Osiris: An Enhanced Banking Trojan

By deugenio

Published: 2018-07-31 · Archived: 2026-04-18 02:11:52 UTC

Research By: Yaroslav Harakhavik and Nikita Fokin

Following our recent analysis of the Kronos banking Trojan, we discovered that Kronos has also now been enhanced to hide its communication with C&C server using Tor. While the author of Kronos continues to remain an issue of controversy, the new banking Trojan, Osiris, is thought to be primarily based on Kronos source code and most likely has the same author.

Background

Kronos banking Trojan first made an appearance on the Russian underground forum, exploit.in, in 2014 where it was advertised for $2000 a month:

Figure 1: The advertisement of Kronos on the exploit.in.

It had the following technical capabilities:

- Form grabber and Zeus-like web-injects compatible with the major browsers (IE, Chrome, Firefox, Microsoft Edge)
- 32-bit and 64-bit ring 3 rootkit
- Antivirus and sandbox bypassing
- Encrypted communications with C&C server
- Credit cards grabber (as an additional module)

From the leaked Kronos panel source code it was discovered that Kronos supposedly had two additional plugins: remote control via VNC and a Keylogger functionality.

In April 2018, however, the promotional campaign of a new banking Trojan called Osiris began.

Figure 2: The advertisement of Osiris from exploit.in forum.

As we had an opportunity to obtain a sample of Osiris we performed an initial analysis and it turned out that the malware highly resembled Kronos.

Kronos Clues

The similarities of Osiris and Kronos first came to light after analyzing the functions which create a global mutex. Kronos uses an MD5 hash of the hard drive serial number and if it fails to get the serial number of a system volume it assigns the mutex name to *MD5("Kronos").*

In the same way, the exact same algorithm is used by Osiris:

Figure 3: Mutex name generation algorithm for Osiris and Kronos.

Comparing the primary functionality of Osiris and Kronos which was previously analyzed by Check Point Research it was confirmed that, in general, both Trojans do an identical job:

Figure 4: Kronos Main vs. Osiris Main.

Further comparison between samples of Kronos and Osiris showed that the two malwares have almost the same functionality, as the following matching capabilities indicate:

- Almost the same executional graph and a lot of totally identical code;
- Global mutex and event names are generated in the identical manner;
- The same persistence methods;
- The same encryption algorithms;
- The same anti-VM and anti-sandboxing techniques;
- The same code obfuscation method (such as using raw syscalls).

As a result, all of these clues show that Osiris is definitely based on Kronos and is actually an enhanced version of Kronos.

Earlier this month it was discovered that Osiris erupted into the wild and was detected as being Kronos by different AV solutions.

Kronos vs Osiris

Resolving ntdll.dll functions

To prevent detection by some monitoring software, Kronos calls native API from its own code without using ntdll.dll as a proxy. The malware then obtains appropriate syscalls numbers from the ntdll.dll and the functions are identified by hashes of their names.

Figure 5: Kronos and Osiris syscalls hashes.

The numbers of syscalls are stored in variables which XORed with the *0x57ED* constant. Every syscall has its own wrapper function.

Figure 6: Kronos and Osiris wrappers for NtSetValueKey.

Anti-VM and Anti-Sandbox

It is now known then that Kronos and Osiris use identical evasions check. Indeed, they search for the existence of different processes and loaded modules which can point to the environment where the malware is executed. The results of these checks are then stored in a dedicated variable.

If a debugger, virtual machine or a sandbox is detected then in that particular moment the flag which represents the machine architecture will be flipped. As a result this will cause a crash during the raw syscalls loading:

Figure 7: Osiris/Kronos inverts the flag of VM was detected.

Figure 8: Architecture dependent calls.

Persistence Mechanism

Kronos copies itself into the file named by the first eight characters of *MD5(system_volume_serial)* in *%APPDATA%\Microsoft\<GENERATED_GUID>* and writes this path to the Registry under *Softwrae\Windows\CurrentVersion\Run* (*HKCU* and, if it was run by admin, in *HKLM*).

Regarding Osiris though, the malware is supposed to use the same well-known method, however our researched sample builds all the corresponding paths and names but does not make itself persistent in the system. This is probably due to the malware still being in development.

Global Objects in OS

As described above, the global mutex name is generated by getting an MD5 hash of the string which represents the system volume serial number. If the operation of getting the serial number fails, the hardcoded string “Kronos” will be used instead.

Osiris like Kronos creates other global objects in OS which can be used as IOCs. There are a global mutex and a global event where names are generated by the same algorithm with the “salt” of this algorithm passed on in code constants. Therefore the following two names will be present in any system infected by Osiris/Kronos:

Mutex name: **Global\{AD3EBBCA-D942-886C-AD3E-CABB824AEA00}**
Event name: **Global\{2C240B38-28B0-DE58-2C24-380BA08C4000}**

Configuration File

Kronos can then download a configuration file with Zeus web-injections from the C&C server and the configuration file must be located in the same path with the Kronos executable. A bot sends a beacon to C&C server which consists of:

1. A hash of the configuration file (or the sequence of ‘X’ characters if there is none).
2. The GUID generated by *CoCreateGuid()* which represents BotId.

Figure 9: Reading Kronos’s configuration.

The researched sample of Osiris did not have such feature so it uses the sequence of ‘X’ characters in the beacon every time it is loaded.

Figure 10: Osiris reading configuration stub.

Process Injection & User Land Rootkit Functionality

Kronos tries to escalate its process token to *SeDebugPrivilege* and injects the malicious thread to other processes:

Figure 11: Pseudocode of the privilege escalation procedure in Kronos.

Although the identical code which is responsible for thread injection is present in Osiris's binary, it is not called. This fact might also indicate that the malware is still currently in development.

Stealing Browsers & Outlook User Data

Kronos and Osiris also use the same technique to collect and decrypt users' browser data and mail client.

The grabbing of Firefox data is executed using the following flow:

1. Osiris gets the path to Profile folder where all the changes of user names in Firefox are stored by accessing *Path* value in *%APPDATA%\Mozilla\Firefox\profiles.ini*.
2. Then it loads the nss3.dll library from the Firefox install path which is gotten from *HKLM\Software\Mozilla\Mozilla Firefox\Current Version\Main\Install Directory* Registry key.
3. The malware then uses nss3.dll functions to decrypt Firefox user data.

Chrome user data is stored in a sqlite database file –*C:\Users\%current_user%\AppData\Local\Google\Chrome\User Data\Default\Login Data*. So, Osiris resolves functions of sqlite3 library and collects data from *Login Data* file using SQL queries.

Outlook Profiles data is searched in several Registry paths:

HKEY_CURRENT_USER\Software\Microsoft\Windows Messaging Subsystem\Profiles\9375CFF0413111d3B88A00104B2A6676

HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows Messaging Subsystem\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676

HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676

HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Profiles\Out

look\9375CFF0413111d3B88A00104B2A6676

The malware also tries to collect the values of the next registry keys:

- IMAP Server
- POP3 Server
- Email
- IMAP Password
- SMTP Password
- POP3 Password

The collected data is then decrypted by *CryptUnprotectData()*.

Osiris Enhancements

As mentioned above, the original Kronos supported VNC and keylogging functionality as additional plugins but Osiris has these capabilities out-of-the-box. It therefore, uses a modified *LibVNCServer* library for providing a remote control over the bot via RFB protocol:

Figure 12: Strings from LibVNCServer library in unpacked Osiris.

Osiris also runs a keylogger thread inside processes where it is able to inject the malicious code. The keylogger thread then installs a keyboard hook using *SetWindowsHookEx()*:

Figure 13: Pseudocode of the keylogger routine in Osiris.

The keylogger callback collects the process name, text of the window's title bar and everything which is typed inside the hooked windows. The main thread then sends the collected content of each window to C&C server.

But the most distinctive feature of Osiris is using Tor for communication with the C&C server alongside basic Kronos encryption techniques. Like Kronos, Osiris also communicates with the C&C server via the HTTP protocol using almost the same commands. (The communication protocol of Kronos is well documented in several sources).

However by using Tor the malware does not interact with the C&C server itself but hides its communication under TLS traffic between a Tor relay and the bot:

Figure 14: Network activity during the connection to C&C.

Conclusions

All of the information provided may indicate that both Trojans, Kronos and Osiris, actually have the same origin and although Kronos has evolved since 2014 it is still detected by the same IOCs. However in light of all improvements seen in Osiris, it should certainly still be considered as a new threat.

IOCs

Mutex	Global\{AD3EBBCA-D942-886C-AD3E-CABB824AEA00}
Event	Global\{2C240B38-28B0-DE58-2C24-380BA08C4000}

References

- Inside the Kronos malware – part 1: https://blog.malwarebytes.com/cybercrime/2017/08/inside-kronos-malware/
- Inside the Kronos malware – part 2: https://blog.malwarebytes.com/cybercrime/2017/08/inside-kronos-malware-p2/
- Overview of the Kronos banking malware rootkit: https://www.lexsi.com/securityhub/overview-kronos-banking-malware-rootkit/?lang=en
- Kronos: decrypting the configuration file and injects: https://www.lexsi.com/securityhub/kronos-decrypting-the-configuration-file-and-injects/?lang=en

Check Point Anti-Bot blade provides protection against this threat:

Trojan-Banker.Win32.Osiris.A

Trojan-Banker.Win32.Osiris.B

Trojan-Banker.Win32.Osiris.C

Trojan-Banker.Win32.Osiris.D

Trojan-Banker.Win32.Osiris.E

Trojan-Banker.Win32.Osiris.F
