

---

# **The BTPS Security Package**

***Release 2020***

**Robert H. Osborne**

**Nov 17, 2023**



## CONTENTS:

<b>1</b>	<b>Functionality in The Blue Team PowerShell Security Package</b>	<b>5</b>
<b>2</b>	<b>Using the “microsoft-teams” branch repository</b>	<b>17</b>
<b>3</b>	<b>Using the Installer.ps1 File to Get Started</b>	<b>19</b>
3.1	Download PDF Instructions for Installer.ps1 . . . . .	20
<b>4</b>	<b>Using the Canary Files</b>	<b>21</b>
4.1	How To Set Up Your Canary Tokens . . . . .	21
<b>5</b>	<b>Configure WinRM over HTTPS</b>	<b>23</b>
5.1	Useful WinRM Info and Commands to Know . . . . .	23
5.2	Group Policy WinRM Settings . . . . .	24
<b>6</b>	<b>Configure Windows Event Forwarding</b>	<b>29</b>
6.1	Configure Source Initiated WEC Server . . . . .	30
6.2	Common Issues to Troubleshoot . . . . .	31
<b>7</b>	<b>Windows Event Forwarding (WEF) Application</b>	<b>35</b>
7.1	Perquisites and Setup Instructions . . . . .	37
7.2	File List Overview . . . . .	38
7.3	Setup the WEF Application . . . . .	38
7.4	Reference Links . . . . .	41
<b>8</b>	<b>Execute Scripts with Task Scheduler</b>	<b>43</b>
<b>9</b>	<b>Solo Sysmon Setup</b>	<b>45</b>
<b>10</b>	<b>Disclaimer</b>	<b>47</b>
<b>11</b>	<b>Indices and tables</b>	<b>49</b>



- OsbornePro Site
- GitHub Page
- GitLab Page
- PayPal Donations
- LiberPay Donations
- Report Issues

## Contributions

This is open source so if you wish to help contribute to the contents of this project feel free to reach out to me at [rosborne@osbornepro.com](mailto:rosborne@osbornepro.com) with your thoughts and ideas. For more general information on this feel free to refer to the [CONTRIBUTING](#) documentation.

**General Summary for this project can be read at** <https://github.com/OsbornePro/BTPS-SecPack/blob/master/README.md> This repo contains a collection of PowerShell tools that can be utilized to protect defend an environment based Microsoft's recommendations. This repo also assumes that you have referenced the Windows Event Logging Cheat Sheet for logging in your environment. Use [LOG-MD](#) or [CIS-CAT](#) to ensure the recommended logging is configured.

The [Installer.ps1](#) script is good to go. I created a virtual environment and ran everything from scratch to ensure you get the max protection and visibility possible with the least amount of fuss. If you experience any trouble please let me know so I am aware and can fix it. If you experience any issues or need help, feel free to reach out to me. My aim is to make this as easy to get going as possible. If something is too difficult or confusing please tell me about it. [rosborne@osbornepro.com](mailto:rosborne@osbornepro.com) I am still adding content to this site as it is fairly new.

## EMAIL AUTHENTICATION

I am now familiar with useage of the Azure Key Vault and will look at incorporating it into another repository for this package. Using the below code you can retrieve a Secret from the Azure Key Vault using a ServicePrincipalName and associated Certificate for Certificate authentication. The returned secret is used to create a Credential object in PowerShell for the Send-MailMessage cmdlet.

```
$Modules = "Microsoft.PowerShell.SecretManagement","Az.Accounts","Az.KeyVault"
ForEach ($Module in $Modules) {

    If (!(Get-Module -Name $Module -ListAvailable)) {

        Install-Module -Name $Module -Force

    } # End If

} # End ForEach
Import-Module -Name $Modules -Force -ErrorAction SilentlyContinue

$Thumbprint = "FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF" # Thumbprint of SPN AppIds
↪ authentication certificate
$AppId = "aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa" # Application ID containing the
↪ ServicePrincipal Name with a certificate attached to it
$DirectoryID = "kkkkkkkk-kkkk-kkkk-kkkk-kkkkkkkkkkkk" # Azure Key Vault Tenant ID

$ConnectionResult = Connect-AzAccount -Tenant $DirectoryID -ApplicationId $AppId -
↪ CertificateThumbprint $Thumbprint -ServicePrincipal
$KeyVaultName = 'SMTPAccount'
$SecretName = "AlertPassword"
```

(continues on next page)

(continued from previous page)

```
# BUILDS A LOCAL VAULT THAT USES AZURE AUTH
#$RegisterVault = Register-SecretVault -Name $KeyVaultName -ModuleName Az.KeyVault -
↪VaultParameters @{AZKVaultName=$AzKeyVaultName; SubscriptionId = 'ssssssss-ssss-ssss-
↪ssss-ssssssssssss'} -ErrorAction SilentlyContinue

$EmailPassword = Get-Secret -Vault $KeyVaultName -Name $SecretName # Returns a Secure_
↪String instad of clear text preventing clear text from entering memory
$EmailCredential = New-Object -TypeName System.Management.Automation.PSCredential -
↪ArgumentList @("support@transmedics.com", $EmailPassword)

Send-MailMessage -To $To -From $From -Credential $EmailCredential -Body "Test" -Subject
↪"Example" -SmtpServer smtp.office365.com -Port 587 -UseSSL
```

## EXPAND THIS PROJECT BY UTILIZING ELASTICSEARCH

- **ELK SIEM Tool:** I was going to set up a configuration for the ELK SIEM tool, but they make changes so often that it is too much work to keep up with. I do not receive donations for this project which discourages me from putting that kind of time in. The Elasticsearch tool is free for certain uses and offers a purchase if desired. It includes [Elasticsearch](#), [Kibana](#), and [Elastic Agent](#). The configuration should use the source collection Windows Event Forwarding (WEF) configuration to collect logs using WinRM over HTTPS directly on the Elasticsearch sever, which then locally imports the local Sysmon and local Forwarded Event logs into Elasticsearch. The purpose of this is to prevent the need to install agents on the devices in your environment. The free version does not offer LDAP authentication unfortunately. The configuration will use TLS certificates to encrypt communications on the local host and listen for outside connections.
- I am **NO** longer planning to integrate the [Virus Total API](#) for MD5 hash comparisons. This does not provide enough cost per value; however, I included a script to do this in case it is valuable to your situation. The script is located here: <https://github.com/OsbornePro/BTPS-SecPack/blob/master/Sysmon/HashValidator.ps1> and can be used if desired. This will be more beneficial in smaller environments I would think.

**IMPORTANT:** This **Blue Team PowerShell Security Package**, assumes that you have referenced the [Windows Event Logging Cheat Sheet](#) for logging in your environment. Use [LOG-MD](#) or [CIS-CAT \(an SCAP Tool\)](#) to ensure the recommended logging is configured. These logging recommendations adhere to commonly accepted guidelines in the cyber security community. Even without the use of this security application, these guidelines should be followed to better assist your organization in the event of a compromise.

**CODE CONTRIBUTIONS** I am always open to suggestions and ideas as well as contributions if anyone wishes to help add to this package. Credit will of course be given where credit is due. If you wish to contribute, I have placed info on that [HERE](#).

## What Purpose Does This Serve?

This repository contains a collection of PowerShell tools that can be utilized to protect and defend an environment based on the recommendations of multiple cyber security researchers at Microsoft. These tools were created with a small to medium size mostly Windows environment in mind as smaller organizations do not always have the type of funding available to overly spend on security. The goal of this project lines up with the goals of [OsbornePro LLC](#). This exists to help add value to a smaller organization's security by creating more visibility for the IT Administrator or Security Team.

## IMPORTANT NOTE FOR LARGE ENVIRONMENTS

For the case of organizations with 1,000's of devices; you may find that this entire suite does not apply to you. This has to do with how some of the discoveries operate. For example, the alert I have in the [Device Discovery](#) directory relies on DHCP assigned IP addresses. All DHCP servers in an environment are queried to create a list of known MAC addresses. This information is then saved to a CSV file for reference in discovering any new devices that join a network. This file could become too large to be effective. The other alert I can see not being effective is the "[Local](#)

**Port Scan Alert**". This is because if there is an overabundance of connections the script will not be able to cover all the connections quickly enough. Other alerts in this security package are still appropriate no matter the network size as they are Event ID based typically. To begin, I suggest setting up WinRM over HTTPS in your environment.





## FUNCTIONALITY IN THE BLUE TEAM POWERSHELL SECURITY PACKAGE

**Account Lockout Notification** This alert lets you know when a user account has been locked-out. There is also an alert to be notified when a locked-out account has been manually unlocked.

AD Event: Account Lockout

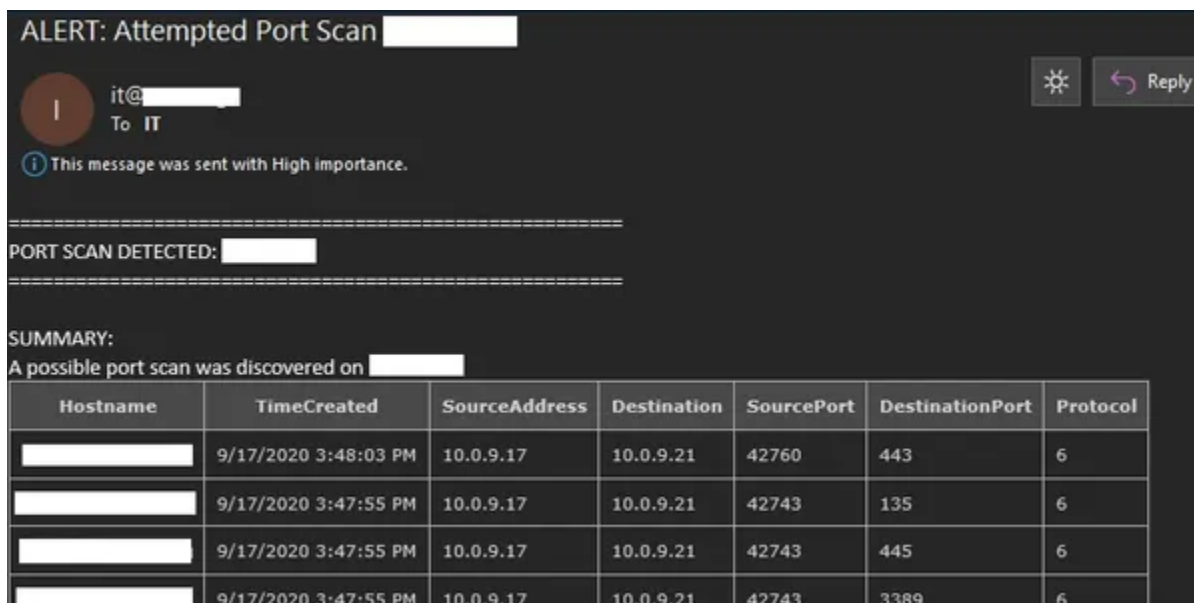
it@ [redacted]  
To: IT

The below table contains information on the user whose account was locked out.

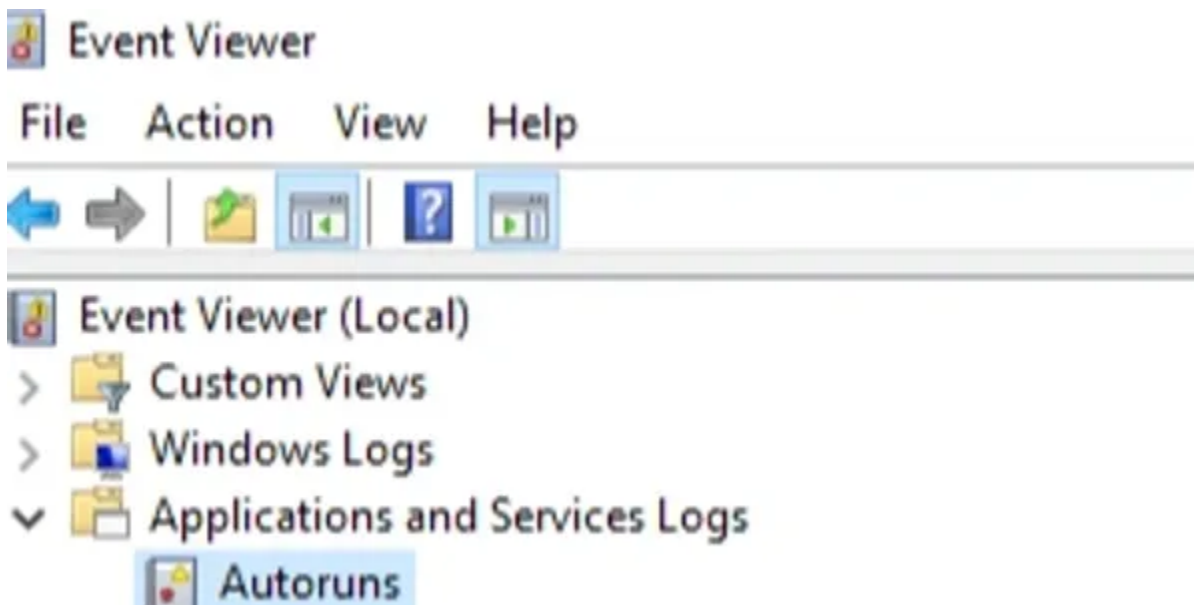
EventID	User	MachineName	ProcessID	SID
4740	[redacted]	[redacted].org	672	S-1-5-[redacted]

*This Message was Sent on 09/20/2020 21:22:20*

**Attempted Port Scan** This alert informs the cyber security team when a port scan has been attempted against a server. This currently does not work for File Servers or VoIP servers which have hundreds of unique IP addresses connecting a minute.



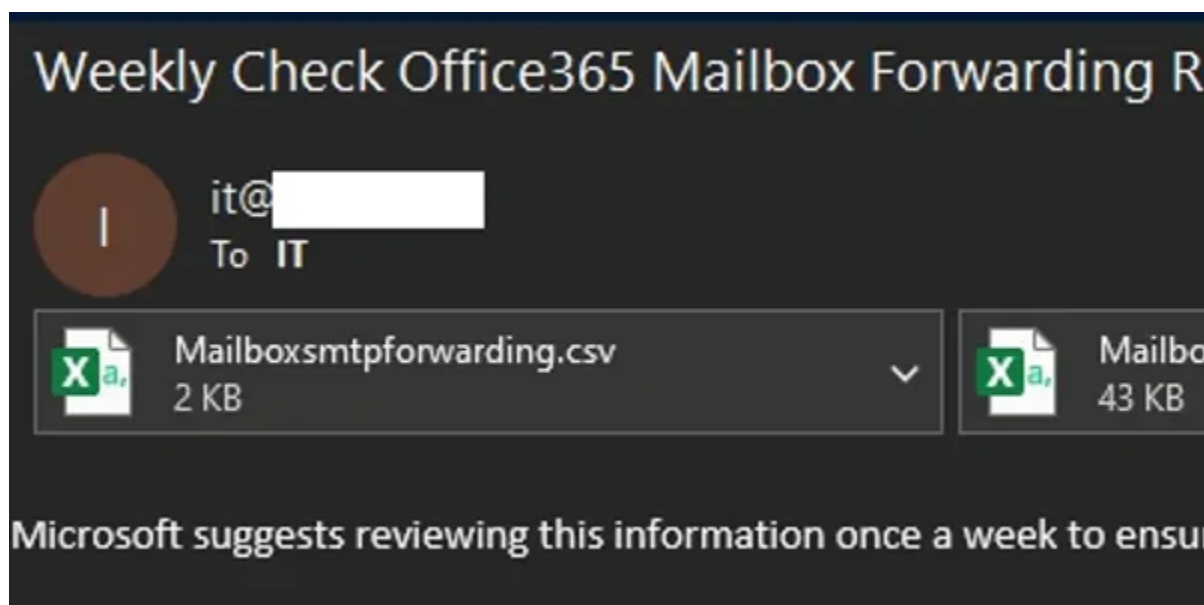
**AutoRuns Logging** This is showing the Autoruns Event Viewer entry that gets created. An easy-to-read CSV file also gets saved to C:\Program Files\AutorunsToWinEventLog\AutorunsOutput.csv Thanks to: <https://github.com/palantir/windows-event-forwarding/tree/master/AutorunsToWinEventLog>



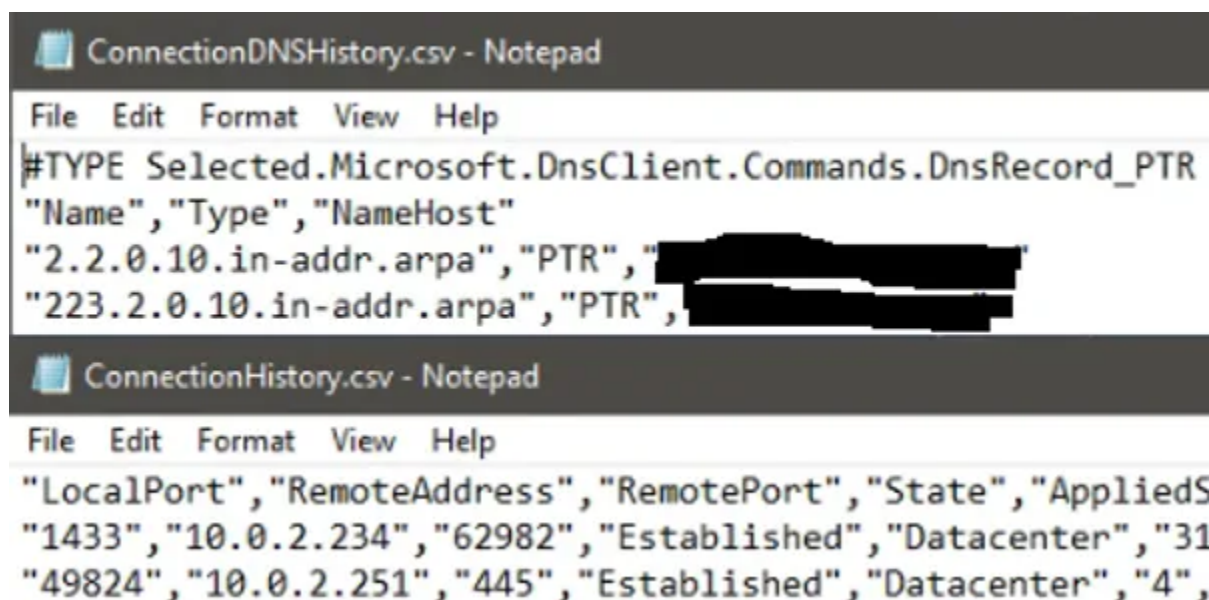
**Blacklisted IP Logging** When a device establishes a connection to an IP that is on 4 or more blacklists or a domain that is less than 2 years old you will be informed. The events are also stored in the Event Viewer under “**MaliciousIPs**”. An IP address on 3 blacklists does not necessarily mean it is dangerous or something to do anything about. Correct me if my analysis is wrong.

TimeCreated	MachineName	Significance	Message	Id
10/6/2020 10:51:00 AM	[REDACTED]	Connection to an IP that is on a blacklist or a domain less than 2 years old	IP Address was found to be on the following Blacklists: IP Address: 128.116.114.3 Blacklist: dnsbl-1.uceprotect.net Network connection detected: RuleName: UtcTime: 2020-10-06 16:47:15.584 ProcessGuid: {84d771a9-9c51-5f7c-0000-001047098185} ProcessId: 35100 Image: C:\Users\[REDACTED]\AppData\Local\Roblox\Versions\version-181941c15130443b\RobloxPlayerBeta.exe User: [REDACTED] Protocol: tcp Initiated: true SourceIsIpv6: false SourceIp: 192.168.1.15 SourceHostname: [REDACTED] SourcePort: 51612 SourcePortName: DestinationIsIpv6: false DestinationIp: 128.116.114.3 DestinationHostname: DestinationPort: 443 DestinationPortName: https	1
10/6/2020 10:36:00 AM	[REDACTED]	Connection to an IP that is on a blacklist or a domain less than 2 years old	IP Address was found to be on the following Blacklists: IP Address: 128.116.114.3 Blacklist: dnsbl-1.uceprotect.net Network connection detected: RuleName: UtcTime: 2020-10-06 16:37:44.312 ProcessGuid: {84d771a9-9c51-5f7c-0000-001047098185} ProcessId: 35100 Image: C:\Users\[REDACTED]\AppData\Local\Roblox\Versions\version-181941c15130443b\RobloxPlayerBeta.exe User: [REDACTED] Protocol: tcp Initiated: true SourceIsIpv6: false SourceIp: 192.168.1.15 SourceHostname: [REDACTED] SourcePort: 51369 SourcePortName: DestinationIsIpv6: false DestinationIp: 128.116.114.3 DestinationHostname: DestinationPort: 443 DestinationPortName: https	1
		Connection to an	IP Address was found to be on the following Blacklists: IP Address: 85.97.201.35 Blacklist: b.barracudacentral.org Network connection detected: RuleName: UtcTime: 2020-10-06 16:07:47.503 ProcessGuid: {87f38d9d-9602-5f7c-0000-	

**Check Forwarding Rules** This alert is meant to come in once a week. It allows administrators to easily view any email forwarding rules and ensure that no information is being compromised unknowingly.



**Track Network Connection History** ListenPortMonitor.ps1's main goal is to keep an eye open for newly opened ports on a server to discover possible Reverse Shells or Bind Shells. A connection history log is kept helping trace connections that have been established with a server.

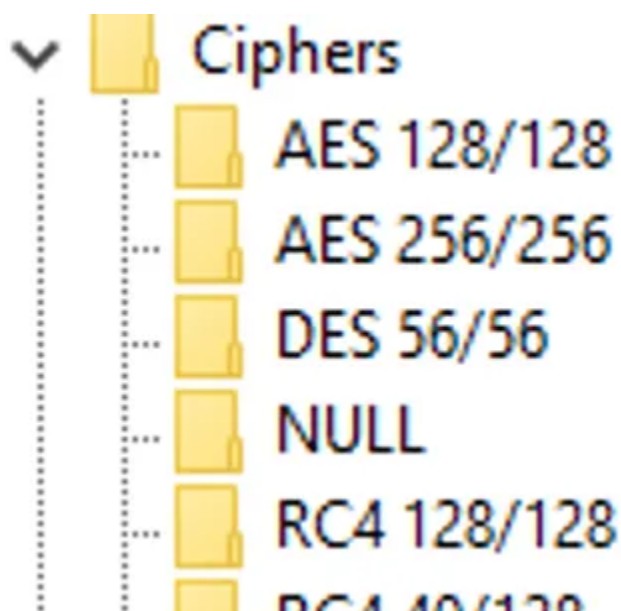


The image shows two Notepad windows. The top window, titled 'ConnectionDNSHistory.csv - Notepad', contains a CSV header row: `#TYPE Selected.Microsoft.DnsClient.Commands.DnsRecord_PTR` and `"Name","Type","NameHost"`. Below the header are two rows of data: `"2.2.0.10.in-addr.arpa","PTR","[REDACTED]"` and `"223.2.0.10.in-addr.arpa","PTR","[REDACTED]"`. The bottom window, titled 'ConnectionHistory.csv - Notepad', contains a CSV header row: `"LocalPort","RemoteAddress","RemotePort","State","AppliedS"`. Below the header are two rows of data: `"1433","10.0.2.234","62982","Established","Datacenter","31"` and `"49824","10.0.2.251","445","Established","Datacenter","4"`.

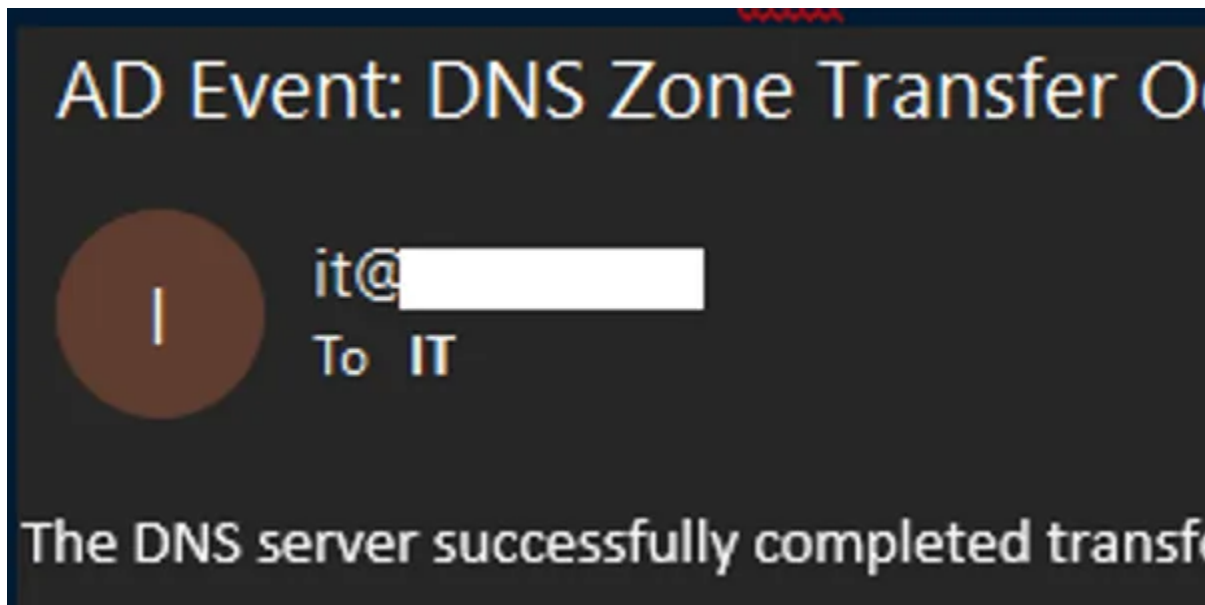
```
File Edit Format View Help
#TYPE Selected.Microsoft.DnsClient.Commands.DnsRecord_PTR
"Name","Type","NameHost"
"2.2.0.10.in-addr.arpa","PTR","[REDACTED]"
"223.2.0.10.in-addr.arpa","PTR","[REDACTED]"

File Edit Format View Help
"LocalPort","RemoteAddress","RemotePort","State","AppliedS
"1433","10.0.2.234","62982","Established","Datacenter","31
"49824","10.0.2.251","445","Established","Datacenter","4",
```

**Disable Weak SSL Ciphers** Hardening cmdlet allows you to quickly and easily disable weak TLS protocols and associated ciphers to prevent weak encryption methods from being used.

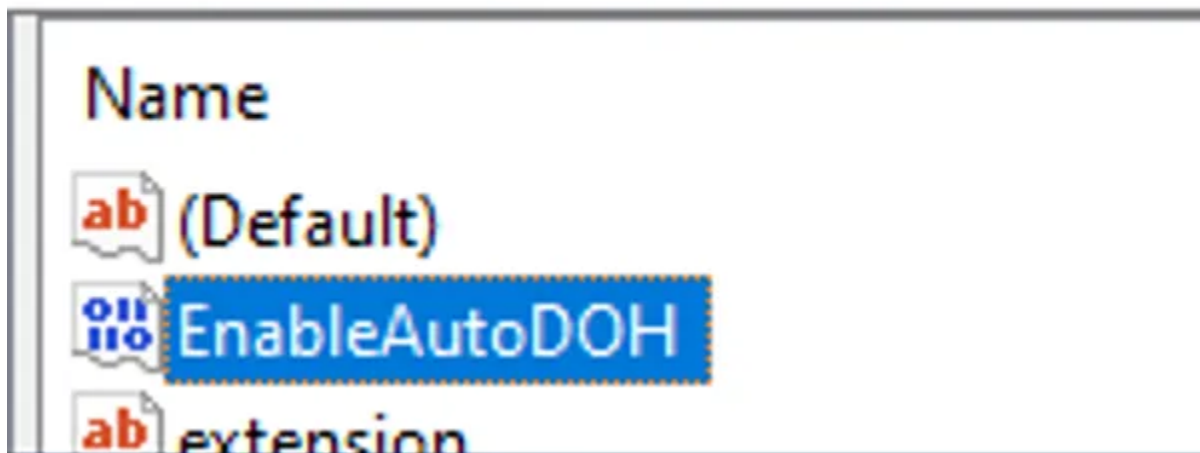


**DNS Zone Transfer Alerts** An executed DNS Zone transfer triggered this alert. Normal DNS server communication does not trigger this alert. Manually requesting a zone transfer will, allowing you to discover any attackers looking to learn about your environment.



**Enable DNS over HTTPS** This hardening cmdlet allows you to easily enable or if desired disable DNS over HTTPS on any Windows 10+ Device in your environment

services\Dnscache\Parameters



**Insecure LDAP Bind Notifications** This alert is to let you know when an unencrypted LDAP Bind occurs. This helps to implement LDAP over SSL in an environment and discovers possible LDAP enumeration attempts from an attacker.

### AD Event: Insecure LDAP Binds Performed

it@ [redacted]  
To [redacted]  
Cc [redacted]

The below table contains information on connections to LDAP over the last 24 hours that did not use SSL.

IPAddress	Port	User	BindType
10.0.2.223	53717	[redacted]	Simple
10.0.2.223	53701	[redacted]	Simple

**New Device Joined Network Discovery** This alert is displaying 4 devices that recently were physically plugged into an Ethernet cable or joined the Wi-Fi network. Now that these devices are known they will not be detected again unless you remove them from the CSV file containing device history info.

### AD Event: New Device Check [redacted] DHCP

it [redacted]  
To IT

This is a list of the newest devices to have joined the USAV Network.

Vendor	HostName	IPAddress	ClientId
Wistron InfoComm(Kunshan)Co, Ltd	[redacted]	10.0.9.23	54-ee-75-62-db-38
Intel Corp	[redacted]	10.0.10.26	e4-f8-9c-c1-f6-7e
Intel Corp	[redacted]	10.0.10.27	00-e1-8c-8b-e4-4e
Intel Corp	[redacted]	10.0.10.96	0c-8b-fd-c5-f1-70

**User Changed Password Notification** Receive an alert whenever a user has changed their password. Compare this value to the Administrator list you receive. This allows you to notice any passwords changing outside normal conditions.

AD Event: Password Change Attempt


 it@  
To IT

The below table contains information on a user whose password was changed.

EventID	User	MachineName	SID
4723		org	S-1-5-

**Admin Changed Another Users Password Notification** This alert informs you when a user's password has been changed by an Administrator or other user. Allowing user to change passwords through Azure will generate this alert as well.

AD Event: Password Change Attempt

 it@  
To IT

The below table contains information on a user whose password was attempted to be changed by another user

EventID	EffectuatedUser	ExecutingUser	MachineName	Date	
4724			org	9/23/2020 4:14:03 PM	An attempt

*This Message was Sent on 09/23/2020 16:14:17*

**List of Expiring Passwords Notification** Receive a list of users whose passwords are expiring in the next 2 weeks. This helps to keep tabs on expected password changes. Someone changing their password not on this list may be under attack.



IT@ [redacted]  
To IT

FYI,


The below table contains info on the users who have received a password expiring notification.

Displayname	Mail	ExpiryDate
B [redacted]	[redacted]@org	9/23/2020 9:32:11 AM
B [redacted]	[redacted]@org	10/5/2020 7:14:58 AM
N [redacted]	[redacted]@org	10/3/2020 7:45:56 AM

**Local Port Scan Notification** This alert informs the cyber security team when a port scan has been attempted against a server. This currently does not work for File Servers or VoIP servers which have hundreds of unique IP addresses connecting a minute. This works as is but still requires some fine tuning. To trigger this alert I have had to perform aggressive nmap scans.

ALERT: Attempted Port Scan [redacted]

it@ [redacted]  
To IT

 This message was sent with High importance.

=====

PORT SCAN DETECTED: [redacted]

=====

SUMMARY:

A possible port scan was discovered on [redacted]

Hostname	TimeCreated	SourceAddress	Destination	SourcePort	DestinationPort	Protocol
[redacted]	9/17/2020 3:48:03 PM	10.0.9.17	10.0.9.21	42760	443	6
[redacted]	9/17/2020 3:47:55 PM	10.0.9.17	10.0.9.21	42743	135	6
[redacted]	9/17/2020 3:47:55 PM	10.0.9.17	10.0.9.21	42743	445	6
[redacted]	9/17/2020 3:47:55 PM	10.0.9.17	10.0.9.21	42743	3389	6

**Remediate a compromised Office365 account** This is some output showing the results of what happens when the `RemediateCompromisedOfficeAccount.ps1` is run. For more information on what this does click [HERE](#).



```

[redacted] account will have remediation actions applied to
An audit report will be saved to C:\Users\Public\Desktop\[redacted].Rem
WARNING: Using New-PSSession with Basic Authentication is going to be de
WARNING: The names of some imported commands from the module 'tmp_lmyns
import-Module command again with the Verbose parameter. For a list of app
Connecting to EOP Powershell Service
WARNING: Your connection has been redirected to the following URI: "http
WARNING: The names of some imported commands from the module 'tmp_4lxvxu
import-Module command again with the Verbose parameter. For a list of app
Password for the user [redacted] was changed to VZ*Q3[]ZGa/_l{
password on the next logon.
Mailbox auditing for user is being enabled...
Current auditing configuration.
Removing Mailbox Delegate Permissions for the affected user [redacted]
Disabling mailforwarding rules to external domains for the affected user
Found the following rules that forward or redirect mail to other account
WARNING: The command completed successfully but no settings of '[redacted]
Mailbox forwarding removal completed. Current configuration is:

```

**Suspicious Event Occurred** This alert is triggered using high priority centralized logs from Windows Event Forwarding that are imported into a SQL server. A System Update, when run on Lenovo computers as a standard user, will create a temp account, add it to the local administrators group and install missing updates. It then removes the created account from the local Administrators group and deletes the account. If this were an attacker trying to cover their tracks this would catch it.

**SUSPICIOUS EVENT TRIGGERED**

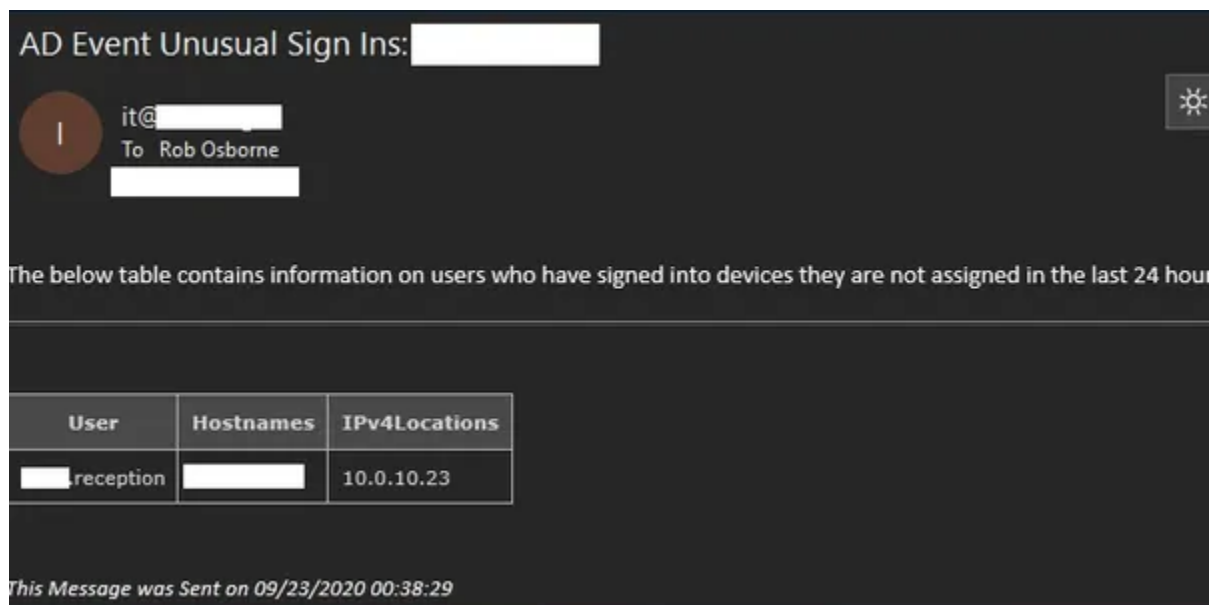
it@ [redacted]  
To IT

Thu 9/24/2020 12:00

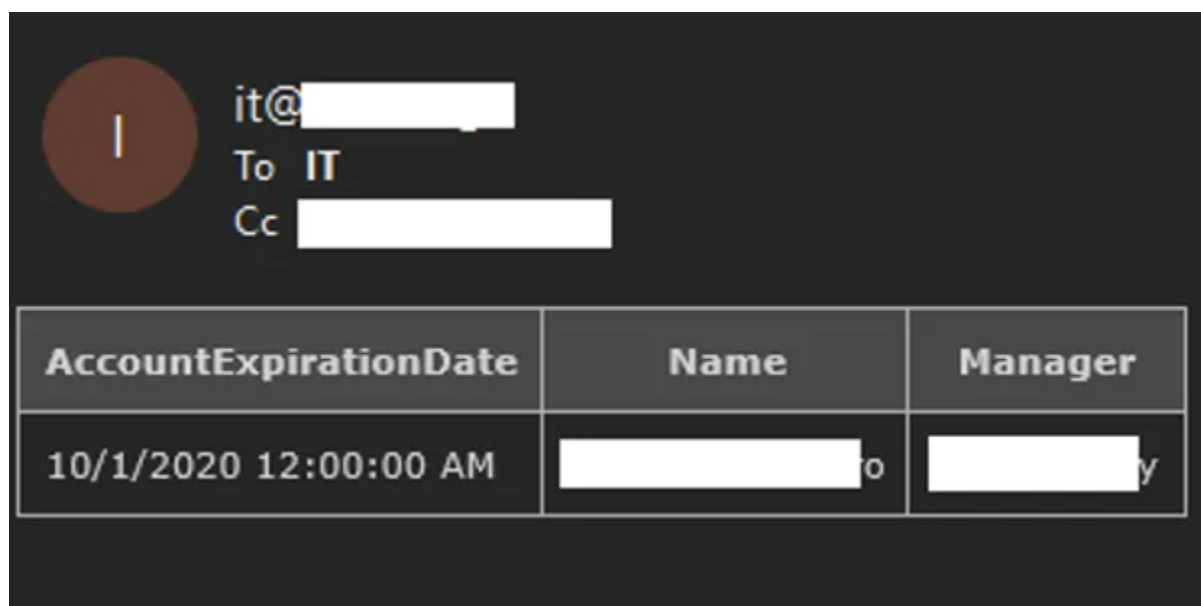
The below table contains suspicious events that were triggered

TimeCreated	MachineName	Significance	Message	Id
9/24/2020 11:08:00 AM	[redacted]	User Added to Privileged Group	A member was added to a security-enabled local group. Subject: Security ID: S-1-5-18 Account Name: [redacted] Account Domain: [redacted] Logon ID: 0x3E7 Member: Security ID: [redacted] Account Name: - Group: Security ID: S-1-5-32-544 Group Name: Administrators Group Domain: Builtin Additional Information: Privileges: -	4732
9/24/2020 11:09:00 AM	[redacted]	User Removed from Privileged Group	A member was removed from a security-enabled local group. Subject: Security ID: S-1-5-18 Account Name: [redacted] Account Domain: [redacted] Logon ID: 0x3E7 Member: Security ID: [redacted] Account Name: - Group: Security ID: S-1-5-32-544 Group Name: Administrators Group Domain: Builtin Additional Information: Privileges: -	4733
9/24/2020 11:08:00 AM	[redacted]	User Account Created	A user account was created. Subject: Security ID: S-1-5-18 Account Name: [redacted] Account Domain: [redacted] Logon ID: 0x3E7 New Account: Security ID: [redacted] Account Name: lenovo_tmp_igojTFYU Account Domain: [redacted] Attributes: SAM Account Name: lenovo_tmp_igojTFYU Display Name: <value not set> User Principal Name: - Home Directory: <value not set> Home Drive: <value not set> Script Path: <value not set> Profile Path: <value not set> User Workstations: <value not set> Password Last Set: <never> Account Expires: <never> Primary Group ID: 513 Allowed To Delegate To: - Old UAC Value: 0x0 New UAC Value: 0x15	4720

**Unusual Sign In Alert** This alert is letting the recipient know the reception account signed into a device outside of its normal assignments

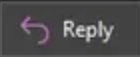




**User Account Expiring Notification** This alert lets IT Administrators know when a user account is about to expire.



**User Account Unlocked** This alert informs administrators when a user account has been manually unlocked. It provides information on who unlocked the account.

Unlocked

 Reply Reply

Information on the user whose account was unlocked.

MachineName	ProcessID	SID	Date
[redacted]org	672	S-1-5-21-[redacted]	9/20/2020 9:24:20 PM

10/2020 21:24:30



## USING THE “MICROSOFT-TEAMS” BRANCH REPOSITORY


If you wish to use Microsoft Teams for sending alerts instead of using email you will want to load the [microsoft-teams](#) branch for this repository which has the Teams alert modifications. It will take me a little while to implement this as a configuration option in the install script. Using Microsoft Teams for alert posts does not remove the need for certain email notifications in this repository. Email will still be used for some of the actions. These Teams Posts can be completed after you first create a webhook. So-called webhooks offer the possibility to send alerts or other notifications to a Microsoft Teams channel.

- [Microsoft Documentation to Create a Webhook](#)
- [Microsoft Documentation on Using Webhooks](#)

You can create a webhook using the following steps (if you are an admin)

1. Open the Microsoft Teams application
2. In the left-hand pane click **Teams**
3. Click the more options icon represented by 3 dots ... next to one of the desired Teams Channels. Example Channel Name: General
4. Clicking those 3 dots will display a dropdown menu. Click **Connectors**
5. Click the **Add** button next to **Incoming Webhook**
6. Click **Add** on the Incoming Webhook pop up screen
7. **On the Incoming Webhook screen perform the following actions**
  1. Define a name for your webhook. In the below image this value is *PowerShell-TeamsMessagePost*
  2. Click **Create**
  3. Optionally you can also use **Upload Image** to select an image for the Incoming Webhook. In the below image I left the default icon which is the light blue triangle on white background.
  4. Click **Create**
8. The Incoming Webhook URL is created. Copy the URL and click **Done**

You can now use the above URL in the B.T.P.S Security Package scripts I provide. You can quickly update the value in the scripts by executing the below commands



PowerShell-TeamsMessagePost 2:50 PM

### Password Change Attempt

DC =   
User =   
SID = S-1-5-21----  
Time = Monday, August 23, 2021  
Event ID = 4723

[← Reply](#)

```

$WebHook = Read-Host -Prompt "Paster your Webhook URL here: "
$SIEM = Read-Host -Prompt "If you have a SIEM in your environment enter the link here: "
$BTPSHome = Read-Host -Prompt "Where did you save the BTPS Security Pacakge git repo?
→EXAMPLE: C:\Users\Administrator\Downloads\BTPS-SecPack-microsoft-teams"
$Files = (Get-ChildItem -Path $BTPSHome -Include "AttemptedPasswordChange.ps1",
→"AttemptedPasswordReset.ps1","Failed.Username.and.Password.ps1","User.Account.Created.
→ps1 ","User.Account.Locked.ps1","User.Account.Unlocked.ps1","DNSZoneTransferAlert.ps1",
→"NewComputerAlert.ps1","Query-InsecureLDAPBinds.ps1","UnusualUserSignInAlert.ps1",
→"Watch-PortScan.ps1 " -Recurse -ErrorAction SilentlyContinue -Force).FullName
ForEach ($File in $Files) {
    ((Get-Content -Path $File -Raw) -Replace "WEBHOOK_URL_REPLACE","$WebHook") | Set-
→Content -Path $File -Force
    ((Get-Content -Path $File -Raw) -Replace "SIEM TOOL LINK","$SIEM") | Set-Content -
→Path $File -Force
} # End ForEach

```

## USING THE INSTALLER.PS1 FILE TO GET STARTED

I wrote the `Installer.ps1` script allow anyone to quickly and easily install as many protections as possible offered by the B.T.P.S. Security Package. Running this script requires very minimal to zero know how. You do not need to know how to download the Git repository. `Installer.ps1` will do it for you :-)

**How can I get started using the `Installer.ps1` install file?** Here is what you need to do in order to execute this file.

1. Log into your Primary Domain Controller using an account with Administrator permissions.
2. Open an Administrative PowerShell session (**Windows Key + X, The press A**).
3. Execute the command in step 4. This can be done by highlighting the command. Right click the highlighted text and select "COPY". Then Right Click inside your PowerShell window. If this does not paste right away you can paste by doing the key combo (**Ctrl + V**). This command executes all the text on that webpage inside of your PowerShell session without downloading the file to your disk drive.
4. `IEX (New-Object -TypeName System.Net.WebClient).downloadString('https://raw.githubusercontent.com/OsbornePro/BTPS-SecPack/master/Installer.ps1')`
5. The installation of the B.T.P.S Security Package should then start. Some Next Generation Anti-Virus providers may block script execution in this manner. If that is the case, use the below method to accomplish the same task.

### IF ABOVE COMMAND METHOD DOES NOT WORK

Some Endpoint Detection and Response (EDR) and Next Generation Anti-Virus providers may block script execution in this manner. If that is the case use the below method to accomplish the same task.

1. Log into your Primary Domain Controller using an account with Administrator permissions.
2. Open an Administrative PowerShell session (**Windows Key + X, The press A**).
3. The command displayed in step 4 will download the script to your disk in your Downloads directory. Copy and paste the command into your admin PowerShell session and press **ENTER** to execute it.
4. `Invoke-WebRequest -Uri "https://raw.githubusercontent.com/OsbornePro/BTPS-SecPack/master/Installer.ps1" -OutFile "$env:USERPROFILE\Downloads\Installer.ps1"`
5. Execute the command line in step 6 to ensure your Execution Policy allows the script to execute easily. Copy and paste the command into your admin PowerShell session and press **ENTER** to execute it.
6. `Set-ExecutionPolicy RemoteSigned -Force`
7. Execute the command line in step 8 to run the script and being installation. Include the period at the beginning of the command. Copy and paste the command into your admin PowerShell session and press **ENTER** to execute it.
8. `."$env:USERPROFILE\Downlods\Installer.ps1"`
9. The installation of the B.T.P.S. Security Package should then begin.

### OTHER DOWNLOAD FILE COMMANDS

As an FYI there are multiple ways to download files from the PowerShell session. If `Invoke-WebRequest` is blocked or does not work for you, one of the below commands may be able to work instead. Each command does the same thing in a different way and each command is one line.

- NET Method

```
(New-Object Net.WebClient).DownloadFile('https://raw.githubusercontent.com/OsbornePro/BTPS-SecPack/master/Installer.ps1', "$env:USERPROFILE\Downloads\Installer.ps1")
```

- Bits Transfer Method

```
Start-BitsTransfer "https://raw.githubusercontent.com/OsbornePro/BTPS-SecPack/master/Installer.ps1" -Destinations "$env:USERPROFILE\Downloads\Installer.ps1"
```

- Certutil Method

```
certutil.exe -urlcache -split -f https://raw.githubusercontent.com/OsbornePro/BTPS-SecPack/master/Installer.ps1 "$env:USERPROFILE\Downloads\Installer.ps1"
```

- Bitsadmin Method

```
bitsadmin /transfer debjob /download /priority normal https://raw.githubusercontent.com/OsbornePro/BTPS-SecPack/master/Installer.ps1 "$env:USERPROFILE\Downloads\Installer.ps1"
```

## 3.1 Download PDF Instructions for Installer.ps1

Below link contains images and walks you through the `Installer.ps1` setup steps and process. I recommend checking this out if you are executing the installer script to make sure there are no misunderstandings during its execution.

<https://github.com/OsbornePro/Documents/raw/main/Installer.ps1%20Demo.pdf>



## USING THE CANARY FILES

I have included fake executables in the BTPS Security Package. These executables do not do anything other than print the help message info from the actual executables. This was done to make them look legitimate if an attacker attempts to execute them. By making these fake executables **Canary Files** we can receive email alerts whenever an attacker executes them. This also works if the attacker downloads the executable to their attack machine and runs the file. (As long as they have internet you should get an email notification)

### 4.1 How To Set Up Your Canary Tokens

1. Simply go to <https://www.canarytokens.org/generate>
2. Select **“Custom exe / binary”** from the dropdown menu
3. Enter the email address to send an alert notification too
4. Set a Reminder to let you know what host you are placing this file on, the name of the fake executable file. This way your alerts will tell you where and what was executed
5. Click **“Generate Canary Token”** to download your new decoy executable file
6. Save the file on a device. I suggest creating a new executable for each device you plan on placing this executable file on. This is to ensure you know where the file was executed from. *This is a free tool\**

**Q.** Where should I save the Canary File?

**A.** Anywhere that makes sense to you. I have included a couple example file locations below with a note on why that location might be good.

- C:\Temp Common directory for storing files an admin may want to delete later but never did
- C:\Windows\Temp` Common directory for storing files an admin may want to delete later but never did
- C:\Windows\System32` In your Path variable to make files easier to execute
- C:\Users\Public\Downloads` Common place for downloaded executables
- C:\Users\Administrator\Downloads` Common place for downloaded executables
- C:\Windows\System32\spool\drivers\color` Commonly used by attackers to save files under the System32 directory tree

Use PowerShell to create a fake custom save location for Microsoft Edge Temp files. When you click “Open” in Microsoft Edge this is where those temporarily saved file locations are placed

```
$Guid = [guid]::NewGuid()
New-Item -Path "$env:USERPROFILE\AppData\Local\Temp\MicrosoftEdgeDownloads" -Name $Guid -
↪ItemType Directory -Force
```

### List of Included Fake Executables

Below is a list of the executables I have included to try and bait an attacker into using.

- `accesscheck.exe` Used for viewing permissions on files and discovering unquoted service paths
- `nc.exe` and `nc64.exe` Used to execute bind and reverse shells or for transferring files
- `procdump.exe` Used for dumping process memory which may contain clear text passwords or other info
- `PSEXEC.exe` Used for executing commands on remote devices using SMB

If any of the above executables are run, they will display the actual executable's help message making it seem legitimate. The goal of this is to trick an attacker into thinking their command line is bad or someone messed up the executables compilation. In the process, we will receive an email alert thanks to the canary token.

## CONFIGURE WINRM OVER HTTPS

I posted a YouTube video covering the settings configured for WinRM over HTTPS communication using Group Policy. These settings can be seen in the sections below.

[YouTube Video : Configure WinRM over HTTPS Instructions](#)

### 5.1 Useful WinRM Info and Commands to Know

Setup WinRM over HTTPS may require you to know a few commands. I have included these commands below.

- `winrm invoke Restore winrm/Config`
- `Enable-PSRemoting -Force` # Enables firewall rules for WinRM
- `winrm qc -q` # Quick config for WinRM 5985
- `winrm enum winrm/config/listener` # Enumerate cert thumbprint used on different winrm ports
- `winrm delete winrm/config/listener?Address=*&Transport=HTTPS` # Delete winrm certificate and stop listener on 5986. This allows new cert to be attached to port
- `winrm create winrm/config/listener?Address=*&Transport=HTTPS` # Creates a WinRM listener on 5986 using any available certificate

The below command defines a certificate to use on port 5986. Certificate Template needed is a Web Server certificate from Windows PKI

```
New-WSManInstance -ResourceUri WinRM/Config/Listener -SelectorSet @{Address = "*";  
→Transport = "HTTPS"} -ValueSet @{Hostname = FqdnRequiredHere.domain.com;  
→CertificateThumbprint = $Thumbprint }
```

#### CERTIFICATE SHOULD BE USED FOR SERVER AND CLIENT AUTHENTICATION WHEN USING WINDOWS EVENT COLLECTION

Another thing you will need to do to use Windows Event Collection is below

1. Log onto your Windows Event Collection Server with an administrator account
2. Open the local machine certificate store (certlm.msc)
3. Drop down Certificates - Local Computer > Personal > Certificates
4. Right click on your WinRM Certificate and go to “All Tasks” > “Manage Private Keys...”
5. Go to the “Security” tab and click the “Add” button
6. Give the NETOWRK SERVICE user “Full Control” to the private key and click Apply

### SERVER CERTIFICATE INFO:

The certificate thumbprint value that you are going to need in “**Group Policy Setting 1**” below, is from your internal domains Private Key Infrastructure (PKI). This value will vary as these values are unique to the certificate. The Root Certificate Authority (CA) assigns certificates to your devices. When a device receives a certificate, it gets assigned under the Root CA’s certificate. This creates what is called a Certificate Chain. If it helps to see this represented in a directory tree format, view the below tree structure. Your domain would not have an Intermediate CA most likely but I included it for the visual.

- **Root CA Certificate** <– *This is the certificate thumbprint you need*
  - **Intermediate CA Certificate**
    - \* Assigned Device Certificate
    - \* Assigned Device Certificate <– *This certificate’s thumbprint gets assigned to port 5986 on the client device*

Add a friendly name to your WinRM over HTTPS server’s certificate. I do this because the code that performs a lookup operation on some OS versions of Windows does not know how to retrieve the friendly name of a certificate in a PKCS#7 file.

**REFERENCE:** <https://docs.microsoft.com/en-us/troubleshoot/iis/error-install-certificate>

### CLIENT CERTIFICATE INFO:

- In the above tree, “**Assigned Device Certificate**” is where the below command would be used

```
New-WSManInstance -ResourceUri WinRM/Config/Listener -SelectorSet @{Address = ".*";  
→Transport = "HTTPS"} -ValueSet @{Hostname = FqdnRequiredHere.domain.com;␣  
→CertificateThumbprint = $Thumbprint }
```

- Notice the “**Hostname**” value includes the domain you are in. This needs to also be true for the “**Common Name**” value when the client device requests the WinRM certificate. This means your CN value is required to be devicename.domainname.com. If you do not include the domain name in the Common Name value your WinRM over HTTPS communication will not work. Subject Alternative Name’s (SAN) will not work either. I have tried adding more than one Common Name values to a certificate and this communication still failed.
- If you are using a Windows Server Certificate Authority, the “**WebServer**” certificate template can be used to request the certificate needed.

## 5.2 Group Policy WinRM Settings

### GROUP POLICY SETTING 1

In your group policy settings go to “**Computer Configuration > Preferences > Control Panel Settings > Services**”. Then right click and add a New Service.

Set the “**Startup Type**” to “**Automatic**”.

Set the “**Service Name**” to **WinRM**, set the “**Service Action**” to “**Start Service**”, set the “**Wait Timeout**” to **30** seconds.

In the recovery tab select “**Restart the Service**” from the three failure options.

Set “**Reset Fail Count after**” to **0** days and “**Reset Service after**” to **1** minutes.

Leave the rest of the values as their defaults and click **OK** to save.

### GROUP POLICY SETTING 2

Next, still on the same policy object, is the list of IP addresses that are allowed to do remote management access on the target computer.

Go to **Computer Configuration > Policies > Administrative Templates > Windows Components > Windows Remote Management (WinRM) > WinRM Services**.

Then double click on **“Allow remote server management through WinRM”** to modify the setting as follows:

Set the policy to **“Enabled”**

Set the **IPv4 Filter** to \* or an all-encompassing subnet for your environment such as 10.0.0.0/16, 127.0.0.0/8

Leave the **IPv6 Filter** blank or set it to a wildcard \* as well.

Click **OK** to save.

### GROUP POLICY SETTING 3

Edit the settings — Opening Firewall ports

Next, we will create a new rule for the Firewall on the targeted client PC's.

Go to **Computer Configurations > Policies > Security Settings > Windows Firewall and Advanced Security > Windows Firewall and Advanced Security**

Then right click on **Inbound Rules > New Rule**

Create a new rule called Allow WinRM over HTTPS

We want to allow the inbound connection on port **5986**

Leave the Tick mark on **“Domain”** and if you plan on enabling WinRM over HTTPS with non-domain joined machines define the **“Private”** profile also

We then want to create a few more firewall rules using the default firewall rules

- Remote Event Log Management (RPC-EPMAP)
- Remote Event Monitor (RPC-EPMAP)
- Remote Event Log Management (RPC)
- Remote Event Monitor (RPC)
- Remote Service Management (RPC-EPMAP)
- Remote Service Management (RPC)
- Remote Scheduled Tasks Management (RPC-EPMAP)
- Remote Scheduled Tasks Management (RPC)

This should be enabled to **Allow inbound traffic** in the **“Domain”** category. If you plan on enabling WinRM over HTTPS with non-domain joined machines define the **“Private”** profile also

For good measure if you like you can deny traffic on port 5985. This is done by adding firewall default firewall rule **“Windows Remote Management (HTTP-In)”**, and Blocking traffic to that port. You will want to change the Firewall Profile to apply to All (Domain, Public, Private) for the deny rule.

### GROUP POLICY SETTING 4

Under **Administrative Templates > Network > Network Connections > Windows Defender Firewall > Domain Profile**

Set the policies for

- Windows Defender Firewall: Allow ICMP exceptions
- Windows Defender Firewall: Allow inbound remote administration exception

- Windows Defender Firewall: Allow inbound Remote Desktop

Set to **“Enabled”**

Define the filter you used in **Group Policy Setting 4** for these allowed values.

For example, you may have used a Wildcard \* or defined an all-encompassing subnet range such as 10.0.0.0/16

### GROUP POLICY SETTING 5

Under **Administrative Templates > System > Credentials Delegation > Allow delegating fresh credentials** and set the values to WSMAN/\*.yourdomain.com.

This will allow WinRM communication between any host ending in yourdomain.com.

Then set **“Allow delegating fresh credentials with NTLM-only server authentication”** under that same tree to that value WSMAN/\*.yourdomain.com.

For example, my email [rosborne@osbornepro.com](mailto:rosborne@osbornepro.com) is in the domain osbornepro.com.

I would set the value to WSMAN/\*.osbornepro.com.

We also want to Enable **“Encryption Oracle Remediation”** and set the drop-down value to **“Force Updated Clients”**.

This is to prevent [CVE-2018-0886](#) exploitation.

### GROUP POLICY SETTING 6

Under **Administrative Templates > Windows Components/Windows Remote Management (WinRM)/WinRM Client** set the below settings.

- **Allow Basic authentication:** Enabled
- **Allow CredSSP authentication:** Enabled This is used with Windows Admin Center (WAC)
- **Allow unencrypted traffic:** Disabled
- **Disallow Digest authentication:** Disabled : This is required for WinRM to be set up initially and have its settings modified
- **Disallow Kerberos authentication:** Disabled : **IMPORTANT:** If you are using [CIS-CAT Pro Dashboard \(CCPD\)](#) or are using Kerberos authentication with Windows Event Forwarding **ENABLE** this instead. This also may be useful if you use Kerberos as a backup to certificate authentication with Windows Event Forwarding)
- **Disallow Negotiate authentication:** Enabled This determines whether authentication is handled by Kerberos or NTLM. Kerberos is the preferred mechanism. Negotiate authentication on Windows-based systems is also called Windows Integrated Authentication.
- **Trusted Hosts:** \*.domain.com

### GROUP POLICY SETTING 7

Under **Administrative Templates > Windows Components/Windows Remote Management (WinRM)/WinRM Service** set the below settings

- **Allow Basic authentication:** Enabled
- **Allow CredSSP authentication:** Enabled
- **Allow remote server management through WinRM:** Enabled
- **IPv4 filter:** 10.0.0.0/16, 127.0.0.0/8
- **IPv6 filter:** \*
- **Allow unencrypted traffic:** Disabled
- **Disallow Kerberos authentication:** Disabled

- **Disallow Negotiate authentication:** Disabled
- **Disallow WinRM from storing RunAs credentials:** Enabled
- **Turn On Compatibility HTTP Listener:** Disabled
- **Turn On Compatibility HTTPS Listener:** Disabled

**GROUP POLICY SETTING 8** Create a Registry Setting that gets pushed out through Group Policy containing the below value

**SETTING:**

Action: Update

Path: HKLM:\Software\Policies\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\010103000F0000F0010000000F0000F0C967A3643C3AD745950DA7859209176EF5B87C875FA20DF21951640E807D7C24\  
Category

Name: State

Value: 1 REG\_DWORD Decimal

**GROUP POLICY SETTING 9** Create a PowerShell startup script to assign the distributed WinRM over HTTP certificates to port 5986

**\*\* CONTENTS OF POWERSHELL SCRIPT\*\***

```
New-Item -Path "$env:TEMP\Script\Logs" -ItemType Directory -Force -ErrorAction_
↳ SilentlyContinue | Out-Null
Try { Start-Transcript -Path "$env:TEMP\Script\Logs\PSTranscript_WinRM_Config.txt" -
↳ Append -ErrorAction SilentlyContinue } Catch { Write-Output -InputObject "[*] $(Get-
↳ Date -Format 'MM-dd-yyyy hh:mm:ss') Transcript already logging session" }
Write-Output -InputObject "[*] $(Get-Date -Format 'MM-dd-yyyy hh:mm:ss') Begin script_
↳ execution"
$ScriptResult = "Successfully"
$Icon = "*"
$Today = Get-Date
$RootCA = "$env:USERDOMAIN-CA01-CA"
$WinRMCertTemplateName = "WinRM over HTTPS"
$FQDN = ([System.Net.Dns]::GetHostByName(($env:COMPUTERNAME))).Hostname
$WinRMCertificate = Get-ChildItem -Path Cert:\LocalMachine\My | Where-Object -
↳ FilterScript { $_.Extensions.Format(0) -match "$TemplateName" }
$Thumbprint = $WinRMCertificate.Thumbprint
$CurrentListener = Get-ChildItem -Path WSMAN:\localhost\Listener | Where-Object -
↳ Property Keys -like "Transport=HTTPS"
If ($CurrentListener) {
    Write-Output -InputObject "[*] $(Get-Date -Format 'MM-dd-yyyy hh:mm:ss') Deleting_
↳ the current WinRM over HTTPS Listener"
    $CurrentListener | Remove-Item -Recurse -Force
    Write-Output -InputObject "[*] $(Get-Date -Format 'MM-dd-yyyy hh:mm:ss') Configuring_
↳ WinRM over HTTPS listener to use certificate $Thumbprint"
    New-WSManInstance -ResourceURI WinRM/Config/Listener -SelectorSet @{Address="*";_
↳ Transport="HTTPS"} -ValueSet @{Hostname=$Hostname; CertificateThumbprint=$Thumbprint}
} ElseIf (!$CurrentListener) {
    Write-Output -InputObject "[*] $(Get-Date -Format 'MM-dd-yyyy hh:mm:ss') Configuring_
↳ WinRM over HTTPS listener to use certificate $Thumbprint"
```

(continues on next page)

(continued from previous page)

```
New-WSManInstance -ResourceURI WinRM/Config/Listener -SelectorSet @{Address="*";  
↪Transport="HTTPS"} -ValueSet @{Hostname=$Hostname; CertificateThumbprint=$Thumbprint}  
} Else {  
    Write-Output -InputObject "[x] $(Get-Date -Format 'MM-dd-yyyy hh:mm:ss') FAILED to  
↪retrieve a certificate to assign to port 5986"  
    $ScriptResult = "in Failure"  
    $Icon = "x"  
} # End If ElseIf Else  
Write-Output -InputObject "[$Icon] $(Get-Date -Format 'MM-dd-yyyy hh:mm:ss') End script.  
↪execution ended $ScriptResult"  
Stop-Transcript -ErrorAction SilentlyContinue
```

Under **Computer Configuration > Windows Settings > Scripts (Startup/Shutdown)** double click **Startup** A new window will open. Select the **PowerShell Scripts** tab Click the **Add** button In **Script Name** enter the following value \\dc01.yourdomain.com\NETLOGON\ConfigWinRMoverHttps.ps1

**CONCLUSION** WinRM over HTTPS is now configured for your environment. Magnificent! When you now use PowerShell commands such as `Invoke-Command` or `New-PSSession` you will need to specify the `-UseSSL` parameter in order to use WinRM over HTTPS. Port 5985 will not accept connections in an ideal setup.

**ERROR MESSAGE** If you are on a domain controller you may receive the error message *The WS-Management service cannot process the request. The service is configured to not accept any remote shell requests.* There is a group policy object which needs to be amended to resolve this issue. The setting can be located at **Computer Configuration > Administrative Templates > Windows Components > Windows Remote Shell > Allow Remote Shell Access**. The run the command `gpupdate /force` and reboot the server.



## CONFIGURE WINDOWS EVENT FORWARDING

**REFERENCE:** <https://docs.microsoft.com/en-us/windows/win32/wec/setting-up-a-source-initiated-subscription>

Set the below Group Policy settings to configure Windows Event Forwarding with HTTPS using Certificate authentication

### GROUP POLICY SETTING 1

The Group Policy setting “**Computer Configuration > Policies > Administrative Templates > Windows Components > Event Forwarding > Configure Target Subscription Manager**” needs to be set to **WinRM over HTTPS (Port 5986)**: In my environment I added 2 entries for this to cover all basis. One has the CA certificate thumbprint with spaces after every 2 numbers, and the other entry is without spaces. The example values are below.

#### 1. Example Entry 1 Uses Certificate Authentication

Server=https://wef.domain.com:5986/wsman/SubscriptionManager/WEC,Refresh=900,IssuerCA=ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

#### 2. Example Entry 2 Uses Certificate Authentication

Server=https://wef.domain.com:5986/wsman/SubscriptionManager/WEC,Refresh=900, IssuerCA=ffffffffffffffffffffffffffffffffffff NOTE: The default Refresh rate value is 900 seconds or 15 minutes. This does not need to be defined. I included it to be thorough.

#### 3. Example Entry 3 Uses Kerberos Authentication (OPTIONAL IF YOU HAVE ARE HAVING TROUBLE WITH CERTIFICATES AND WANT TO TRY AND GET IT TOO WORK)

Using the below value without a certificate defined will allow/use Kerberos for authentication which is fine to use  
Server=https://wef.domain.com:5986/wsman/SubscriptionManager/WEC,Refresh=900

### GROUP POLICY SETTING 2

The Group Policy Setting “**Computer Configuration > Policies > Administrative Templates > Windows Components > Event Log Service > Security > Change Log Access**” needs to be set to the value of the property “**ChannelAccess**” after issuing the below command:

```
wevtutil gl security
```

The below command can be used to obtain the value you need to copy

```
`(wevtutil gl security | Select-String -Pattern "channelAccess").ToString().Trim().  
Replace("channelAccess:  ", "")`
```

Group Policy Setting “**Computer Configuration > Policies > Administrative Templates > Windows Components > Event Log Service > Security > Change Log Access (Legacy)**” needs to be set to the value of the property “**ChannelAccess**” after issuing the below command:

```
wevtutil gl security
```

The below command can be used to obtain the value you need to copy

```
`(wevtutil gl security | Select-String -Pattern "channelAccess").ToString().Trim().
Replace("channelAccess: ", "")`
```

### GROUP POLICY SETTING 3

The group policy setting **Computer Configuration > Administrative Templates > Windows Components > Windows Remote Shell > Allow Remote Shell Access** needs to be set to a value of “Enabled”.

**NOTE:** The CIS Benchmarks have recommendations to Disable this policy setting. There is debate over this given that powershell is now legitimately used by administrators to remotely access devices. I suggest using your Windows firewall and tailoring the other WinRM group policies we have defined above to closely match your environment if this is a concern for you.

## 6.1 Configure Source Initiated WEC Server

We now need to enable the Windows Event Collector service on the server we are forwarding events too. This can be done by opening up “Event Viewer” and clicking “Subscriptions” and then click “Yes” when the prompt appears.

I would suggest using the below commands instead as there is more involved that does not get set up automatically when using HTTPS

You can issue the below commands

```
Write-Output "[*] Giving NETWORK SERVICE permissions to the Security log for WEF"
$NetworkService = (wevtutil gl security | Select-String -Pattern "channelAccess").
ToString().Trim().Replace("channelAccess: ", "")
cmd /c "wevtutil sl Security /ca:$NetworkService"
Write-Output "[*] Add NETWORK SERVICE to event log readers group for WEF"
Add-LocalGroupMember -Group "Event Log Readers" -Member "NETWORK SERVICE"
Write-Output "[*] Starting WEC service and telling it to start up automatically"
Set-Service -Name Wecsvc -StartupType Automatic; Start-Service -Name Wecsvc
Write-Output "[*] Enabling the use of Certificates for authenticated connections"
cmd /c 'winrm set winrm/config/service/auth @{Certificate="true"}'
Enable-PSRemoting -Force
$Thumbprint = Read-Host -Prompt "Enter the WinRM Certificate Thumbprint assigned to
$env:COMPUTERNAME"
$Hostname = Resolve-DnsName -Name $env:COMPUTERNAME -Type A | Select-Object -First 1 -
ExpandProperty Name
New-WSManInstance -ResourceUri WinRM/Config/Listener -SelectorSet @{Address = ".*";
Transport = "HTTPS"} -ValueSet @{Hostname = $Hostname; CertificateThumbprint =
$Thumbprint }
netsh http delete urlacl url=http://+:5985/wsman/
cmd /c 'netsh http add urlacl url=http://+:5985/wsman/ sddl=D:(A;;GX;;;S-1-5-80-
569256582-2953403351-2909559716-1301513147-412116970)(A;;GX;;;S-1-5-80-4059739203-
877974739-1245631912-527174227-2996563517)'
netsh http delete urlacl url=https://+:5986/wsman/
cmd /c 'netsh http add urlacl url=https://+:5986/wsman/ sddl=D:(A;;GX;;;S-1-5-80-
569256582-2953403351-2909559716-1301513147-412116970)(A;;GX;;;S-1-5-80-4059739203-
877974739-1245631912-527174227-2996563517)'
cmd /c 'netsh http add urlacl url=https://+:443/wsman/ sddl=D:(A;;GX;;;S-1-5-80-
569256582-2953403351-2909559716-1301513147-412116970)(A;;GX;;;S-1-5-80-4059739203-
877974739-1245631912-527174227-2996563517)'
net user /add WEFAdmin gemKFueq4bn4nASHwUthf3Pycv2kZu8dKK6v
Add-LocalGroupMember -Group Administrators -Member WEFAdmin
```

(continues on next page)

(continued from previous page)

```

$CATHumbprint = Read-Host -Prompt "Enter the thumbprint of your Root CA"
cmd /c winrm create winrm/config/service/certmapping?Issuer=
→ $CATHumbprint+Subject=*+URI=* @{UserName="WEFAdmin";Password=
→ "gemKFueq4bn4nASHwUth3Pycv2kZu8dKK6v"} -remote:localhost
# EXAMPLE OF ABOVE COMMAND ENUMERATED
# winrm create winrm/config/service/certmapping?
→ Issuer=15c9df608e454371022622588616ca818e658bd4+Subject=*+URI=* @{UserName="WEFAdmin";
→ Password="gemKFueq4bn4nASHwUth3Pycv2kZu8dKK6v"} -remote:localhost
# OR IN POWERSHELL
# New-Item -Path WSMAN:\localhost\ClientCertificate -Subject "Hostname.domain.com" -URI
→ * -Issuer $RootCATHumbprint -Credential (Get-Credential -Username WEFAdmin -Message
→ "This needs to be the credentials of a local administrator") -Force

```

**NEXT** We can then configure the “Domain Computers” and “Domain Controllers” source collection using my prebuilt XML file with the below commands.

**IMPORTANT** Depending on how you followed my instructions you may need to delete the Query Id’s for Autoruns, MaliciousIPs, and Hash Validation for the below commands to work natively.

```

Write-Output "[*] Downloading Configuration files"
$XMLCompContents = Invoke-WebRequest -Uri "https://raw.githubusercontent.com/OsbornePro/
→ BTPS-SecPack/master/WEF%20Application/DomainComputers.xml" -OutFile $env:USERPROFILE\
→ Downloads\DomainComputers.xml
$XMLDCCContents = Invoke-WebRequest -Uri "https://raw.githubusercontent.com/OsbornePro/
→ BTPS-SecPack/master/WEF%20Application/DomainControllers.xml" -OutFile $env:USERPROFILE\
→ Downloads\DomainControllers.xml
Write-Output "[*] Creating source collection from downloaded config files"
wecutil cs $env:USERPROFILE\Downloads\DomainComputers.xml
wecutil cs $env:USERPROFILE\Downloads\DomainControllers.xml

```

## 6.2 Common Issues to Troubleshoot

If your source event collector is not receiving any events yet you will need to do the following things

1. Create a Start up script to put on your client devices that contains the below code

```

Write-Verbose "Giving NETWORK SERVICE permissions to the Security log for WEF"
cmd /c 'wevtutil sl Security /ca:0:BAG:SYD:(A;;0xf0007;;;SY)(A;;0x7;;;BA)(A;;0x1;;;BO)(A;
→ ;0x1;;;SO)(A;;0x1;;;S-1-5-32-573)(A;;0x1;;;S-1-5-20)'
Write-Verbose "Add NETWORK SERVICE to event log readers group for WEF"
Add-LocalGroupMember -Group "Event Log Readers" -Member "NETWORK SERVICE" -ErrorAction
→ SilentlyContinue | Out-Null
Write-Verbose "Ensuring WinRM service is available for WEF communication"
$EventInfo = Get-WinEvent -LogName 'Microsoft-Windows-Forwarding/Operational' -MaxEvents
→ 1
If ($EventInfo.LevelDisplayName -ne "Information") {
    cmd /c 'sc config WinRM type= own'
} # End If
Write-Output "[*] Enabling Certificate Authenticated WEC connections"
$Enabled = (winrm get winrm/config/service |select-string -pattern "Certificate = " |
→ Out-String).Trim()

```

(continues on next page)

(continued from previous page)

```
If ($Enabled -like "*false") {
    Set-Item -Path WSMan:\localhost\Service\Auth\Certificate -Value $True
}
Write-Output "[*] "
```

2. If you have an Intermediate Certificate Authority that assigns certificates you will need to set this registry value on your WEC Server

```
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\Schannel
↳" -Name "ClientAuthTrustMode" -Value 2 -Force
```

### Useful Commands for Troubleshooting

```
winrm get winrm/config # View applied GPO and other
↳ settings for WinRM
winrm get winrm/config/service # View applied settings on
↳ the WinRM service you have configured
winrm enum winrm/config/listener # Confirm the WinRM listener
↳ and certificate attached
winrm get http://schemas.microsoft.com/wbem/wsman/1/config # Confirm a computer
↳ certificate has been installed
winrm invoke Restore winrm/Config # Restore the Listener
↳ configuration
winrm set winrm/config/client @ allowunencrypted= false # Manually prevent
↳ unencrypted communication
winrm set winrm/config/service @ allowunencrypted= true # Manually prevent
↳ unencrypted communication
winrm set winrm/config/service @{CertificateThumbprint="$Thumbprint"}
winrm set winrm/config/client @ trustedhosts= *.domain.com # Manually define trusted
↳ hosts
#
# Test remote connection using PowerShell
Test-WSMan -ComputerName server.domain.com -UseSSL -Authentication ClientCertificate -
↳ CertificateThumbprint <Thumbprint>
#
# CERTMAPPING COMMANDS
winrm get winrm/config/service/certmapper?
↳ Issuer=15c9df608e454371022622588616ca818e658bd4+Subject=*&URI=*
winrm create winrm/config/service/certmapping?Issuer=<Thumbprint>+Subject=*&URI=* @
↳ {Username="WEFAdmin";Password="Password123!"} -remote:localhost
winrm delete winrm/config/service/certmapping?Issuer=<Thumbprint>+Subject=*&URI=*
```

The below command is how I manually test whether or not the WEF connection will work. This allows me to attempt a connection to the WEF server from a client device. The error message returned will be what you see in the event logs.

```
cmd /c 'winrm g winrm/config -r:https://EventCollector.domain.com:5986 -a:certificate -
↳ certificate:"<Thumbprint Local Client WinRM Certificate>"'
```

3. If you come across the below message in your event logs it means there was an issue in validating the certificate on port 443 (Not 5986).

The forwarder is having a problem communicating with subscription manager at address <https://server.doamin.com:5986/wsman/SubscriptionManager/WEC>.

Error code is 2150858882 and Error Message is <f:WSManFault xmlns:f="http://schemas.microsoft.com/wbem/wsman/1/wsmanfault" Code="2150858882" Machine="client.domain.com"><f:Message>The WS-Management service cannot find the certificate that was requested. </f:Message></f:WSManFault>.

**OR**

The forwarder is having a problem communicating with subscription manager at address <https://server.doamin.com:5986/wsman/SubscriptionManager/WEC>. Error code is 2150858882 and Error Message is .

You can solve this with the below PowerShell commands

```
Write-Output "[*] Incorrect certificate on 0.0.0.0:443. We need to replace that value.
↳with the thumbprint on 0.0.0.0:5986."
netsh http show sslcert
$Thumbprint = Read-Host -Prompt "Enter your WinRM Certificate Thumbprint at 0.0.0.0:5986"
$AppID = (netsh http show sslcert) | Select-String -Pattern "0.0.0.0:443" -Context 1,3 |
↳Select-Object -First 1 | Out-String | ForEach-Object { $_.Split(" ")[74] }
$AppID = $AppID.Trim()
cmd /c netsh http delete sslcert ipport=0.0.0.0:443
cmd /c netsh http add sslcert ipport=0.0.0.0:443 certhash=$Thumbprint appid=`"$AppID`"
Restart-Service -Name winrm,wecsvc
```

Another possible solution is going to be the WinRM service is not available. You can correct that by doing

```
$EventInfo = Get-WinEvent -LogName 'Microsoft-Windows-Forwarding/Operational' -MaxEvents
↳1
If ($EventInfo.LevelDisplayName -ne "Information") {
    cmd /c 'sc config WinRM type= own'
} # End If
```



## WINDOWS EVENT FORWARDING (WEF) APPLICATION

ASP.NET Core 3.1 web application that is used to investigate log files that have a high indication of compromise. These logs were collected from all devices in the environment. Below is a list of Events collected.

### Domain Computer Events:

- **1102:** Event Log Cleared
- **4732:** User added to an Administrators group
- **4733:** User removed from an Administrators group
- **4720:** Local User account created
- **4726:** Local User account deleted
- **7045:** New Service Installed

### Domain Controller Events:

- **1102:** Event Log Cleared
- **4732:** User added to an Administrators group
- **4733:** User removed from an Administrators group
- **4720:** Local User account created
- **4726:** Local User account deleted
- **7045:** New Service Installed
- **4756:** Domain User Added to a Universal Security Group
- **4757:** Domain User Removed from a Universal Security Group
- **4728:** Domain User Added to Global Security Group
- **4729:** Domain User Removed Global Security Group
- **4649:** Replay Attack Detected

### Summary

The purpose of this web application is to easily investigate any email alerts received that indicate compromise. This server has Windows Event Forwarding (WEF), configured in a “source collector” set up. This means that the clients in the environment initiate connections to the server before sending logs. When one of the suspicious event IDs occur on any of the devices, the event info is forwarded to the WEF source collector server using WinRM over HTTPS. Collected event logs are imported into a SQL database every hour. Any newly discovered events that indicate possible compromise trigger an email alert which is then sent to the IT Administrators. This application can be used to search that SQL database and view the details of an event. Searching a SQL database is 100x faster than parsing the Windows Event Logs through XML. This method stores critical events for a longer period of time than Event Viewer. When Event Viewer logs reach a certain size, old logs are removed to make room for new ones. Some admins may save backups of the overflow Event Viewer logs which requires more effort to search.

Protection against SQL injections and CSRF have been implemented. Penetration testing has been performed to ensure the security of the application. There are no presently discovered vulnerabilities. Security is a feature! If you find a vulnerability that I have not please let me know so I can fix it. [rosborne@osbornepro.com](mailto:rosborne@osbornepro.com)

### Functionality

- **REORDER:** The columns in the database table can be organized by ascending or descending order. This is done by clicking the linked column header.



- **SEARCH:** The search will parse the database by searching for the specified term in all of the columns. There are presently 10 items that will be displayed per page.
- **DETAILED:** Events can also be shown in a detailed view by clicking the “Details” button next to an event in the table. This will display a few more bits of info for the event.

### Table View of WEF Application

This WEF Application is used to view and list log files that are a high likelihood to be indications of compromise. These logs get collected from all devices in the environment. The following Event ID’s are collected and put into the database: 1, 2, 1102, 4732, 4733, 4720, 4726, 7045, 4756, 4757, 4728, 4729, 4649, and 4723

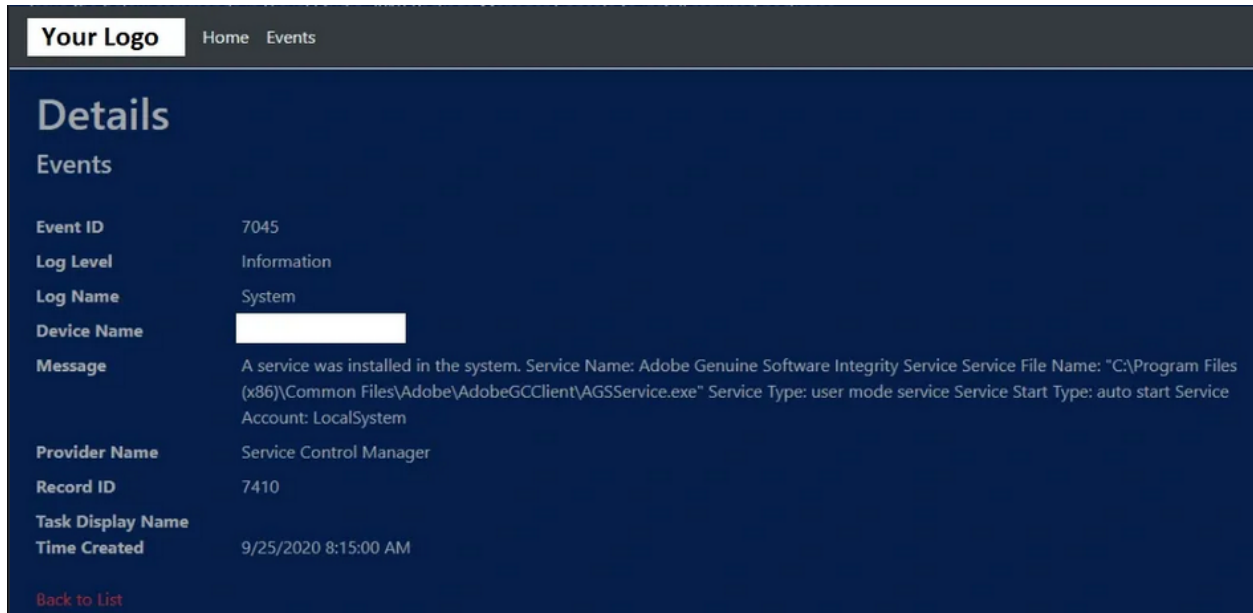
Below is an example of what the table view of an event looks like

<div> <div>Your Logo</div> <div>Home Events</div> </div>						
<div> <div>Index</div> <div> <div>Search</div> <div>Back to Full List</div> </div> </div>						
Event Id	LevelDisplayName	LogName	MachineName	Message	TimeCreated	Options
7045	Information	System		A service was installed in the system. Service Name: NEWTScanner Service Service File Name: %SystemRoot%\SysWOW64\NEWTScannerSvc.exe Service Type: user mode service Service Start Type: auto start Service Account: LocalSystem	9/24/2020 12:53:00 PM	Details
7045	Information	System		A service was installed in the system. Service Name: NEWTScanner Service Service File Name: %SystemRoot%\SysWOW64\NEWTScannerSvc.exe Service Type: user mode service Service Start Type: auto start Service Account: LocalSystem	9/24/2020 12:53:00 PM	Details
7045	Information	System		A service was installed in the system. Service Name: NEWTScanner Service Service File Name: %SystemRoot%\SysWOW64\NEWTScannerSvc.exe Service Type: user mode service Service Start Type: auto start Service Account: LocalSystem	9/24/2020 12:53:00 PM	Details
7045	Information	System		A service was installed in the system. Service Name: NEWTScanner Service Service File Name: %SystemRoot%\SysWOW64\NEWTScannerSvc.exe Service Type: user mode service Service Start Type: auto start Service Account: LocalSystem	9/24/2020 12:53:00 PM	Details
7045	Information	System		A service was installed in the system. Service Name: NEWTScanner Service Service File Name: %SystemRoot%\SysWOW64\NEWTScannerSvc.exe Service Type: user mode service Service Start Type: auto start Service Account: LocalSystem	9/24/2020 12:53:00 PM	Details

### Detail View in WEF Application

Events can also be shown in a detailed view by clicking the “Details” button next to an event in the table. This will display a few more bits of info for the event.





## 7.1 Perquisites and Setup Instructions

This [REPO](#) contains all the files needed for using Windows Event Forwarding to monitor an environment for intruders. This assumes that you have referenced the Windows Event Logging Cheat Sheet for logging in your environment. Use [LOG-MD](#) or [CIS-CAT](#) to ensure the recommended logging is configured. You will also need to configure WinRM in your environment. This can be done by following the instructions at the WinRM over HTTPS page on this site.

**CONFIGURE** (This should be configured in your environment)

1. WinRM over HTTPS

**DOWNLOAD & INSTALL** (I suggest installing these on the centralized WEF collection server to get started)

1. [SSMS \(SQL Server Management Studio\)](#)
2. [Microsoft SQL Server \(Express a.k.a Free Version Will Work\)](#)
3. [ASP.NET 3.1 Core Runtime](#)
4. [ASP.NET 3.1 SDK](#)
5. [Visual Studio](#)

**SIDE NOTE 1:** It is considered best practice to have the SQL server installed on a separate device than the one hosting the application. I did not abide by this here. As long as there are not any SQL Injections and least privilege permissions are applied to accounts that can access the SQL database, this should not be a concern.

**SIDE NOTE 2:** It is also best practice to have a development server that pushes out web application builds to the production server. I do not know what resources or extras may be available to a company. As such I am treating this to be as bare minimum as possible. These are subjects that can be learned in other areas of the web. I am just showing you how to set up this security package and need to draw the line somewhere. Feel free to use my instructions and then uninstall Visual Studio from the server when you are done with setup.

## 7.2 File List Overview

- [DomainComputers.xml](#) Windows Event Forwarding Config file for Domain Computers. can be applied on the source collector using the command: `wecutil cs DomainComputers.xml`
- [DomainControllers.xml](#) Windows Event Forwarding Config file for Domain Controllers can be applied on the source collector using the command: `wecutil cs DomainControllers.xml`
- [Import-EventsHourly.ps1](#) PowerShell script that imports collected WEF events into SQL database
- [Query to Create MSSQL DB Table](#) Creates the required database and table configuration for the MSSQL server database. Open the SSMS application, start a New Query, and execute the contents of this file inside the query. This will build the SQL database used by the WEF Application.
- [ImportTheScheduledTasks.ps1](#) This is an optional script that can be used to import XML files to the task scheduler, effectively creating the required scheduled tasks on the centralized WEF server
- [SQL-Query-Suspicious-Events.ps1](#) PowerShell script that discovers possible indicators of compromise and sends and an email alert.
- [TaskForSQLQueryEventsMonitor.xml](#) Task Scheduler import file that goes with [SQL-Query-Suspicious-Events.ps1](#). This executes on [SQL-Query-Suspicious-Events.ps1](#) on the hour. If any new events are viewed as possible indications of compromise an alert email will be sent out.
- [TaskImportFile.xml](#) Task Scheduler Import file that goes with [Import-EventsHourly.ps1](#). This task will execute the [Import-EventsHourly.ps1](#) script once an hour on the half hour, importing the most recent events into the SQL database.
- [WEFStartupScript.ps1](#) This should be the startup script on all devices sending events to the source WEF collector. This ensures that the WinRM service has the correct permissions, the service is running, and the service is available to send info to the source event collector.

## 7.3 Setup the WEF Application

### Intro

Now that WinRM over HTTPS is configured and the Group Policy Settings have been applied using instructions from the previous page we can begin setting up the configuration of the source collector. If you have not done this yet that is not a big deal as it should not matter what order these tasks are carried out.

### Step 1

In order to use the [DomainComputers.xml](#) and [DomainControllers.xml](#) config files in Windows Event Forwarding the below commands must be issued in an Administrator Command Prompt. Place the files [DomainComputers.xml](#) and [DomainControllers.xml](#) in the directory `C:\Users\Public\Documents`. Then open a Command Prompt or PowerShell window as an Administrator. This can be done with the key combo Windows Key + X, A

```
wecutil cs C:\Users\Public\Documents\DomainComputers.xml
wecutil cs C:\Users\Public\Documents\DomainControllers.xml
```

### Step 2

Create the SQL database schema and table.

1. Open **SSMS** (SQL Server Management Studio)
2. Sign in using an Administrator account to the default selected instance
3. Click **“Execute New Query”** in the top ribbon. This will open a text window

4. Copy and paste the contents of [Query to Create MSSQL DB Table](#) into the query and click “Execute”. This builds your SQL Database table where events will be imported.

### (Step 3 and Step 4)

The below two steps can be accomplished by executing the [ImportTheScheduledTasks.ps1](#). I have this automatically search for the XML files that need to be imported into Task Scheduler. Long as you do not rename the files in this repository this should go off without a hitch.

**USER YOU SELECT:** The above script will prompt you for a username to use in the Task Scheduler's execution of alerting and import files. The user you define will need to have been configured to have “**Log on as batch job**” permissions and “**Log on as service**” permissions. This is done through Group Policy at “**Computer Configuration > Windows Settings > Security Settings > User Rights Assignment**” Both of the mentioned settings will be in this GPO location.

**SQL USER PERMISSIONS:** Once this is done you will need to assign this user with “**Log on as batch job**” permissions to have read and write access to the SQL database. Do this by opening SSMS and signing into the default instance. In the “**Object Explorer Window**” on the left hand side you will need to expand the Databases tree, expand the Security Tree, expand the Users tree. The right click on User and select Add User if the name is not there. Add the user with “Log on as batch” permissions. Right click on the newly added user who is now existing in the expanded User tree. Assign the user **db\_datareader** and **db\_datawriter** permissions.

**SKIP:** You can skip to Step 5 if you have imported the tasks by executing [ImportTheScheduledTasks.ps1](#)

### Step 3

Create the Scheduled Task to Import Events into SQL Database

- Place the powershell script [Import-EventsHourly.ps1](#) into C:\Users\Public\Documents (*this is to match the Task Template in this repo*) or wherever you prefer to store this script. Be sure to sign it with a trusted Code Signing Certificate in your environment (*Import Code Signing Cert Info “Trusted Publishers” store in certmgr.msc*). This prevents it from running malicious code. Modify the permissions so only administrators can modify the script. Have this run every **hour on minute 55**. This leaves time for the events to get imported into the SQL database. Then on the hour, have the next task run.
- Create a scheduled task that runs once an hour on the hour. You can use my template [TaskImportFile.xml](#). Import this task and you should only need to define the user with Batch and Service permissions to run the script.

Below is the PowerShell Command to use code signing certificates to sign a script.

```
Set-AuthenticodeSignature C:\Users\Public\Documents\Import-EventsHourly.ps1 @(Get-ChildItem Cert:\CurrentUser\My -CodeSigningCert)[0]
```

### Step 4

Create Monitoring and Alert Task

Add [SQL-Query-Suspicious-Events.ps1](#) to C:\Users\Public\Documents which will match with the location of my XML template. Be sure to sign in with a trusted Code Signing Certificate (*Import Code Signing Cert Info “Trusted Publishers” store in certmgr.msc*) in your environment to prevent it from running malicious code. Modify the permissions so only administrators can modify the script. **Have this task run on the hour.**

Below is the PowerShell Command to use code signing certificate to sign a script

```
Set-AuthenticodeSignature C:\Users\Public\Documents\Import-EventsHourly.ps1 @(Get-ChildItem Cert:\CurrentUser\My -CodeSigningCert)[0]``
```

- Open Task Scheduler (taskschd.msc) and import the task from [TaskForSQLQueryEventsMonitor.xml](#) that runs once a day. This is done to execute [SQL-Query-Suspicious-Events.ps1](#). This is the script that alerts you when a suspicious event has been discovered.

- Edit the file [SQL-Query-Suspicious-Events.ps1](#) so the email variables are set to match your environment. Some of the SQL queries will also need to be modified in order to add accounts that commonly receive special permissions such as accounts that are used for LDAP binds or domain controllers system accounts (**Example: DC01\$**). Or don't use any special filtering. Whatever floats your boat. The SQL queries only return events from the last hour. This is significantly faster than filtering the Windows Event log through XML which also will eventually delete logs to make room for newer logs.

Once run, the script returns event information on the below possible indications of compromise from all those devices forwarding events.

- Were any Event Logs Cleared?
- Was a new Local or Domain User created anywhere (*Excluding WDAGUtilityAccount*)?
- Was a user added to a high privileged security group (*Administrators, Domain Admins, Schema Admins, Enterprise Admins, Print Operators, Server Operators, Backup Operators*)?
- Was a user removed from a high privileged security group (*Possible covering tracks*)?
- Were any new services run/created?
- Were any accounts locked out?
- Were any accounts unlocked?
- Were any special privileges assigned outside the norm (*Normal accounts: admin, dnsdynamic?*)
- Were any replay attack attempts detected?

### Step 5

To ensure the correct permissions are set on the Windows Event Log Source Collector issue the below commands (*on the Windows Event Forwarding Collection Server*). Open an Administrator PowerShell or Command Prompt session (**Windows Key + X, A**). Then execute the below commands:

```
netsh http delete urlacl url=http://+:5985/wsman/
netsh http add urlacl url=http://+:5985/wsman/ sddl=D:(A;;GX;;;S-1-5-80-569256582-
↪2953403351-2909559716-1301513147-412116970)(A;;GX;;;S-1-5-80-4059739203-877974739-
↪1245631912-527174227-2996563517)
netsh http delete urlacl url=https://+:5986/wsman/
netsh http add urlacl url=https://+:5986/wsman/ sddl=D:(A;;GX;;;S-1-5-80-569256582-
↪2953403351-2909559716-1301513147-412116970)(A;;GX;;;S-1-5-80-4059739203-877974739-
↪1245631912-527174227-2996563517)
```

### So What Now?

If you are having troubles with some Desktops and Servers that are not connecting to the source collector then push out the [WEFStartupScript.ps1](#) through Group Policy by making it a startup script. This startup script will run the WinRM service as it's own parent process which frees up it's usage for WEF. When the script gets triggered it performs a search on all collected targeted events for the last 1 hour and 5 minutes only. You can change this in the task and/or SQL Query script. The results will not always mean compromise but they will definitely help to discover them when they happen.

**NOTE:** Microsoft says the max limit of machines to source collect events from is 2,000 to 4,000.

**REFERENCE:** <https://support.microsoft.com/en-gb/help/4494356/best-practice-eventlog-forwarding-performance>

## 7.4 Reference Links

- <https://blog.netnerds.net/2013/03/importing-windows-forwarded-events-into-sql-server-using-powershell/>
- <https://docs.microsoft.com/en-us/archive/blogs/jepayne/monitoring-what-matters-windows-event-forwarding-for-everyone-even>
- <https://support.microsoft.com/en-us/help/4494462/events-not-forwarded-if-the-collector-runs-windows-server>
- <https://serverfault.com/questions/769282/windows-event-log-forwarding-permission>



## EXECUTE SCRIPTS WITH TASK SCHEDULER

External Link to a site that covers executing PowerShell scripts with Task Scheduler <https://www.windowcentral.com/how-create-automated-task-using-task-scheduler-windows-10>





## **SOLO SYSMON SETUP**

In case you wish only to set up Sysmon in your environment; I have put together a PDF walkthrough on how to use my configuration. The below link contains images and a step by step walkthrough for deploying Sysmon in your domain environment. <https://github.com/OsbornePro/Documents/raw/main/Sysmon%20Setup-0001.pdf>



## DISCLAIMER

**DISCLAIMER:** This suite nor any other security suite or tool can completely prevent or detect all security vulnerabilities. This tool adds monitoring to an environment and may not catch every possible scenario and is no guarantee of discovery.



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`