New Malware with Ties to SunOrcal Discovered

researchcenter.paloaltonetworks.com/2017/11/unit42-new-malware-with-ties-to-sunorcal-discovered/

Josh Grunzweig, Jen Miller-Osborn November 10, 2017

By Josh Grunzweig and Jen Miller-Osborn

November 10, 2017 at 1:00 PM

Category: Unit 42

Tags: Reaver, SunOrcal

Summary
Unit 42 has discovered a new malware family we've named "Reaver" with ties to attackers who use SunOrcal malware. SunOrcal activity has been documented to at least 2013, and based on metadata surrounding some of the C2s, may have been active as early as 2010. The new family appears to have been in the wild since late 2016 and to date we have only identified 10 unique samples, indicating it may be sparingly used. Reaver is also somewhat unique in the fact that its final payload is in the form of a Control panel item, or CPL file. To date, only 0.006% of all malware seen by Palo Alto Networks employs this technique, indicating that it is in fact fairly rare.
While we don't have information on the intended targets in this case, previous reports on this activity have identified targeting primarily among the "Five Poisons" which are movements the Chinese government perceives as dangerous. They are:

- Uyghurs, particularly those supporting East Turkestan independence
- Tibetans, particularly those supportive of Tibetan independence

- Falun Gong practitioners
- Supporters of Taiwan independence
- Supporters of Chinese democracy

The attackers used both families concurrently from late last year through November 2017 and there is some C2 infrastructure overlap between the two families, as well as links to historical reporting. We explore those ties and provide an in-depth analysis of the new malware below.

Reaver Malware Analysis

To date, Palo Alto Networks Unit 42 has identified 10 unique samples and three distinct variants of a new malware family we have named “Reaver”. As such, we identify each variant as Reaver.v1, Reaver.v2, and Reaver.v3.

Reaver.v1 has been observed delivering a payload that uses HTTP for network communication, while versions 2 and 3 use a payload that uses raw TCP connections for this communication.

The flow for Reaver is as shown:

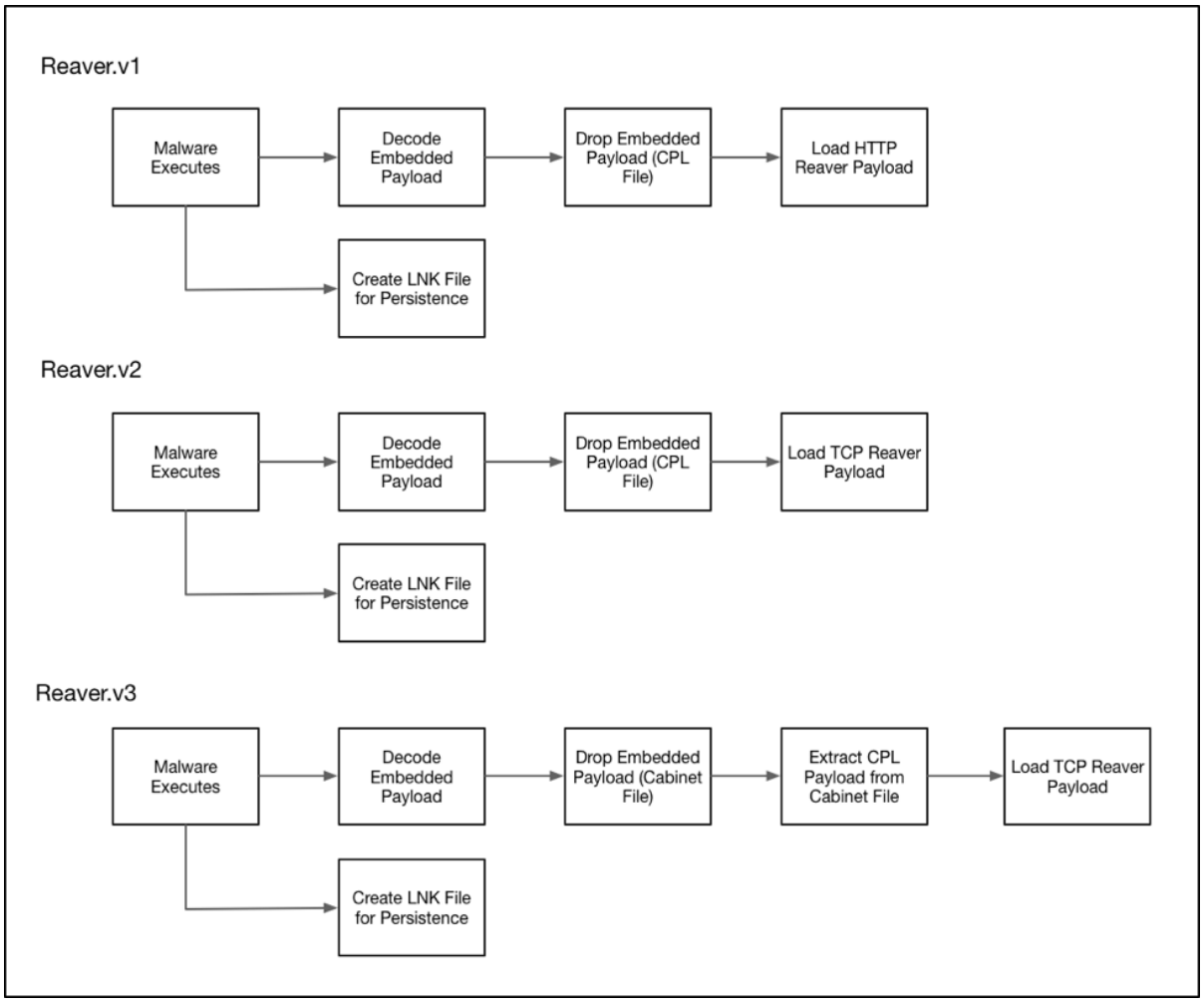

Figure 1 Reaver execution flow diagram

Reaver.v1

The earliest variant of Reaver begins by attempting to enable the SeDebugPrivilege privilege for the running process. In the event this is successful the malware will use the following path to store any dropped files:

%COMMONPROGRAMFILES%\services\

In the event it is not successful, this alternative path will be used instead:

%APPDATA%\microsoft\mmc\

It proceeds to load and decrypt and embedded bitmap resource file. This decrypted data is written to the following location:

%TEMP%\WUpdate.~tmp

This 'WUpdate.~tmp' file is then copied to a filename of 'Applet.cpl', which is placed in the previously identified file path.

The malware proceeds to identify the file path of either the common startup folder, or the user's startup folder depending on if the SeDebugPrivilege privilege was obtained. In the event this privilege was obtained, the common startup folder is queried by reading the following registry key:

HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Common Startup

Alternatively, if the privilege was unable to be obtained, Reaver.v2 will obtain the user's startup folder by querying the following registry key:

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Startup

Reaver proceeds to write a shortcut file to '%TEMP%\~WUpdate.lnk'. This file is then copied to a filename of 'Windows Update.lnk', which is placed in the startup path previously identified. This shortcut file points to the path of the previously written 'Applet.cpl' file. Finally, Reaver.v1 will execute the '~WUpdate.lnk' file in a new process, thus loading the recently dropped malicious CPL file.

Reaver.v2

Reaver.v2 begins by attempting to enable the SeDebugPrivilege privilege for the running process. In the event this is successful, the malware will use the following path to store any dropped files:

%COMMONPROGRAMFILES%\services\

In the event it is not successful, this alternative path will be used instead:

%APPDATA%\microsoft\mmc\

Reaver.v2 proceeds to decrypt an embedded file using a simple XOR obfuscation routine. This file is written to the following file path:

% TEMP%\Update.~tmp

After the file is written, it is then copied to a filename of 'winhelp.cpl' in the directory that was initially chosen. After this file is copied, the original 'Update.~tmp' file is deleted. At this stage the malware will identify the correct startup path using the same technique witnessed in earlier variants.
A shortcut file is generated in the following path:

%TEMP%\~Update.lnk

This '~Update.lnk' file is then copied to a filename of 'Windows help.lnk', which is placed in the startup path previously identified. This shortcut file points to the path of the previously written 'winhelp.cpl' file. It will specifically load this CPL file via a call to the built-in Microsoft Windows 'control.exe' utility. Finally, Reaver.v2 will execute the '~Update.lnk' file in a new process, thus loading the recently dropped malicious CPL file.

Reaver.v3
Like Reaver.v2, Reaver.v3 begins by attempting to enable the SeDebugPrivilege privilege for the running process. In the event this is successful, the malware will use the following path to store any dropped files:

%COMMONPROGRAMFILES%\services\

In the event it is not successful, this alternative path will be used instead:

%APPDATA%\microsoft\credentials\

Reaver.v3 proceeds to write an embedded Microsoft Cabinet (CAB) file to the following location:

%TEMP%\winhelp.dat

This cabinet file is then extracted to the previously identified file path. The contents of this cabinet file consist of a Microsoft Control Panel item with a filename of 'winhelp.cpl'.
Much like the previous version of Reaver, Reaver.v3 will query the necessary registry keys to determine the correct startup path to use. Again, a shortcut file is written to the %TEMP% path with a name of '~Update.lnk', which is in turn copied to the identified startup path with a filename of 'Windows help.lnk'. This shortcut file calls the built-in 'control.exe' utility to in turn load the previously dropped malicious CPL file of 'winhelp.cpl'.
Finally, the malware calls the 'winhelp.cpl' file in a new process via the following command:

control [path_previously_identified]\winhelp.cpl

Reaver HTTP Payload
The malicious CPL payload of Reaver has the following two exported functions:

- CPlApplet
- DllEntryPoint

When the CPlApplet function is loaded, Reaver will initially determine if the SeDebugPrivilege privilege is able to be obtained. The malware proceeds to decrypt and embedded configuration of 128 bytes using a simple XOR routine. The following example decrypted configuration is as follows:

```
1  00000000: 77 77 77 2E 74 61 73 68  64 71 64 78 70 2E 63 6F  www.tashdqdxp.co
2  00000010: 6D 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  m...............
3  00000020: 38 30 00 00 00 00 00 00  00 00 00 00 00 00 00 00  80..............
4  00000030: 33 30 00 00 00 00 00 00  00 00 00 00 00 00 00 00  30..............
5  00000040: 57 69 6E 64 6F 77 73 20  55 70 64 61 74 65 00 00  Windows Update..
6  00000050: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
7  00000060: 41 70 70 6C 65 74 00 00  00 00 00 00 00 00 00 00  Applet..........
8  00000070: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

As we can see, the following information is present within this configuration:

- Remote Command and Control (C2) server
- Remote port
- Sleep timer

Reaver continues to collect various information from the victim machine, including the following:

- CPU speed
- Computer name
- Username
- IP Address
- Microsoft Windows version
- Physical and virtual memory information

The malware proceeds to communicate with the remote server via HTTP GET and POST requests. Data that is sent is compressed and then base64-encoded before being included in the requests.
We have observed the following capabilities of this payload:

- Get drive information
- Read files
- Write files
- Delete files

- Move files
- Spawn processes
- Create directories

Reaver TCP Payload
The malicious CPL payload of Reaver has the following three exported functions:

- ServiceMain
- CPlApplet
- DllEntryPoint

When the malware is initially loaded, DllEntryPoint will be called, which in turn will call a function that is responsible for decompressing a blob of data. The decompressed data consists of various key/value pairings that represent important strings used by Reaver. An example of this decompressed data can be seen below:

```
1   RA@10001=ole32.dll
2   RA@10002=CoCreateGuid
3   RA@10003=Shlwapi.dll
4   RA@10004=SHDeleteKeyA
5   RA@10005=wininet.dll
6   RA@10006=InternetOpenA
7   [TRUNCATED]
8   RA@10288=%s\%s
9   RA@10289=CMD.EXE
10  RA@10290=%s=
11  RA@10311=\%sctr.dll
12  RA@10312=\uc.dat
13  RA@10313=ChangeServiceConfig2A
14  RA@10314=QueryServiceConfig2A
```

When the malware wishes to retrieve one of these decoded strings, it will simply call a function with an integer argument that is responsible for providing it. For example, calling this function with an argument of ‘10001’ would retrieve a string of ‘ole32.dll’.
The DllEntryPoint function proceeds to attempt to obtain the SeDebugPrivilege privilege, and also calls WSAStartup for future network activity.
When the CPlApplet function is loaded, it will begin by decompressing an embedded configuration using the same compression algorithm used previously. An example of this decompressed configuration may be seen below:

```
debug027:005C21E8 dword_5C21E8 dd 0                    ; DATA XREF: Stack[00000D4C]:0022E778↑o
debug027:005C21EC dd 0
debug027:005C21F0 dd 0
debug027:005C21F4 dd 50h
debug027:005C21F8 dd 2710h
debug027:005C21FC dd 1
debug027:005C2200 dd 0
debug027:005C2204 dd 42F68h
debug027:005C2208 aWww_tashdqdxp_ db 'www.tashdqdxp.com',0
debug027:005C221A db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
debug027:005C221A db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
debug027:005C221A db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
debug027:005C221A db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
debug027:005C221A db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
debug027:005C221A db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
debug027:005C221A db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
debug027:005C221A db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
debug027:005C22C8 aHelpsvc db 'helpsvc',0
debug027:005C22D0 db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
debug027:005C22E0 aWindowsMultime db 'Windows Multimedia Service for media devices',0
debug027:005C230D db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
debug027:005C230D db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
debug027:005C230D db 0, 0, 0, 0, 0
debug027:005C2340 aWindowsMulti_0 db 'Windows Multimedia Service',0
debug027:005C235B db 0, 0, 0, 0, 0
debug027:005C2360 a2016TqS db '2016-tq-s',0
debug027:005C236A db    0
debug027:005C236B db    0
```

Figure 2 Decompressed Reaver configuration

This configuration contains multiple pieces of information, including the following:

- Network port
- Sleep timer between network requests
- Remote Command and Control (C2)
- Service Name
- Service Description
- Service Display Name
- Hardcoded String. This may be either a campaign identifier, or perhaps a malware versioning string.

The malware proceeds to check to see if the original dropped malware file exists. In the event it does, Reaver will move this file to '%TEMP%\~FJIOW.tmp' and delete this new file. This simply acts as cleanup to ensure original file artifacts no longer reside on the infected machine. Reaver will then install itself as a service in the event it is running with SeDebugPrivilege privileges. The service is configured with a name, description, and display name that is provided within the configuration.
Reaver continues to collect various information from the victim machine, including the following:

- Computer name
- Volume serial number
- Microsoft Windows version
- CPU speed
- ANSI code page
- OEM code page identifier for the operating system

- Physical and virtual memory information

Reaver encrypts this data using an incremental XOR key and uploads it to the configured remote server on the port specified. The following example Python code shows how this encryption takes place:

```
1  c = 0
2  out = ""
3  for d in data:
4    out += chr((ord(d) ^ ((c % 256) + 92)) & 0xFF)
5    c += 1
```

After this data is exfiltrated, the malware expects 8 bytes of data that contains two DWORDs. These DWORDs contain both a major command and a sub-command.
The following capabilities have been observed in this payload:

- Get drive information
- Modify files
- Modify directories
- Modify registry
- Spawn process
- Terminate process
- Modify services
- Kill self

Ties to SunOrcal

Reaver was used concurrently with SunOrcal over the past year, to include two Reaver samples dropped from zip files hosted on a domain also being used as a SunOrcal C2 (www.fyoutside[.]com), and there is also passive DNS overlap amongst the C2s. Specifically, Reaver to date has used www.tashdqdxp[.]com for C2, which overlaps with www.weryhstui[.]com, another C2 used by SunOrcal samples during the same timeframe. Both domains have resolved to 98.126.156[.]210. Several of those same SunOrcal samples were also using www.fyoutside[.]com as an additional C2. This led to further C2 ties within SunOrcal samples, to include samples beaconing to www.olinaodi[.]com; all of this is shown below in Figure 3. The latter has been previously reported in activity targeting Hong Kong democracy activists and that activity is in turn tied to a report targeting Tibetan, Hong Kong, and Taiwanese activists, and another blog about targeting Taiwanese activists.

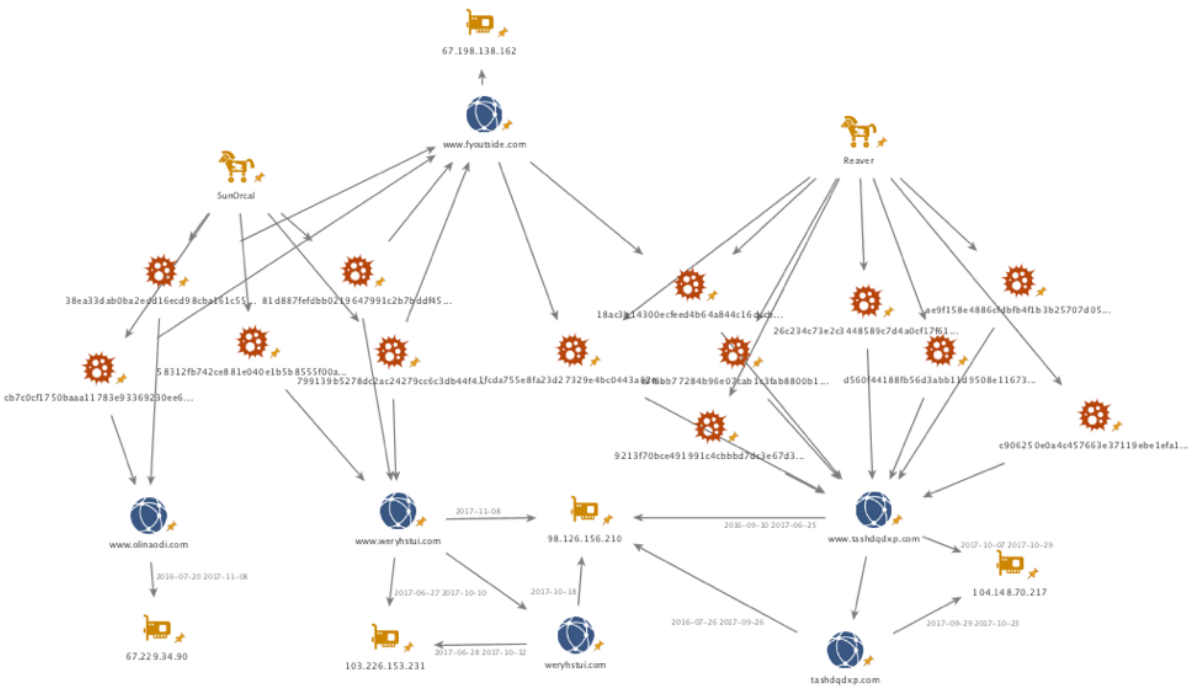

Figure 3. Chart showing overlaps between Reaver and SunOrcal. All IOCs are in the appendix at the end of this blog.

Conclusion

The attackers behind SunOrcal, whose activity dates to at least 2013 and possibly 2010, remain active and are still developing new custom malware to use against their targets. The new malware, Reaver, appears to have been in the wild since late 2016 with less than a dozen known samples, among which there are three variants. It is also unique in the fact that its final payload is in a CPL file, a technique which Palo Alto Networks has seen with only 0.006% of all malware samples we have analyzed. The attackers used both families concurrently from late last year through November 2017 and there is some C2 infrastructure overlap between the two families, as well as links to historical reporting. We will continue to monitor these attackers for new activity and report as appropriate.

Palo Alto Networks customers are protected by the following:

- Wildfire and Traps identifies both malware families as malicious.
- The C2 domains are blocked via Threat Prevention.

Appendix

SHA2556 – Reaver.v1

d560f44188fb56d3abb11d9508e1167329470de19b811163eb1167534722e666

SHA2556 – Reaver.v2
98eb5465c6330b9b49df2e7c9ad0b1164aa5b35423d9e80495a178eb510cdc1c
05ddbd0506ec95fb460b3994e5b21cdb0418ba4aa406374ca1b91249349b7640

SHA2556 – Reaver.v3
18ac3b14300ecfeed4b64a844c16dccb06b0e3513d0954d6c6182f2ea14e4c92
c0f8bb77284b96e07cab1c3fab8800b1bbd030720c74628c4ee5666694ef903d
9213f70bce491991c4cbbbd7dc3e67d3a3d535b965d7064973b35c50f265e59b
26c234c73e2c3448589c7d4a0cf17f615ad3666541a4e611e2d8b77637205bcf
ae9f158e4886cfdbfb4f1b3b25707d05f6fd873d0be9d8e7334a2c28741228ee
1fcda755e8fa23d27329e4bc0443a82e1c1e9a6c1691639db256a187365e4db1
c906250e0a4c457663e37119ebe1efa1e4b97eef1d975f383ac3243f9f09908c
1813f10bcf74beb582c824c64fff63cb150d178bef93af81d875ca84214307a1

SHA256 – SunOrcal
799139b5278dc2ac24279cc6c3db44f4ef0ea78ee7b721b0ace38fd8018c51ac
81d887fefdbb0219647991c2b7bddf45c2fede4dc6fc18408f1706e0279615b2
58312fb742ce881e040e1b5b8555f00a402b8dd4fc886acaae2f862040b3bfc5
38ea33dab0ba2edd16ecd98cba161c550d1036b253c8666c4110d198948329fb
cb7c0cf1750baaa11783e93369230ee666b9f3da7298e4d1bb9a07af6a439f2f

C2 domains and IP addresses
www.tashdqdxp[.]com
www.weryhstui[.]com
www.fyoutside[.]com

www.olinaodi[.]com
104.148.70[.]217
98.126.156[.]210