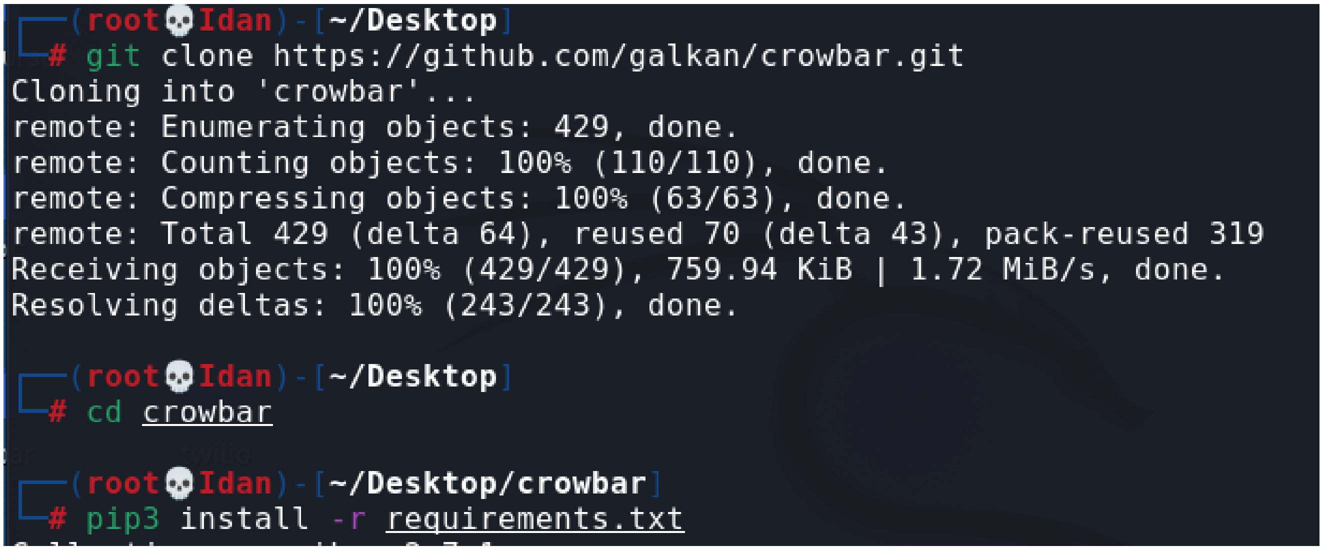
Unmasking RedLine Stealer

medium.com/@idan_malihi/redline-stealer-malware-analysis-76506ef723ab

Idan Malihi November 4, 2023



```
┌──(root💀Idan)-[~/Desktop]
└─# git clone https://github.com/galkan/crowbar.git
Cloning into 'crowbar'...
remote: Enumerating objects: 429, done.
remote: Counting objects: 100% (110/110), done.
remote: Compressing objects: 100% (63/63), done.
remote: Total 429 (delta 64), reused 70 (delta 43), pack-reused 319
Receiving objects: 100% (429/429), 759.94 KiB | 1.72 MiB/s, done.
Resolving deltas: 100% (243/243), done.

┌──(root💀Idan)-[~/Desktop]
└─# cd crowbar

┌──(root💀Idan)-[~/Desktop/crowbar]
└─# pip3 install -r requirements.txt
```

Idan Malihi

--

Executive Summary

The ‘RedLine’ malware was discovered in 2020 during the COVID-19 outbreak. This information-stealing variant allows attackers to steal personal and sensitive data such as login credentials, web browsing history, crypto wallets, geographical locations, etc.

After extensive research on the ‘RedLine’ malware, I discovered many threat actors were using it to sell stolen information on the Dark Web and Telegram. I decided to delve deeper into the topic by analyzing a sample of the ‘RedLine’ malware and conducting a high-level malware analysis.

RedLine Data Logs For Sale in Telegram

Technical Details:

Filename: NetFlix Checker by xRisky v22.exe
File Type: Executable
Architecture: PE32 (32-bit)

Size: 6.47MB
SHA256: e3544f1a9707ec1ce083afe0ae64f2ede38a7d53fc6f98aab917ca049bc63e69
MD5: 8556792f20126e1ed89f93e1e26030e5

Infection Diagram

RedLine Stealer’s Infection Diagram

Static Analysis

Malware’s Architecture

The malware is an executable file that works with 32-bit architecture.

Malware’s Architecture

Scanning the Malware in the VirusTotal

55 out of 71 anti-virus engines identified the binary as malicious.

Malware Scan in VirusTotal

Strings

Several strings indicate the malware resources and modules usage that are coded in .NET, such as the mscorlib, System, Object, and System.Reflection, etc.
Also, the malware uses functions that can indicate the malware’s functionality, such as:

Aes - the malware uses the AES encryption, a symmetric block cipher.

MemoryStream - Creates a stream whose backing store is memory.

GetBytes - Function used to encode strings into bytes.

SymmetricAlgorithm - Represents the abstract base class from which all implementations of symmetric algorithms must inherit.

ICryptoTransform - Basic operations of cryptographic transformations.

CreateDecryptor - Creates a symmetric decryptor object.

CryptoStream - Represents the abstract base class from which all implementations of symmetric algorithms must inherit.

CryptoStreamMode - This function specifies the mode of a cryptographic stream.

FromBase64String - Convert base64 encoded strings.

Malware Resources and Modules

Initial Execution

The malware uses an AES encryption, which, after execution, decrypts the encryption and injects it to an executable file named ‘winlogon.exe’ and drops it to the %AppData% directory path.

The AES encryption data is the actual RedLine malware.

AES Encryption
Code Injection into the Winlogon.exe File
The malware employs loops in the code that lead back to the same code. It drops an executable file titled ‘NetFlix Checker by xRisky v2.exe’ in the Desktop directory. Additionally, it drops two executable files named ‘chrome.exe’ and ‘svchost.exe’ in the %AppData% directory.

‘NetFlix Checker by xRisky v2.exe’ File Drop
‘svchost.exe’ and ‘chrome.exe’ Files Drop
In summary, for the initial execution, the malware drops four executable files into the endpoint.

Three Files in %AppData%
A File in the Desktop
The malware adds chrome.exe to the endpoint’s system-scheduled tasks for persistent data collection.

Task Scheduler

Winlogon.exe Code Analysis

In the initial execution of the malware, three executable files (winlogon.exe, svchost.exe, and chrome.exe) are dropped in the %AppData% path.
However, the actual RedLine malware is in the ‘winlogon.exe’ file.

During the execution, the ‘winlogon.exe’ file attempts to send stolen data to the C2 server.

Initially, the malware author uses an obfuscator to make the executable’s source code unintelligible, which complicates code review.

Code Obfuscation Detection

In the source code of the stealer, the malware's actions are exposed, which will be performed during execution, such as stealing the following data: Chrome cookies, Opera cookies, crypto wallets, system hardware, geo-location, files, etc.

Malware's Actions

Browsers

Users commonly save login credentials, auto-fill, and credit card info in browsers like Chrome, Opera, and Firefox for faster form-filling and account access.
Stealers like 'RedLine' specifically target browser information in order to steal victims' accounts and access them, especially for credit card information, to steal money and use it to make unauthorized purchases online.

The malware steals the browser's version information, account credentials, auto-fill data, cookies, credit cards, login data, and geolocation from Chrome and Opera browsers.

Account Credentials Theft
The malware steals the autofill data from the browsers.

Autofill Data Theft
Autofill Data Theft 2
The malware gathers information about the browser installed on the endpoint.

Browser Information Theft
The malware steals credit card information from the browsers.

Credit Card Information Theft
Credit Card Information Theft 2
The malware steals login data from the Chrome browser.

Login Data Theft
The malware steals cookies from the browsers.

Cookies Theft
The malware uses 'GeoPlugin,' a geolocation web service API, to determine the location of an endpoint based on its IP address.

GeoLocation API
The malware uses the OpenSubKey function to access the registry path SOFTWARE\Clients\StartMenuInternet and retrieve the string value using the GetValue method.

Registry Path Access and Read

Crypto Wallets

Most threat actors use cryptocurrency wallets for anonymity. These wallets generate unique wallet addresses for victims to transfer money anonymously.
Furthermore, some people invest in cryptocurrency coins like Bitcoin, Ethereum, Tether, Solana, etc.
Threat actors target crypto wallets to steal victims' crypto wallet information and money.

The malware searches and steals information and files from the list of wallets, such as Coinbase, Yoroi, Atomic, Wombat, Jaxx Liberty wallets, Saturn, etc.

Cryptocurrency Wallets List
Cryptocurrency Wallets List 2
Cryptocurrency Wallets List 3
Cryptocurrency Wallets List 4
Cryptocurrency Wallets List 5
Cryptocurrency Wallets List 6
The malware searches for Armory .wallet files in the %AppData% directory.

Armory Wallet Files Theft
The malware attempts to steal cryptocurrency wallet files in the 'atomic' directory.

Atomic Wallet Files Theft
The malware searches for JSON files or any files within the '\Exodus\' directory and locates the 'exodus.wallet' file on the endpoint.

Exodus Wallet Files Theft
The stealer attempts to steal information from the Jaxx Liberty cryptocurrency wallet directory.

Jaxx Liberty Wallet Files Theft
The stealer attempts to search for any Coinomi crypto wallet files within the \Coinomi directory located in the %AppData% path.

Coinomi Wallet Files Theft
The stealer attempts to steal all files related to the Electrum wallet located in the %AppData%\Electrum\wallets directory.

Electrum Wallet Files Theft
The stealer tries to collect information about the Guarda wallet in the %AppData% directory.

Guarda Wallet Files Theft

System

Malware authors program stealers to gather system information, such as country, city, hardware, and IP addresses. This is done to profile victims, enable geographic targeting, assess hardware vulnerabilities, deliver content in victims' languages, etc.

The malware gathers information from the endpoint, including city, country, file location, hardware, IP address, language, machine name, and zip code.

System Information Gathering
System Information Gathering
The malware author uses the WQL command 'SELECT * FROM Win32_Processor' to steal information about the endpoint, including the number of cores in the processor and running processes.

SELECT * FROM Win32_Processor
In addition, the malware author uses the WQL command 'SELECT * FROM Win32_VideoController' to steal information about the RAM in the endpoint.

SELECT * FROM Win32_VideoController
Also, the malware uses the WQL command 'SELECT * FROM Win32_DiskDrive' to retrieve the disk drives connected to the endpoint and their serial number.

SELECT * FROM Win32_DiskDrive
The malware uses the WQL command 'SELECT * FROM Win32_Process Where SessionId=' to retrieve session IDs, names, and command lines.

SELECT * FROM Win32_Process Where SessionId=
The malware collects 'ProductsName' and 'CSDVersion' values from the 'SOFTWARE\Microsoft\Windows NT\CurrentVersion' registry path and system architecture information.

Collects 'ProductsName' and 'CSDVersion' Values
Once the stealer attempts to steal data from the endpoint, it stores the acquired information in a list that includes languages, browsers, FTP connections, chat logs for games, game launcher files, installed browsers, message client files, Nord accounts, open processes, Proton, scanned files, scanned wallets, security utilities, software, and hardware components of the system.

Saves Information in Lists
The malware gathers IPv4, city, country, and zip code data from the endpoint.

IP, City, Country, and ZipCode Theft
IP and Location Theft
Country and PostalCode Theft
CurrentInputLanguage Theft

TimeZoneInfo.Local Theft

Scanned Files

Malware authors program stealers to steal the known and sensitive files on the victims' system, such as docx, txt, doc, and csv.
Threat actors would want to steal sensitive information for further exploitation, financial gain, identity theft, and data extortion.
The stolen files can also be used for espionage, intelligence gathering, or resale on the dark web.

The malware attempts to steal exe, docx, txt, doc, csv, and DLL files from the endpoint.

exe, docx, txt Files Theft
doc, csv, docx, doc, DLL Files Theft
The malware attempts to extract account details from the \\FileZilla\\sitemanager.xml file located in the %AppData% directory.

FileZilla sitemanager.xml Theft
The malware searches for files and directories in Program Files (x86) and ProgramData paths.

Searches Files and Directories in Program Files (x86) and ProgramData
The malware uses the GetDirectories method to retrieve the names of subdirectories and the GetFiles method to retrieve the names of files within those directories.

Gets Directories and Sub-Directories
The method JavaScriptSerializer can be used to convert JSON strings into objects.

JavaScriptSerializer
The malware uses the LoadLibrary function to load two DLL files, kernel32.dll and user32.dll. It then executes the GetConsoleWindow command to fetch the window handle of the console associated with the process, followed by the ShowWindow command to set the show state of the specified window.

GetConsoleWindow and ShowWindow Commands

VPN Software

Stealers' malware may target VPN software files on victims' systems to disrupt their anonymity and online privacy, steal login credentials and configuration details, conduct targeted surveillance, and exfiltrate sensitive data protected by the VPN.

The malware searches for two specific files, ‘BirdVPN’ and ‘NordVpn.exe,’ within the directory %USERPROFILE%\AppData\Local\ to obtain the username and password for both VPN software.

BirdVPN and NordVPN Files Theft
BirdVPN and NordVPN Files Theft
The malware attempts to steal the ovpn files of ProtonVPN from the %AppData%\Local directory.

ProtonVPN Files Theft
The stealer attempts to steal the OpenVPN ovpn files in the ‘%AppData%\Roaming\OpenVPN\profiling’ directory.

OpenVPN Files Theft

Software

Stealer malware targets third-party software, such as Telegram, Steam, or Discord, to steal user login credentials, authentication tokens, etc. This data enables threat actors to gain unauthorized access, commit identity theft, and potentially generate financial gains.
This kind of stealing can reveal victims’ interests and affiliations, and it can also be used for extortion and resale on the dark web.

The malware attempts to extract a user’s Telegram profile from their endpoint.

Telegram Profile Files Theft
Telegram Profile Files Theft
Telegram Profile Files Theft
The stealer attempts to steal Discord information by searching for .log and .ldb files in the %AppData%\discord\Local Storage\leveldb directory.

Discord Files Theft
The malware attempts to steal data by accessing the registry key ‘Software\Valve\Steam’ and extracting the values of SteamPath, ssfn, config, and .vdf.

Steam Files Theft

Malware’s C2 Server

The threat actor conceals his IP address with a dynamic DNS service that links his IP address 192.169.69.26 to the ‘siyatermi.duckdns.org’ domain.

Malware’s C2 Server
Malware’s C2 Server in Wireshark

The transmission of the stolen information is sent to ‘siyatermi.duckdns.org’ through SOAP message in HTTP protocol.

SOAP is a protocol that is used for structuring messages in web services and facilitating communication between different applications or systems over the internet and the data represented in XML format.

SOAP Data Transmission
The stealer verifies the connection established by the malware to the C2 server.

Connection Verification

Conclusion

RedLine Stealer is a dangerous type of malware that can cause serious harm to both individuals and organizations. It is crucial to protect your systems from RedLine Stealer by using strong passwords, keeping your software up to date, and being cautious about which emails you open and what attachments you download.

MITRE ATT&CK Mapping

MITRE ATT&CK Mapping

Indicators of Compromise (IoCs)

1. %AppData%/winlogon.exe
2. %AppData%/chrome.exe
3. %AppData%/svchost.exe
4. Desktop/NetFlix Checker by xRisky v2.exe
5. siyatermi.duckdns.org:17044
6. 192.169.69.26

YARA Rule

The following YARA rule detects the ‘winlogon.exe’ RedLine malware.

Yara Rule

RedLine Detection With the Yara Rule

RedLine Detection