A Look Into Purple Fox's Server Infrastructure

trendmicro.com/en_us/research/21/l/a-look-into-purple-fox-server-infrastructure.html

December 13, 2021

Introduction

In one of our previous blog entries, we analyzed the Purple Fox botnet by providing an overview of how it worked. In addition, we also examined its initial access techniques and some of its associated backdoors.

In this research, we shed greater light on the later stages of its infection chain that we have observed via Trend Micro Managed XDR — specifically how it infects SQL databases by inserting a malicious SQL CLR assembly to achieve a persistent and stealthier execution. It should be noted that most files used in this attack are not stored on the disk and are either executed from memory after either being pulled from the command-and-control (C&C) server or encrypted, after which these are loaded by another process.

We also discuss the botnet's underlying C&C infrastructure and the motivation behind Purple Fox operators' choice to target SQL servers in their recent activities.

By examining Purple Fox's routines and activities, both with our initial research and the subject matter we cover in this blog post, we hope to help incident responders, security operation centers (SOCs), and security researchers find and weed out Purple Fox infections in their network.

Process injection

Let's begin by analyzing Purple Fox's process injection routine. The malware first loads its various components by spawning a suspended svchost.exe (changed to fontdrvhost.exe by the accompanied rootkit) process. It then loads the DLL component in the process address space, then redirects execution to the loaded DLL.

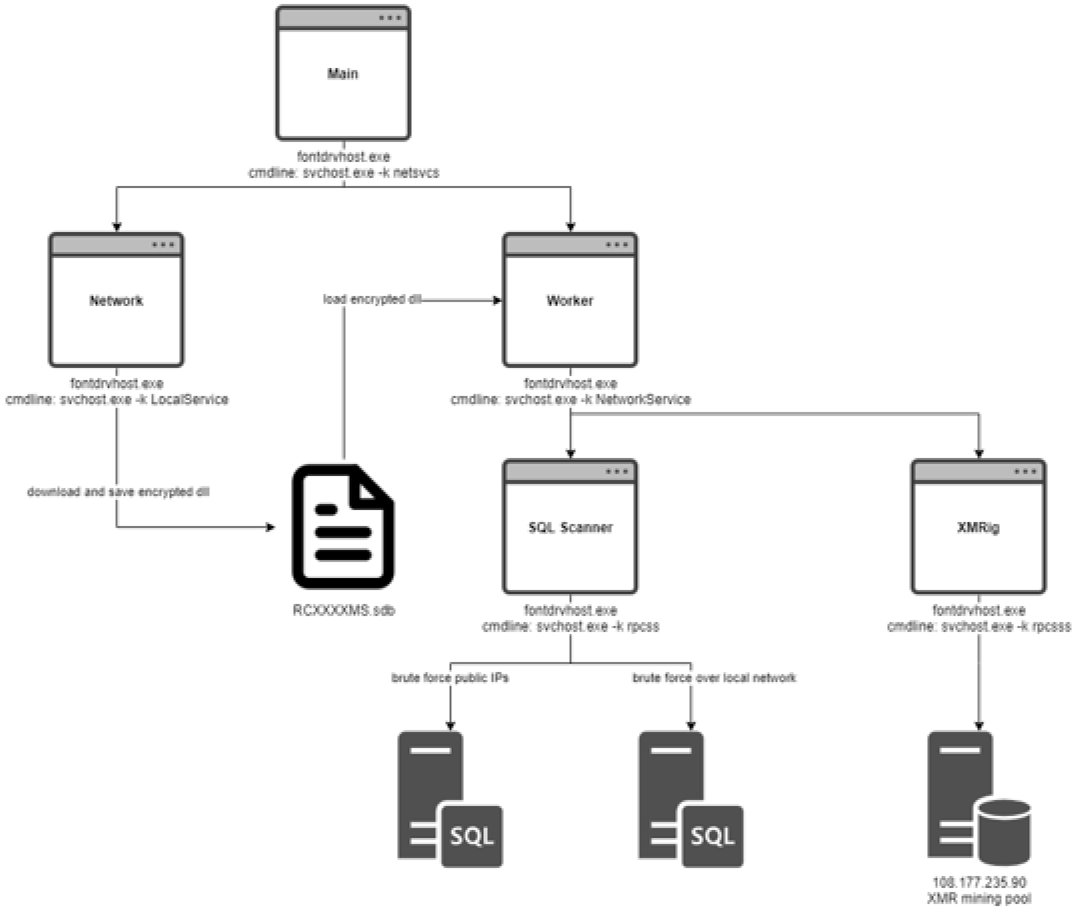

Figure 1. Process tree overview

The C&C server

The malware has three different ways to communicate with its C&C servers. Each method is used at a particular stage of execution for various purposes.

DNS

The DNS is used to get a list of C&C IP addresses at the start of each process execution. It is also used to renew this list if all servers fail to respond during this stage, or in a later stage as we see next. One thing to note is that the IP addresses received by the DNS requests are not the real IP addresses used for the C&C server. Although those received by the DNS requests are encoded, they can be decoded by subtracting a fixed number from the IP address. The following table shows examples of this.

IP address from DNS request	Decoded IP address
178[.]195.162.94	216[.]189.159.94:12113

79[.]222.214.20	117[.]216.211.20:10669
145[.]68.65.106	183[.]62.62.106:13600
73[.]127.195.228	111[.]121.192.228:14640
53[.]238.137.143	91[.]232.134.143:18372

Table 1. Examples of IP addresses received via DNS requests and the actual decoded IP addresses

The following are the domains used for the DNS requests:

- Kew[.]8df[.]us which points to m[.]tet[.]kozow[.]com
- ret[.]6bc[.]Us which points to a[.]keb[.]kozow[.]com

The list of returned IP addresses changes every few minutes or so, in order to cycle through the botnet C&C infrastructure.

UDP

The second communication method, User Datagram Protocol (UDP), is used for various types of messages and includes the building of a cache IP address list that will be used for further communication. In addition, it is used for pulling configuration for running tools and for retrieving the IP:PORT list for the HTTP traffic discussed in the next section.

After selecting an IP address from the DNS, it is decoded to the real IP address and a port number, after which a request is made to pull the cache IP address list. If at any point this cache list fails, the malware will return to the DNS to pull a new IP address to build another cache IP address list.

HTTP

To start performing its routine on the system, the malware pulls encrypted DLLs by issuing a GET request in the format http://IP:PORT/xxxx[.]moe, where IP:PORT is selected by a UDP message and xxxx[.]moe is one of the worker DLLs. These DLLs are saved in a file and are loaded by the worker process that decrypts, decompresses, and executes them.

The Worker DLLs

The SQL Server Scanner [32A7E157.moe]

The first of the worker DLLs is a SQL Server scanner that pulls its core module from /3FE8E22C.moe using the HTTP communication described previously. This core module is injected to a new process and the scanner configuration is pulled using UDP communication, which has the starting public address for scanning.

It scans local and public IP addresses for SQL Server over port 1433. If it finds an open port, it begins a brute-force attack for the SQL Server authentication using the 10 million-strong word list.

When the malware is authenticated, it executes an SQL script that installs a backdoor assembly (evilclr.dll) on the SQL Server database that is used to facilitate executing commands using SQL statements. Using this assembly, PowerShell commands are executed on the SQL Server to start Purple Fox's infection chain as discussed in our previous blog entry.

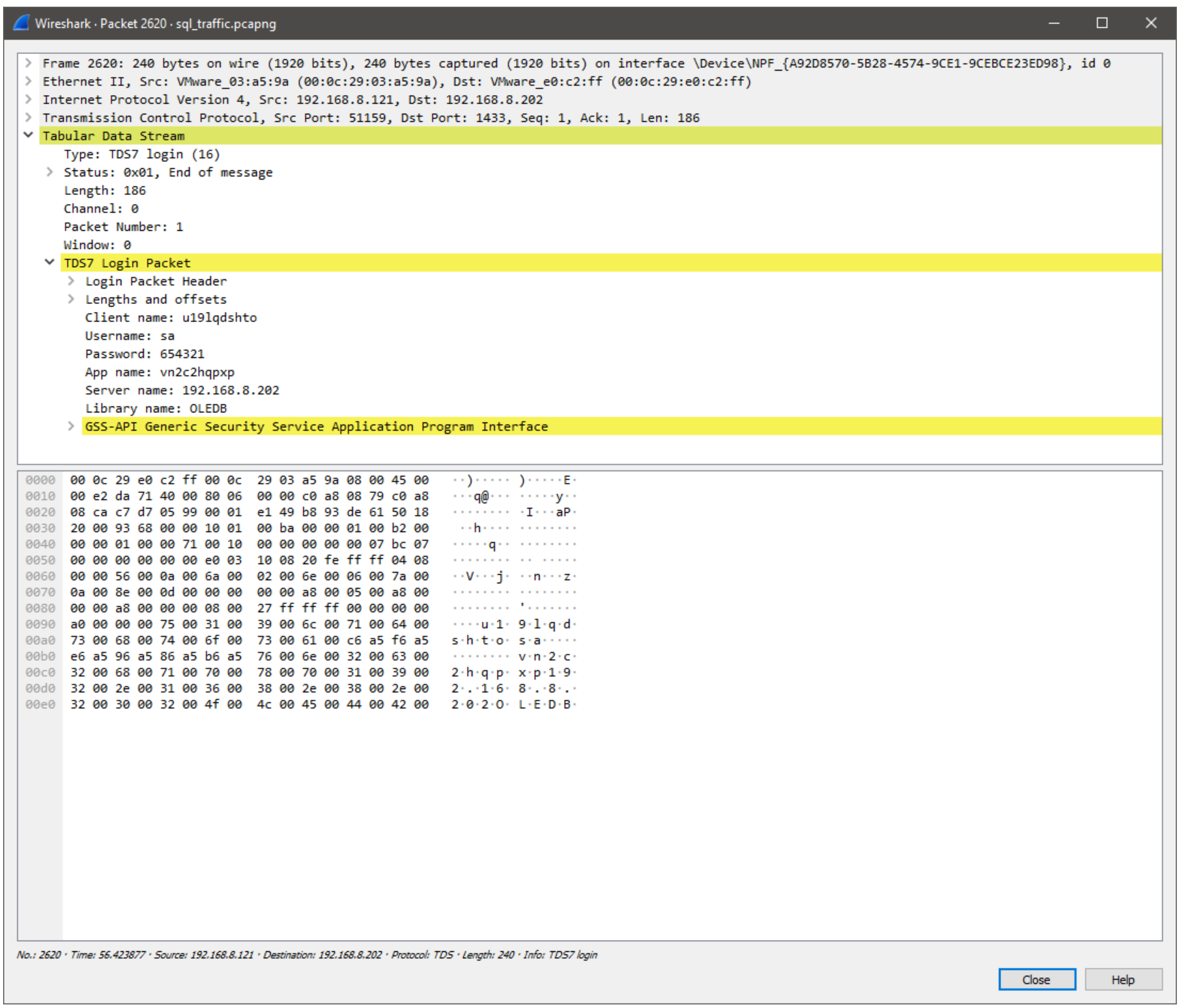

Figure 2. An SQL brute-force request

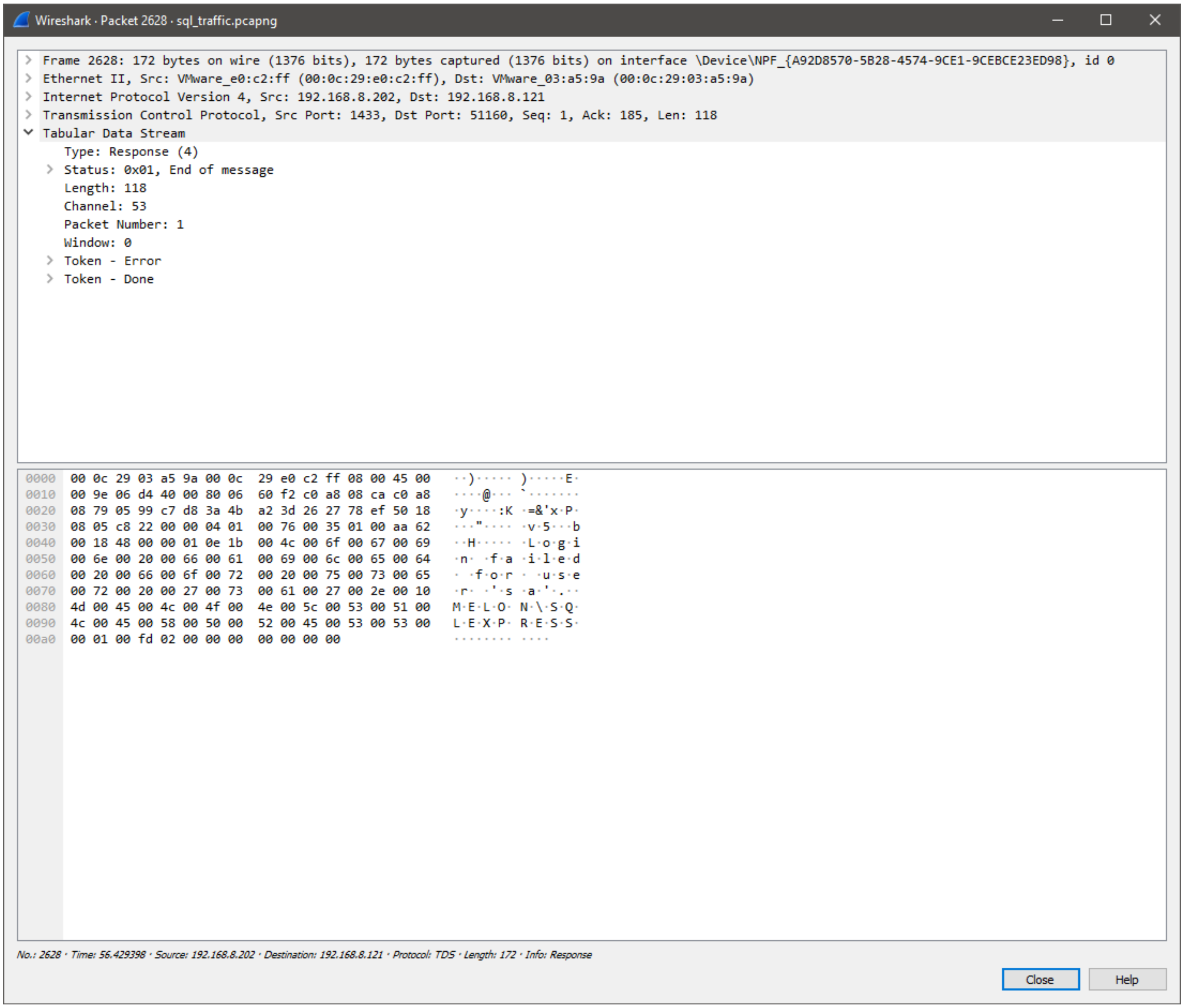

Figure 3. A failed response to the SQL brute-force request

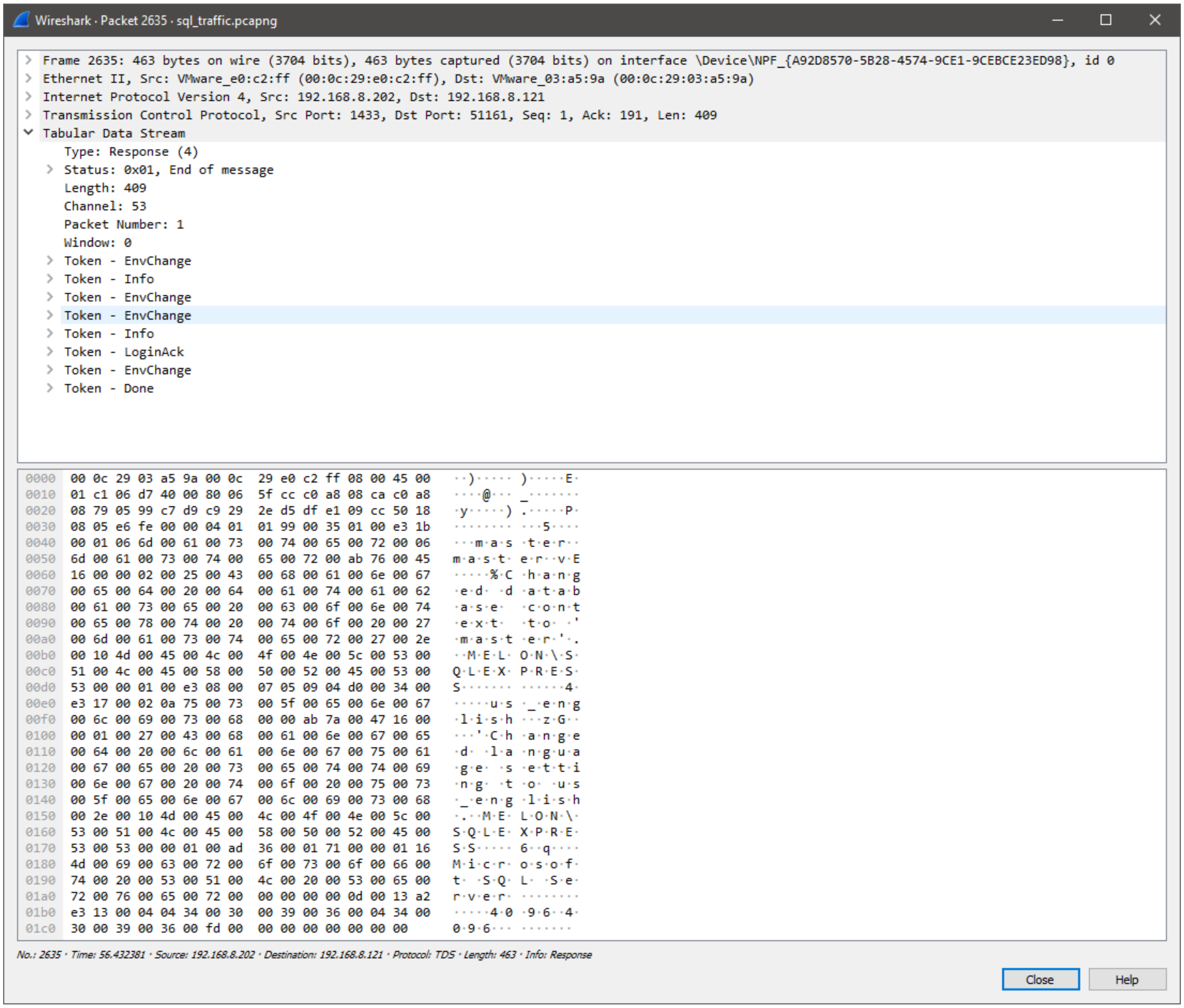

Figure 4. A successful response to the SQL brute-force request

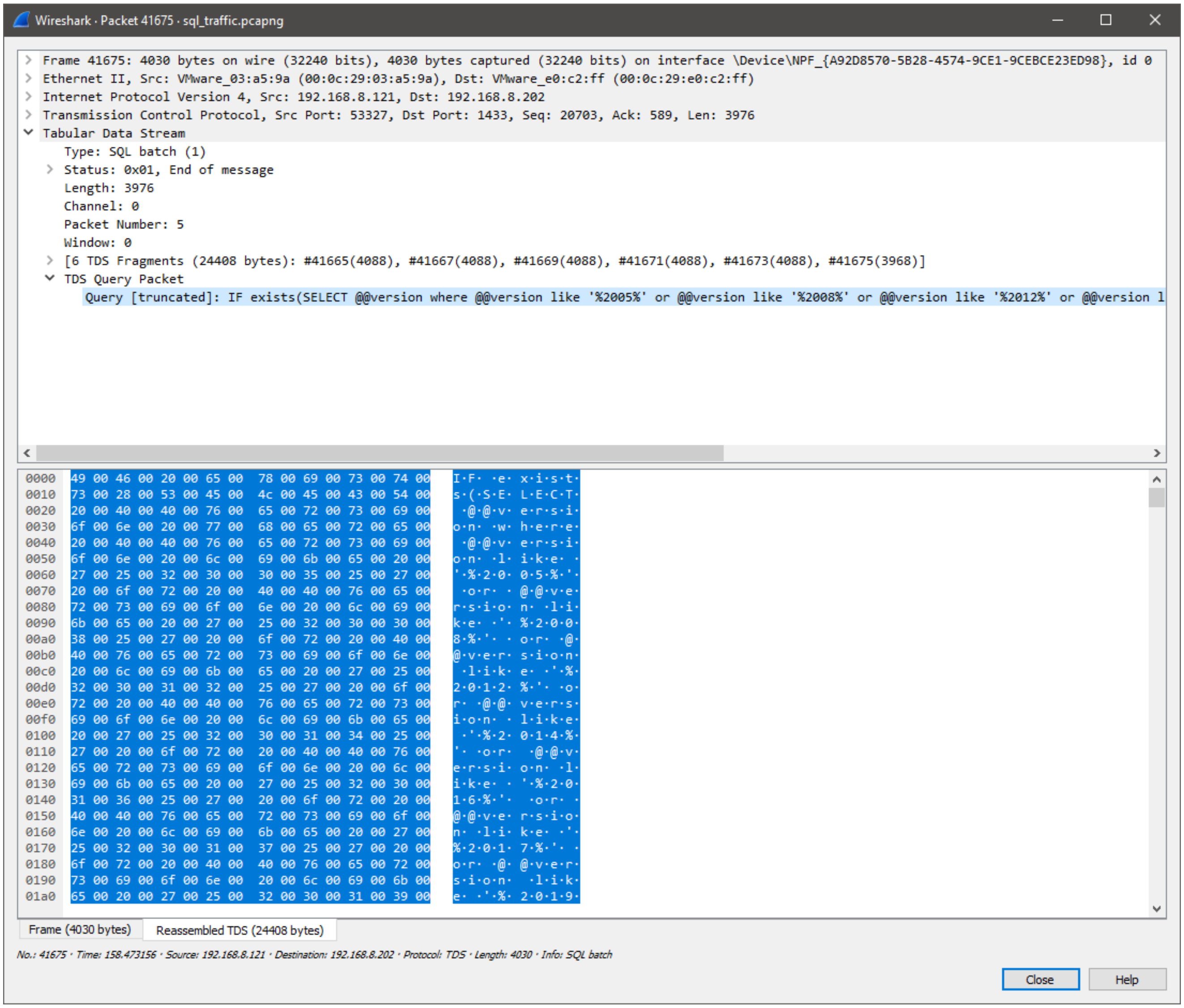

Figure 5. Executing SQL statements

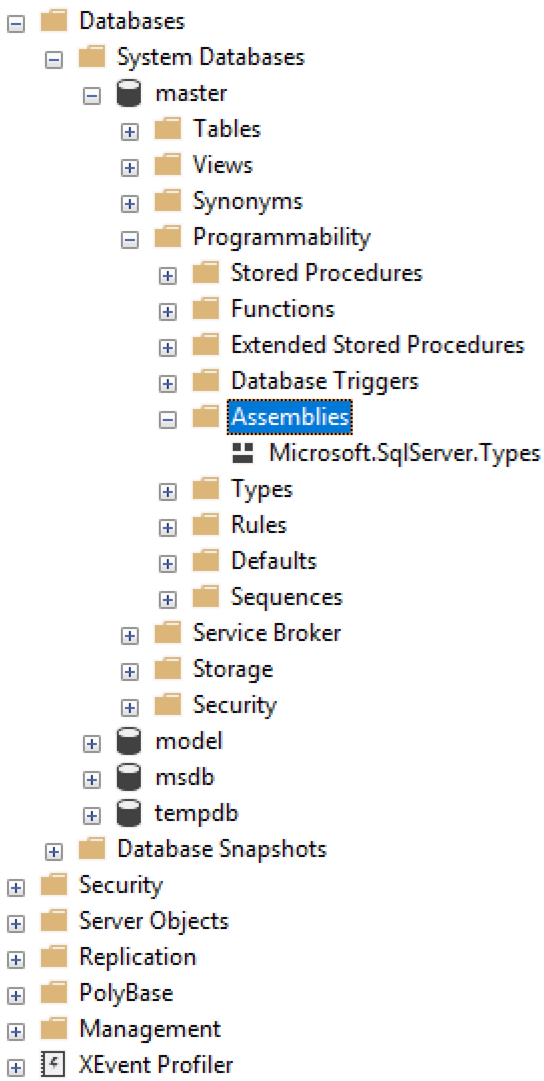

Figure 6. Database before infection

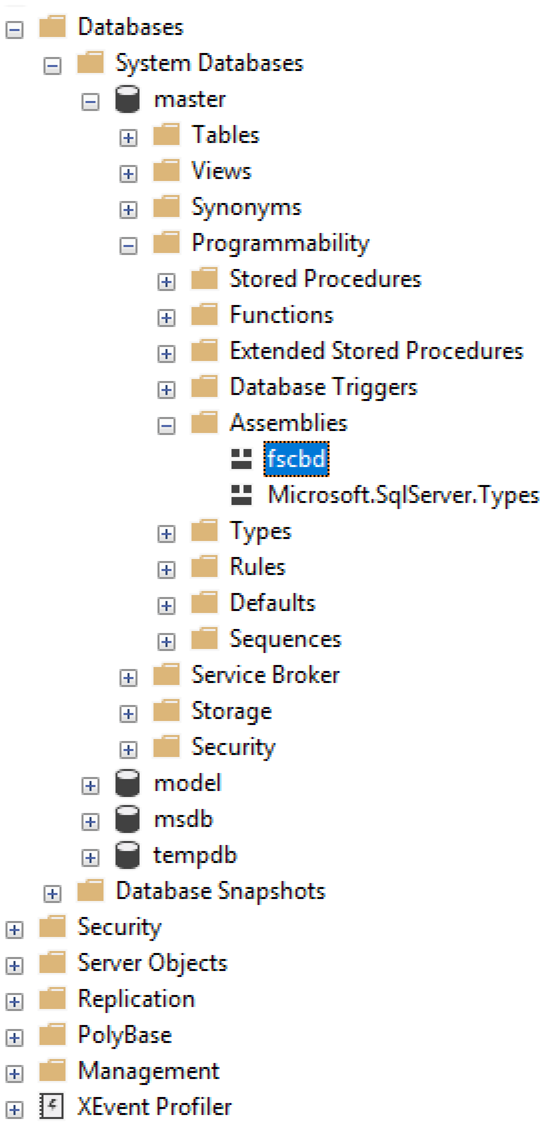

Figure 7. Database after infection

XMR Coinminer [F30DC9EB.moe]

The second worker DLL is an XMR Coinminer that starts its routine by retrieving the configuration over UDP. It then begins executing an embedded XMRig binary with the configuration pulled, making the bot join the mining pool on 108[.]177[.]235[.]90:443.

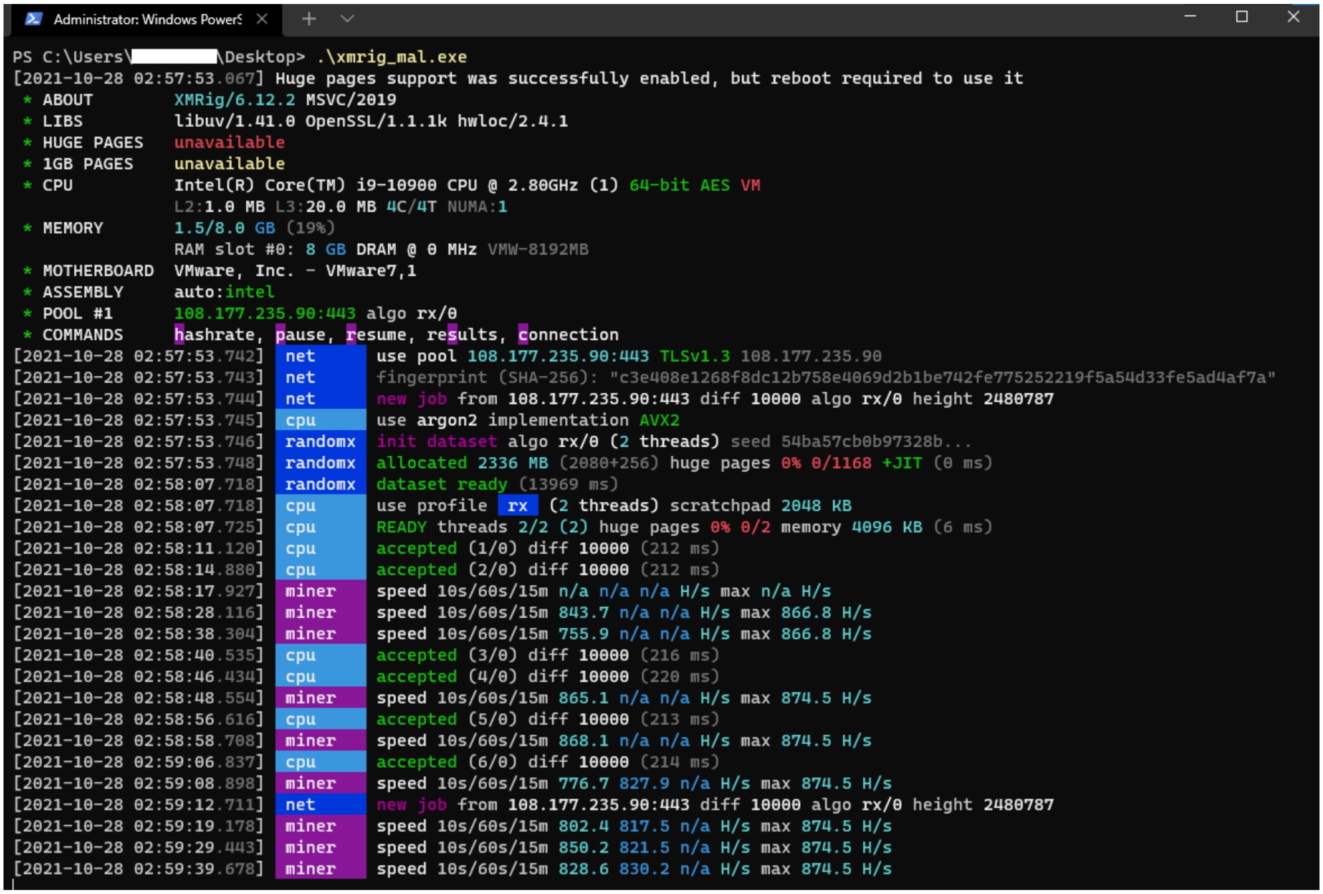

Figure 8. Custom XMRig running in the foreground

Operating system execution via SQL Server

Purple Fox focuses on SQL servers as its target as opposed to normal computers for the former's cryptocurrency-mining activities. This is mainly because of the more powerful hardware configuration — for both CPU and memory — that the servers would usually have. More specifically for SQL servers, the combination of CPU, memory, and disk factors should scale with the database-related operations to avoid bottlenecks in performance.

These machines normally possess much greater computing power compared to normal desktops, as such servers are usually fitted with hardware such as the Intel Xeon line of CPUs that produces a significantly higher amount of hash-based calculations (hash rates), making a server more advantageous to coinmining compared to a typical desktop computer.

Since SQL databases support different vectors for executing operating system commands directly, Purple Fox has leveraged the stealthiest method of having a binary inserted in the SQL server database that can be executed via TSQL commands. The following interfaces are available from the SQL components for the malicious actors to use when targeting an SQL server:

Method	Details

NET	• New Process() + UseShellExecute • System.management.automation.powershell • Common Language Runtime (CLR) Assemblies
C++	• ShellExecute/ShellExecuteEx • xp_cmdshell
COM objects	• wscript.shell • shell.application

Table 2. The available interfaces from the SQL components

Purple Fox opted to go with the .NET method using CLR Assemblies, a group of DLLs that can be imported into a SQL Server, in its infection chain instead of the more popular xp_cmdshell, which is heavily monitored by security analysts. Once the DLLs have been imported, they can be linked to stored procedures that can be executed via a TSQL script. The affected versions for this vector start from SQL Server 2008.

This method, which requires a system administrator role by default, executes as an SQL Server service account. By leveraging this interface, an attacker is able to compile a .NET assembly DLL and then have it imported into the SQL server. It is also able to have an assembly stored in the SQL Server Table, create a procedure that maps to the CLR method, and finally, run the procedure.

The CLR Assemblies method is reported to have been used before by groups other than Purple Fox, such as MrbMiner and Lemon Duck.

Infrastructure

The C&C servers used in the communication schemes that have been described here are infected servers that are part of the botnet used to host the various payloads for Purple Fox. We deduced this via the following facts:

- The C&C servers are SQL Servers themselves.
- The HTTP server header is mORMot, which is written in Delphi, the same language used for the various components.
- There is a large number of servers (1,000+ in just over a week).

Both initial DNS requests are CNAMEs to subdomains under kozow[.]com, which is a free dynamic domain service provided by dynu[.]com. This service can be updated with an API to make it point to different IP addresses — a technique the attacker uses to change the IP address at a regular interval.

Other notable characteristics

Using our telemetry, we found non-server systems infected with Purple Fox, indicating that there are other possible initial access methods other than the SQL Server brute-force attack to spread the malware.

This activity is similar to the ones seen in Lemon Duck attacks and even shares some techniques, like the use of PowerSploit for reflective PE loading and implementing the same backdoor, evilclr.dll, for the SQL Server assembly. Both attacks also share the same goal of mining Monero.

Security recommendations

Upon observing any suspicious activities related to the Purple Fox botnet on a SQL server, we recommend the following steps to completely remove all the malicious remnants from the infection.

- Review all the SQL Server's Stored Procedures and Assemblies for any suspicious assemblies not recognized by the DBAs. Remove any of these assemblies if detected.
- Execute the following TSQL script to remove the following remnants of malicious CLR assemblies that are inserted into the database:
- USE [master]
- GO
- DROP ASSEMBLY [fscbd]
- GO
- Disable all the unknown accounts on the database server and change all the passwords.
- As a defensive posture, do not publish externally exposed port TCP 1433 to an untrusted zone. In addition, secure the SQL server hosts via a perimeter firewall in a DMZ zone with well-protected access policies.
- Implement proper network microsegmentation and network zoning while also applying a zero trust policy via your network security controls.
- Restrict the traffic to and from SQL servers. These servers have a very specific function; therefore, they should only be allowed to communicate with other trusted hosts. Inbound and outbound internet accessibility should also be controlled.

Detections and Mitigations

Trend Micro Vision One™ with Managed XDR focuses on both the early stages of the attack kill chain (covered in the previous research) and the final payloads intended to do the actual damage, thereby protecting users of this service against the damage caused by the latest evolution of this botnet.

Both the Vision One platform and Managed XDR threat experts can correlate the suspicious activities observed from the protected SQL servers. An environment that has any of the behavioral detections found in our Vision One heuristics rules might mean that the SQL servers within the environment have already been affected by an attack. This extends even to stealthy malware, such as Purple Fox, that does not store majority of its files on the disk.

- Since servers have a predictable network footprint and behavior, unusual or unexpected network patterns could be a sign of botnet propagation.
- The same goes for unusual and unexpected SQL server application login failures that seem like brute-force attacks . The main propagation method for Purple Fox when infecting SQL servers uses brute-force attacks rather than acting as a worm that exploits only the vulnerable services.
- When a SQL server starts having unusual traffic related to UDP and TCP, there should be a massive surge in traffic since it scans public IP addresses and the local network. This will create a domino effect within an environment due to most organizations having more than one SQL server, such as standby or backup servers.
- Unusual network traffic patterns and login failures on the SQL server are also a good indicator for this threat.
- A sudden and unexpected spike in CPU utilization on the SQL server could also be a sign of SQL bottlenecks or an infection with the XMR Coinminer. Furthermore, there could also be unusual amounts of network traffic on the server as it joins the mining pool.

Malware

By examining Purple Fox's routines and activities, both with our initial research and the subject matter we cover in this blog post, we hope to help incident responders, security operation centers (SOCs), and security researchers find and weed out Purple Fox infections in their network.

By: Jay Yaneza, Abdelrhman Sharshar, Sherif Magdy December 13, 2021 Read time: (words)

Content added to Folio