

Conti ransomware: Evasive by nature

news.sophos.com/en-us/2021/02/16/conti-ransomware-evasive-by-nature/

February 16, 2021



Editor's note: This is one of a series of articles focused on the Conti ransomware family, which include a detailed analysis of a Conti attack, [A Conti Ransomware Attack Day-By-Day](#), and a guide for what IT administrators can expect when Conti ransomware hits.

For the past several months, both SophosLabs and the Sophos Rapid Response team have been collaborating on detection and behavioral analysis of a ransomware that emerged last year and has undergone rapid growth. The ransomware, which calls itself Conti, is delivered at the end of a series of Cobalt Strike/meterpreter payloads that use reflective DLL injection techniques to push the malware directly into memory.

Because the reflective loaders deliver the ransomware payload into memory, never writing the ransomware binary to the infected computer's file system, the attackers eliminate a critical Achilles' heel that affects most other ransomware families: There is no artifact of the ransomware left behind for even a diligent malware analyst to discover and study.

That isn't to say there aren't artifacts and components to look at. The threat actors involved in attacks using Conti have built a complex set of custom tooling designed not only to obfuscate the malware itself, when it gets delivered, but conceal the internet locations from which the attackers have been downloading it during attacks, and prevent researchers from obtaining a copy of the malware that way as well.

Two-stage loading process

The first stage of the Conti ransomware process involves a Cobalt Strike DLL, roughly 200kb in size, that allocates the memory space needed to decrypt and load meterpreter shellcode into system memory.

Address	Hex	ASCII
0000000001D90000	FC 48 83 E4	F0 E8 C8 00 00 00 41 51 41 50 52 51
0000000001D90010	56 48 31 D2	65 48 88 52 60 48 88 52 18 48 88 52
0000000001D90020	20 48 88 72	50 48 0F B7 4A 4A 4D 31 C9 48 31 C0
0000000001D90030	AC 3C 61 7C	02 2C 20 41 C1 C9 0D 41 01 C1 E2 ED
0000000001D90040	52 41 51 48	88 52 20 88 42 3C 48 01 D0 66 81 78
0000000001D90050	18 08 02 75	72 88 80 88 00 00 00 48 85 C0 74 67
0000000001D90060	48 01 D0 50	88 48 18 44 88 40 20 49 01 D0 E3 56
0000000001D90070	48 FF C9 41	88 34 88 48 01 D6 4D 31 C9 48 31 C0
0000000001D90080	AC 41 C1 C9	0D 41 01 C1 38 E0 75 F1 4C 03 4C 24
0000000001D90090	08 45 39 D1	75 D8 58 44 88 40 24 49 01 D0 66 41
0000000001D900A0	88 0C 48 44	88 40 1C 49 01 D0 41 88 04 88 48 01
0000000001D900B0	D0 41 58 41	58 5E 59 5A 41 58 41 59 41 5A 48 83
0000000001D900C0	EC 20 41 52	FF E0 58 41 59 5A 48 88 12 E9 4F FF
0000000001D900D0	FF FF 5D 6A	00 49 BE 77 69 6E 69 6E 65 74 00 41
0000000001D900E0	56 49 89 E6	4C 89 F1 41 BA 4C 77 26 07 FF D5 48
0000000001D900F0	31 C9 48 31	D2 4D 31 C0 4D 31 C9 41 50 41 50 41
0000000001D90100	BA 3A 56 79	A7 FF D5 E9 93 00 00 00 5A 48 89 C1
0000000001D90110	41 88 88 01	00 00 4D 31 C9 41 51 41 51 6A 03 41
0000000001D90120	51 41 BA 57	89 9F C6 FF D5 EB 79 58 48 89 C1
0000000001D90130	31 D2 49 89	D8 4D 31 C9 52 68 00 32 C0 84 52 52

A portion

of meterpreter shellcode, extracted from memory on an infected machine.

The shellcode, XORed in the DLL, unfurls itself into the reserved memory space, then contacts a command-and-control server to retrieve the next stage of the attack.

This C2 communication is distinctive for a number of reasons. First, the malware appears to be using a sample Cobalt Strike configuration script named `trevor.profile`, published on a [public Github archive](#). The profile serves as a sort of homage to an incident in which security researchers attending a conference [found an insect in a milkshake](#) at a restaurant outside the conference center.

```

249 lines (176 sloc) | 10.7 KB
1  #trevorforget
2  #xx0hcd
3
4  set sleeptime "30000";
5  set jitter    "20";
6  set useragent "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)";
7  set dns_idle  "8.8.8.8";
8  set maxdns    "235";
9

```



An excerpt from the sample Cobalt Strike configuration script

But it doesn't appear that the Conti attackers have modified this sample script very much, which makes the C2 communication notable in two ways: The script designates certain characteristics used during this phase of the attack, including a User-Agent string ("**Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)**") that mimics that of a computer running Windows 7 but, distinctively, fails to identify the specific browser; and a static URI path ("**/us/ky/louisville/312-s-fourth-st.html**") that includes the address of the infamous restaurant where the researcher discovered the bug in their shake.

```

126 http-stager {
127
128     set uri_x86 "/menus.aspx";
129     set uri_x64 "/Menus.aspx";
130
131
132     client {
133
134     #     header "Host" "[REDACTED]";
135         header "Accept" "*/*";
136         header "Accept-Language" "en-US,en;q=0.5";
137         header "Referer" "https://[REDACTED]/us/ky/louisville/312-s-fourth-st.html";
138         header "Connection" "close";
139

```



The sample Cobalt Strike configuration uses a URI path that includes “Menus” (with a capital M) to indicate that the infected machine is running a 64-bit operating system, and to deliver the appropriate payload for that architecture.

The initial connection to the C2 server is to a page named **Menus.aspx** on the server; That page delivers the next payload, which the first one loads into memory — another Cobalt Strike shellcode loader that contains the reflective DLL loader instructions.

Full request URI	Protocol	Request	Destination
http://docns.com/Menus.aspx	HTTP	GET	23.106.160.174
http://docns.com/us/ky/louisville/312-s-fourth-st.html	HTTP	GET	23.106.160.174
http://docns.com/us/ky/louisville/312-s-fourth-st.html	HTTP	GET	23.106.160.174
http://tapavi.com/Menus.aspx	HTTP	GET	23.82.140.137
http://tapavi.com/us/ky/louisville/312-s-fourth-st.html	HTTP	GET	23.82.140.137
http://tapavi.com/us/ky/louisville/312-s-fourth-st.html	HTTP	GET	23.82.140.137
http://tapavi.com/us/ky/louisville/312-s-fourth-st.html	HTTP	GET	23.82.140.137
http://tapavi.com/us/ky/louisville/312-s-fourth-st.html	HTTP	GET	23.82.140.137
http://tapavi.com/us/ky/louisville/312-s-fourth-st.html	HTTP	GET	23.82.140.137
http://tapavi.com/us/ky/louisville/312-s-fourth-st.html	HTTP	GET	23.82.140.137
http://tapavi.com/us/ky/louisville/312-s-fourth-st.html	HTTP	GET	23.82.140.137



If that works successfully, the malware then contacts the “312-s-fourth-st.html” page on the same C2 server. The attackers only trigger these chains of events during an active attack, placing the ransomware binary on the C2 server so that it can be retrieved by this process only while the attack is ongoing, and removing it immediately afterwards.

Elusive ransomware payloads

Because of the ephemeral nature of the placement of the ransomware payload, analysts had difficulty obtaining samples for research. But we were able to salvage some of the in-memory code from infected computers where the malware was still running.

The ransomware process is not particularly unique, but it does reveal the ransomware creator’s ongoing interest in thwarting analysis by security researchers.

The ransomware itself uses a relatively common anti-analysis technique sometimes referred to as “API-by-hash,” in which Conti uses hash values to call specific API functions; Conti has an added layer of encryption over the top of these hashes to further complicate the work of a reverse engineer. The malware has to perform two cycles of decryption on itself in order to perform those functions.

Among the behavior observed by responders, the ransomware immediately begins a process of encrypting files while, at the same time, sequentially attempting to connect to other computers on the same network subnet, in order to spread to nearby machines, using the SMB port.

TCPView - Sysinternals: www.sysinternals.com

File Options Process View Help

Process	PID	Protocol	Local Address	Local Port	Remote Address	Remote Port
	10180	TCP	192.168.89.135	10978	192.168.89.0	445
	10180	TCP	192.168.89.135	10979	192.168.89.1	445
	10180	TCP	192.168.89.135	10980	192.168.89.2	445
	10180	TCP	192.168.89.135	10981	192.168.89.3	445
	10180	TCP	192.168.89.135	10982	192.168.89.4	445
	10180	TCP	192.168.89.135	10983	192.168.89.5	445
	10180	TCP	192.168.89.135	10984	192.168.89.6	445
	10180	TCP	192.168.89.135	10985	192.168.89.7	445
	10180	TCP	192.168.89.135	10986	192.168.89.8	445
	10180	TCP	192.168.89.135	10987	192.168.89.9	445
	10180	TCP	192.168.89.135	10988	192.168.89.10	445
	10180	TCP	192.168.89.135	10989	192.168.89.11	445
	10180	TCP	192.168.89.135	10990	192.168.89.12	445
	10180	TCP	192.168.89.135	10991	192.168.89.13	445
	10180	TCP	192.168.89.135	10992	192.168.89.14	445
	10180	TCP	192.168.89.135	10993	192.168.89.15	445
	10180	TCP	192.168.89.135	10994	192.168.89.16	445
	10180	TCP	192.168.89.135	10995	192.168.89.17	445
	10180	TCP	192.168.89.135	10996	192.168.89.18	445
	10180	TCP	192.168.89.135	10997	192.168.89.19	445
	10180	TCP	192.168.89.135	10998	192.168.89.20	445
	10180	TCP	192.168.89.135	10999	192.168.89.21	445
	10180	TCP	192.168.89.135	11000	192.168.89.22	445
	10180	TCP	192.168.89.135	11001	192.168.89.23	445
	10180	TCP	192.168.89.135	11002	192.168.89.24	445
	10180	TCP	192.168.89.135	11003	192.168.89.25	445
	10180	TCP	192.168.89.135	11004	192.168.89.26	445
	10180	TCP	192.168.89.135	11005	192.168.89.27	445
	10180	TCP	192.168.89.135	11006	192.168.89.28	445
	10180	TCP	192.168.89.135	11007	192.168.89.29	445
	10180	TCP	192.168.89.135	11008	192.168.89.30	445

SOPHOS labs

SMB scanning by Conti during the infection

Conti’s developers have hardcoded the RSA public key the ransomware uses to perform its malicious encryption into the ransomware (files are encrypted using the AES-256 algorithm). This isn’t unusual; It means that it can begin encrypting files even if the malware is unable to contact its C2.

Unfortunately, that isn’t the only threat this ransomware poses to its targets: Conti ransomware has also adopted a “leaks” site like several other ransomware threat actor groups. The attackers spend some time on the target network and exfiltrate sensitive, proprietary information to the cloud (in recent attacks, the threat actors have used the cloud storage provider Mega).

```
*readme.txt - Notepad
File Edit Format View Help
All of your files are currently encrypted by CONTI strain.

As you know (if you don't - just "google it"), all of the data that has been encrypted by our software cannot be recovered by any means without contacting our team directly.
If you try to use any additional recovery software - the files might be damaged, so if you are willing to try - try it on the data of the lowest value.

To make sure that we REALLY CAN get your data back - we offer you to decrypt 2 random files completely free of charge.

You can contact our team directly for further instructions through our website :

TOR VERSION :
(you should download and install TOR browser first https://torproject.org)

http://conti[REDACTED].onion/

HTTPS VERSION :
https://conti[REDACTED]

YOU SHOULD BE AWARE!
Just in case, if you try to ignore us. We've downloaded a pack of your internal data and are ready to publish it on out news website if you do not respond. So it will be better for both sides if you contact us as soon as possible.

---BEGIN ID---
[REDACTED]
---END ID---
```



Under a header labeled *YOU SHOULD BE AWARE!*, the ransom note threatens, “Just in case, if you try to ignore us. We’ve downloaded a pack of your internal data and are ready to publish it on out (sic) news website if you do not respond. So it will be better for both sides if you contact us as soon as possible.”

Detection guidance

Conti ransomware, on its own, is unable to bypass the CryptoGuard feature of Sophos Intercept X; Our endpoint products may detect components of Conti under one or more of the following definitions: **HPmal/Conti-B**, **Mem/Conti-B**, **Troj/Swrort-EZ**, **Troj/Ransom-GEM**, or **Mem/Meter-D**. Network protection products like the Sophos XG firewall can also block the malicious C2 addresses to prevent the malware from retrieving its payloads and completing the infection process.

Indicators of compromise for malware samples examined in this research has been [posted to the SophosLabs Github](#).