RedLine Technical Analysis Report

web.archive.org/web/20230606224056/https://apophis133.medium.com/redline-technical-analysis-report-5034e16ad152

APOPHIS June 6, 2023

APOPHIS

--

RedLine is a stealer distributed as cracked games, applications, and services. The malware steals information from web browsers, cryptocurrency wallets, and applications such as FileZilla, Discord, Steam, Telegram, and VPN clients. The binary also gathers data about the infected machine, such as the running processes, antivirus products, installed programs, the Windows product name, the processor architecture, etc. RedLine also can be used as a malware loader or dropper for extended malicious impact. For instance, it can be used to infect the victim with additional malwares like ransomware.

The stealer implements the following actions that extend its functionality: Download, Download, And Execute PE files, Open Link, and Cmd. The extracted information is converted to the XML format and exfiltrated to the C2 server RedLine is considered as one of the most severe threats that are currently in the wild.

So I will be describing its utilities and how to prevent it.

> **RedLine entrypoint**

hardcoded C2
decrypting the c2 server using the key

cyberchef

> **c2 Communication**

RedLine will check if it can reach the C&C server using the functions `RequestConnection()` and `TryGetConnection()`.

Then, The stealer communicates with the c2 server using “SOAP” messages. The following SOAP requests will determine how the malware will act and the malware uses these requests to avoid infection, here are some HTTP requests I have captured using Wireshark network communications with the C2 server, We can notice some IP addresses corresponding to VPNs or online sandboxes that the malware wants to avoid

The malicious process could enable/disable some functionalities based on the SOAP response. For example, by specifying a false value in the ScanWallets field, the binary doesn’t scan the system for crypto wallets, For not taking a lot of time it will skip what isn't here on the victim machine

scanning args
And will save the result in a structure called ScanResult:

- An ID that corresponds to the infected machine

- The Release ID that is hard-coded in the binary

- The machine name which is in fact the username associated with the process

- The OS version

- The culture of the current input language

Like that :

scanning result function
It’s important to know the inner structures of the `ScannedFile` class which is used to populate the exfiltrated files. As you can see in the screenshot, once a file is being instantiated, nearly all of its important contents are stolen.

scanned file
one of the interesting functions is the function which harvests data about the machine like storing data such as the antiviruses, a list of installed input languages, a list of installed programs, a list of running processes, and information about the processors and the graphics device in a class called ScanDetails, as highlighted below:

scandetails function
The malware uses the WMI to enumerate any installed security solution. It allows it to get any Antivirus, anti-spyware, and Firewall (third-party) software using the `rootSecurityCenter` strings.

rootsecuritycenter in strings
OpenSubKey opens the “SOFTWARE\Clients\StartMenuInternet” registry key. The name of a browser is obtained by the function call to (GetValue) and then the path from the “shell\open\command” registry key:

shell\open\command

opening the “SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall” registry key, which contains the installed programs. to get the program name and version:

our infostealer extracts the serial number of the physical disk drives:

The list of running processes is retrieved by running the “SELECT * FROM Win32_Process” query. The malware creates a list that contains the session ID of the current process, the process ID and the name of a process extracted from the query, and the command line:

select from Win32_process
This function is used to make a list of running processes’ names and the path to the executable files:

names
if-statements to decide if the infostealer will harm a machine from certain countries or not (Russians don’t play with their citizens) :

blocking countries

RedLine is famous for its intelligence from stealing browsers, The stealer targets Chromium-based browsers like :(Chrome and Opera) and Geckobased browsers like :(Mozilla Firefox)

browser scanning
it also checks the brwoser name, version, and where it has been installed too

With their potent combination of advanced hacking techniques and stealthy operations, redline stealing crypto-wallets pose a grave threat to the security and stability of the digital currency ecosystem, RedLine targets many wallets like Armory, Exodus, Ethereum, Monero, Atomic, BinanceChain, Jaxx, Electrum, Guarda

targeted wallets
you can notice (The AllWallets class)which is used for generic crypto wallets.

RedLine look uses the `GetProcessesByName()` function to get the `ExecutablePath` for Telegram running process. Then, it looks for the folder `tdata`. This is where Instant Messenger stores its session data, including images and conversations

telegram

RedLine extracts the Discord tokens using the `Discord.GetTokens()` function and get chat logs from the “.log” and “.ldb” files like that :

Discord

RedLine targets the VPN clients of the following services: NordVPN, OpenVPN, and ProtonVPN.

proton VPN
RedLine stealer searches the filesystem for the “%USERPROFILE%\AppData\Local\NordVPN” directory, which corresponds to the NordVPN,

The credentials stored in the “user. config” file are extracted by the malware :

Nord VPN
Then, the credentials are decoded from Base64 and then stored in Account. Username and Account.Password:

A File Transfer Protocol (FTP) client is a software application that enables users to connect to an FTP server and transfer files between their local computer and the remote server. It provides a graphical or command-line interface that allows users to browse, upload, download, and manage files and directories on the FTP server. FTP clients authenticate users with login credentials (username and password) to establish a secure connection and facilitate file transfers using the FTP protocol.RedLine uses the `ScanCredentials()` function to extract the required credentials and put them in `Account` class which will contain the URL + username + password.

filezilla

RedLine doesn't stop at being just a stealer, it also plays as a loader. RedLine takes the role of a malware loader.

The “DownloadAndEx” function downloads an executable and executes it by using the Process.Start function:

downloadandexecute
some dynamic analysis showed that executable being downloaded and executed

child of RedLine
Log files that have been created in hidden paths by RedLine

created log files

```
c2 = 188.124.36.242:258028.253.95.121 [fg.download.windowsupdate.com.c.footprint.net]
[wu-bg-shim.trafficmanager.net] [ctldl.windowsupdate.com]192.229.221.95
[fp2e7a.wpc.phicdn.net] [fp2e7a.wpc.2be4.phicdn.net] [ocsp.digicert.com]52.40.92.150
[m.stripe.com]MD5:74200bd872e0b3d75b1d85332c3be083childMD5:9F325813208D96F9CD73D915553
```

```
COLLECTION     : Data from Information Repositories T1213
|DEFENSE EVASION: Obfuscated Files or Information T1027
|DISCOVERY      : File and Directory Discovery T1083
|                 Query Registry T1012
|                 System Information Discovery T1082
```

```
rule redline : infostealer
{
 meta:
  description = "redline yara"
  author = "apophis"

   strings:  $s1 = "IRemoteEndpoint"  $s2 = "ITaskProcessor"  $s3 = "ScannedFile"
$s4 = "ScanningArgs"  $s5 = "ScanResult"  $s6 = "DownloadAndExecuteUpdate"  $s7 =
"CommandLineUpdate"  $s8 = "TryCompleteTask"  $s9 = "TryGetTasks"  $s10 =
"TryInitBrowsers"  $s11 = "InstalledBrowsers"  $s12 =
"asdk9345asd,asdk8jasd,ылв92р34выа,аловй,ResFac.ыал8р45,ываш9р34,ывал8н34,вал93тфыв,ва
  $s13 = "Chr_0_M_e"  $s14 = "EL3_K_Tr00M"  $s15 = "BCryptSetProperty"  $s16 =
"E_x0_d_u_S"   $s17 = "Guarda"   $s18 = "g_E_c_к_0"  $s19 = "C_o1_n0_ми"    $s20 =
"%USERPstring.ReplaceROFILE%\\Apstring.ReplacepData\\Locastring.Replacel"  $s21 =
"ScannedCookie"  condition:  any of them}
```