 52.185.73.156 50256 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:25 ALLOW TCP 10.0.2.15 52.185.73.156 50257 443 0 - 0 0 0 - - - SEND
```



```
2025-10-01 20:58:17 ALLOW TCP 10.0.2.15 52.185.73.156 50240 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:18 ALLOW TCP 10.0.2.15 52.185.73.156 50241 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:18 ALLOW TCP 10.0.2.15 52.185.73.156 50242 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:19 ALLOW TCP 10.0.2.15 52.185.73.156 50243 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:19 ALLOW TCP 10.0.2.15 52.185.73.156 50244 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:20 ALLOW TCP 10.0.2.15 52.185.73.156 50245 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:20 ALLOW TCP 10.0.2.15 52.185.73.156 50246 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:21 ALLOW TCP 10.0.2.15 52.185.73.156 50247 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:21 ALLOW TCP 10.0.2.15 52.185.73.156 50248 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:22 ALLOW UDP 10.0.2.15 75.75.75.75 64897 53 0 - - - - - - - SEND
2025-10-01 20:58:22 ALLOW UDP 10.0.2.15 75.75.76.76 64897 53 0 - - - - - - - SEND
2025-10-01 20:58:22 ALLOW TCP 10.0.2.15 52.185.73.156 50249 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:22 ALLOW TCP 10.0.2.15 52.185.73.156 50250 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:23 ALLOW TCP 10.0.2.15 52.185.73.156 50251 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:23 ALLOW TCP 10.0.2.15 52.185.73.156 50252 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:24 ALLOW TCP 10.0.2.15 52.185.73.156 50253 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:24 ALLOW TCP 10.0.2.15 52.185.73.156 50254 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:25 ALLOW TCP 10.0.2.15 52.185.73.156 50255 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:25 ALLOW TCP 10.0.2.15 52.185.73.156 50256 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:25 ALLOW TCP 10.0.2.15 52.185.73.156 50257 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:26 ALLOW TCP 10.0.2.15 52.185.73.156 50258 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:27 ALLOW UDP 10.0.2.15 75.75.75.75 55016 53 0 - - - - - - - SEND
2025-10-01 20:58:27 ALLOW TCP 10.0.2.15 52.185.73.156 50259 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:29 ALLOW TCP 10.0.2.15 155.226.153.84 50260 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:29 ALLOW UDP 10.0.2.15 75.75.75.75 53222 53 0 - - - - - - - SEND
2025-10-01 20:58:29 ALLOW UDP 10.0.2.15 75.75.75.75 61750 53 0 - - - - - - - SEND
2025-10-01 20:58:29 ALLOW UDP 10.0.2.15 75.75.76.76 61750 53 0 - - - - - - - SEND
2025-10-01 20:58:29 ALLOW UDP 10.0.2.15 75.75.76.76 53222 53 0 - - - - - - - SEND
2025-10-01 20:58:29 ALLOW TCP 10.0.2.15 20.50.80.209 50261 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:30 ALLOW UDP 10.0.2.15 75.75.75.75 58838 53 0 - - - - - - - SEND
2025-10-01 20:58:30 ALLOW UDP 10.0.2.15 75.75.75.75 51282 53 0 - - - - - - - SEND
2025-10-01 20:58:30 ALLOW TCP 10.0.2.15 20.3.187.198 50262 443 0 - 0 0 0 - - - SEND
2025-10-01 20:58:40 ALLOW TCP fd17:625c:f037:2:5c9f:e14d:a852:b29d 2001:558:feed:443::166 50263 443
 0 - 0 0 0 - - - SEND
2025-10-01 20:58:41 ALLOW TCP fd17:625c:f037:2:5c9f:e14d:a852:b29d 2a06:98c1:310b::ac40:9bd1 50264
443 0 - 0 0 0 - - - SEND

PS C:\Users\jeyth\Desktop>
```

9. Recommendations

To sustain these improvements, the following recommendations are proposed:

- Deploy all scripts organisation-wide using Group Policy or Task Scheduler.
- Centralise log collection via a SIEM tool such as Splunk or Microsoft Sentinel.
- Train IT staff to interpret alerts and respond to flagged activities promptly.
- Conduct quarterly firewall audits to ensure evolving policy compliance.
- Integrate automated email notifications for real-time threat response.

These recommendations align with NIST CSF categories: Detect (DE.AE-1) and Protect (PR.AC-5).

They ensure that Wonderville’s network remains secure while being manageable by its limited IT staff.

10. Case Reflection

- Assumptions made include a fixed subnet and limited administrative privileges for remote staff.
- During implementation, challenges included configuring event log access and validating external traffic simulations.
- Nonetheless, the project provided practical insights into Windows system hardening using built-in tools.
- I learned how PowerShell scripts can automate complex administrative tasks, bridge visibility gaps, and support compliance with industry standards.
- This experience reinforced my understanding of defence-in-depth and the critical role of host-level controls in preventing lateral movement and privilege escalation.

11. References

1) Holmes, L. (2023). *Windows PowerShell Cookbook*. O’Reilly Media.
2) National Institute of Standards and Technology. (2020). *NIST SP 800-53 Revision 5: Security and Privacy Controls for Information Systems and Organisations*.

3) Centre for Internet Security. (2023). *CIS Critical Security Controls v8*.
Microsoft. (2023). *PowerShell Documentation*. Retrieved from https://learn.microsoft.com/en-us/powershell/

4) SANS Institute. (2022). *Host-Based Security Monitoring: Best Practices for Defenders*.