Analyzing Ursnif’s Behavior Using a Malware Sandbox - VMRay

By VMRay Labs

Published: 2019-06-25 · Archived: 2026-04-05 18:55:16 UTC

Ursnif is a group of malware families based on the same leaked source code. When fully executed Urnsif has the capability to steal banking and online account credentials. In this blog post, we will analyze the payload of a Ursnif sample and demonstrate how a malware sandbox can expedite the investigation process.
Ursnif (also known as Gozi) is a banking Trojan that generally collects system activity, records keystroke data, and keeps track of network and

Access the VMRay Analyzer Report for Ursnif

This blog post will cover a behavioral analysis of a single Ursnif variant. It does not provide comprehensive insights into web injects, infrastructure or attribution. For additional Ursnif analysis see Appendix D.

OLSTEALER steals data from Outlook, including login information, and stores it in a local file. The internal name of the module is visible in Function log:

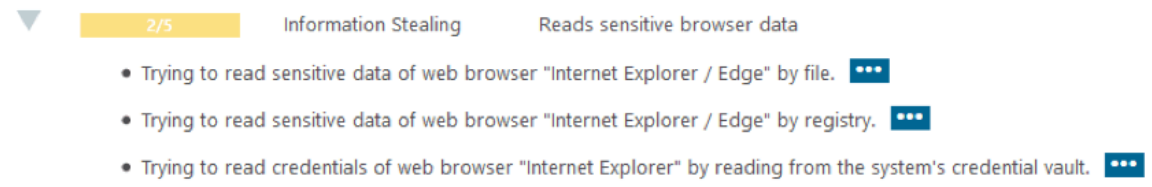
```
[0096.008] lstrlenA (lpString="#OLSTEALER#\n") returned 12
```

The contents of the created file appear as follows:

```
1  #OLSTEALER#
2  SMTP Server:
3  POP3 Server:
4  Email:
5  POP3 User:
```

The IESTEALER module reads Internet Explorer history and passwords.

```
[0096.536] lstrlenA (lpString="#IESTEALER#\n") returned 12
```



After stealing from Internet Explorer, the malware also looks for Thunderbird, though the name of the Thunderbird stealer module (TBSTEALER) did not explicitly appear.

```
[0099.209] StrStrIW (lpFirst="Software\\Mozilla", lpSrch="Thunderbird") returned 0x0
[0099.209] LocalAlloc (uFlags=0x40, uBytes=0x1080) returned 0x72f7c00
[0099.209] RegOpenKeyExW (in: hKey=0xffffffff80000002, lpSubKey="Software\\Mozilla", ulOptions=0x0, samDesired=0x20219,
    phkResult=0x54dfd78 | out: phkResult=0x54dfd78*=0xb5c) returned 0x0
[0099.210] GetProcAddress (hModule=0x7fefd710000, lpProcName="RegEnumKeyExW") returned 0x7fefd72c310
[0099.210] RegEnumKeyExW (in: hKey=0xb5c, dwIndex=0x0, lpName=0x72f7c00, lpcchName=0x54dfd68, lpReserved=0x0, lpClass=0x0,
    lpcchClass=0x0, lpftLastWriteTime=0x0 | out: lpName="Firefox", lpcchName=0x54dfd68, lpClass=0x0, lpcchClass=0x0,
    lpftLastWriteTime=0x0) returned 0x0
```

System Info Gathering

Using built-in Windows system tools Ursnif gathers information about the system. The tools used are:

- *systeminfo.exe* – various info about the system including OS version, installed patches, domain, and basic hardware information
- *net view* – show network shares
- nslookup 127.0.0.1 – local IP
- *tasklist.exe /SVC* – Services
- *driverquery.exe* – Installed drivers
- (Installed software)*reg.exe query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall reg.exe query "HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall*

Data Exfiltration

Ursnif caches stolen data to the hard drive into temp files, compresses them into CAB files, and uploads them.

Steps followed to create the CAB:

1. The various stealer modules create files on the hard drive. Some use the *%TEMP%* directory, others use the random directory created earlier.

Create Temp File	C:\Users\aETAdzjz\AppData\Local\Temp\1D0E.tmp	path = C:\Users\aETAdzjz\AppData\Local\Temp\
Create Temp File	C:\Users\aETAdzjz\AppData\Local\Temp\2855.tmp	path = C:\Users\aETAdzjz\AppData\Local\Temp\
Create Temp File	C:\Users\aETAdzjz\AppData\Local\Temp\1FB1.tmp	path = C:\Users\aETAdzjz\AppData\Local\Temp\
Create Temp File	C:\Users\aETAdzjz\AppData\Local\Temp\E3D6.tmp	path = C:\Users\aETAdzjz\AppData\Local\Temp\
Create Temp File	C:\Users\aETAdzjz\AppData\Local\Temp\DB32.tmp	path = C:\Users\aETAdzjz\AppData\Local\Temp\

Source: https://www.vmray.com/cyber-security-blog/analyzing-ursnif-behavior-malware-sandbox/