CVE-2023-27997 - Forensics short notice for XORtigate

Tue 13 June 2023 by Maxime Chouquet in Csirt.

Context

Following the release of the CVE-2023-27997 on our blog and its section "A few notes for blue teamers", here are some Forensics tips and the traces left by a successful execution of an exploit of this CVE.

This blog is not intended to represent a real case of a CVE-2023-27997 compromise, it is for information purposes only and is intended to help identify a real compromise.

The environment used is a test environment composed of a single FortiOS v7.2.4 build 1396 and may therefore be the cause of a gap between our tests and yours. Similarly, this equipment contains no storage space or centralized logs (e.g. FortiAnalyzer), so everything has been done in memory. *Finally, please note that the equipment logging has not been configured to the state of the art, only a quick search has been conducted.*

Collection approach

Taking this into account, we first tried to locate the execution of different exploits in the equipment's GUI. Some crash of the `/bin/sslvpnd` process helps to identify a few, but as explained in the article (and tested), the vulnerability can be exploited without crashing the `/bin/sslvpnd` process.

In order to obtain interesting traces, two approaches were used:

1) Use of the exploit to take control of the equipment in order to collect system evidence and RAM;

This step will not be detailed here. However, it is important to note that the exploitation of the CVE allows the entire device to be compromised, without leaving any trace on the disk, and allows the attacker to modify any evidence if he wishes. *Likewise, we will not go into the various ways of setting up persistence.*

2) Use of the equipment command line interpreter;

We tried that way because it fits with our test environment, you may access these logs in a different way.

```
# Set up the logs on your prompt
CLI > diagnose debug enable
```

```
# Set up the configuration
diagnose debug application sslvpn 0xflag_you_want_in_hexa
```

The configurations set up to identify the first traces of CVE execution are the following:

```
sslvpn debug level is 2167679 (0x21137f)

Log Level:
Emergency               0x00000001 : enable
Alert                   0x00000002 : enable
Critical                0x00000004 : enable
Error                   0x00000008 : enable
Warning                 0x00000010 : enable
Notice                  0x00000020 : enable
Information             0x00000040 : enable
Debug                   0x00000080 : disable

Log Module:
SSL Information         0x00000100 : enable
HTTP proxy              0x00000200 : enable
RADIUS Frame IP         0x00000400 : disable
Mod gzip                0x00000800 : disable
Authentication          0x00001000 : enable
FTP                     0x00002000 : disable
SMB                     0x00004000 : disable
SFTP                    0x00008000 : disable
HTTP request            0x00010000 : enable
DNS                     0x00020000 : disable
DTLS state              0x00040000 : disable
DTLS tunnel             0x00080000 : disable
LibPPP                  0x00100000 : disable
WebSocket               0x00200000 : enable
Telnet                  0x00400000 : disable
SSH                     0x00800000 : disable
RDP                     0x01000000 : disable
VNC                     0x02000000 : disable
```

and,

```
sslvpn debug level is 2167807 (0x2113ff)

Log Level:
Emergency               0x00000001 : enable
Alert                   0x00000002 : enable
Critical                0x00000004 : enable
Error                   0x00000008 : enable
Warning                 0x00000010 : enable
Notice                  0x00000020 : enable
Information             0x00000040 : enable
Debug                   0x00000080 : enable

Log Module:
SSL Information         0x00000100 : enable
HTTP proxy              0x00000200 : enable
RADIUS Frame IP         0x00000400 : disable
Mod gzip                0x00000800 : disable
Authentication          0x00001000 : enable
FTP                     0x00002000 : disable
SMB                     0x00004000 : disable
SFTP                    0x00008000 : disable
HTTP request            0x00010000 : enable
DNS                     0x00020000 : disable
DTLS state              0x00040000 : disable
DTLS tunnel             0x00080000 : disable
LibPPP                  0x00100000 : disable
WebSocket               0x00200000 : enable
Telnet                  0x00400000 : disable
SSH                     0x00800000 : disable
RDP                     0x01000000 : disable
VNC                     0x02000000 : disable
```

It has to be noted that we found evidence of the execution on both configurations. Thus, the Log level at "Debug" (0x00000080) may not be necessary to detect the exploit.

Short analysis

Below is an extract of the executions that can be found on the test environment with the 2nd collection approach.

This summary is not an exhaustive analysis, but it does highlight a few patterns.

32-bit environment

The 1st step which appears is the step to obtain the salt. As explained in the blog, `salt` is a random value created by the server, which can be retrieved by issuing a GET request to `/remote/info`.

The second request that appears `/remote/hostcheck_validate` corresponds to the set up of the heap.

```
[2000:root:2]req: /remote/info
[2000:root:2]capability flags: 0x1cdf
[2000:root:2]SSL state:fatal decode error (10.0.1.3)
[2000:root:0]ap_read,105, error=1, errno=0 ssl 0x7f13c25fd000 Success, error:0A000126:SSL routines::unexpected eof while reading
[2000:root:3]SSL state:before SSL initialization (10.0.1.3)
[2000:root:3]SSL state:before SSL initialization (10.0.1.3)
[2000:root:3]no SNI received
[2000:root:3]SSL state:SSLv3/TLS read client hello (10.0.1.3)
[2000:root:3]SSL state:SSLv3/TLS write server hello (10.0.1.3)
[2000:root:3]SSL state:SSLv3/TLS write certificate (10.0.1.3)
[2000:root:3]SSL state:SSLv3/TLS write key exchange (10.0.1.3)
[2000:root:3]SSL state:SSLv3/TLS write server done (10.0.1.3)
[2000:root:3]SSL state:SSLv3/TLS write server done (10.0.1.3)
[2000:root:3]SSL state:SSLv3/TLS read client key exchange (10.0.1.3)
[2000:root:3]SSL state:SSLv3/TLS read change cipher spec (10.0.1.3)
[2000:root:3]SSL state:SSLv3/TLS read finished (10.0.1.3)
[2000:root:3]SSL state:SSLv3/TLS write change cipher spec (10.0.1.3)
[2000:root:3]SSL state:SSLv3/TLS write finished (10.0.1.3)
[2000:root:3]SSL state:SSL negotiation finished successfully (10.0.1.3)
[2000:root:3]SSL established: TLSv1 ECDHE-RSA-AES256-SHA
[2000:root:2]req: /remote/hostcheck_validate
[2000:root:2]hostcheck validate failed
[2000:root:4]SSL state:before SSL initialization (10.0.1.3)
[2000:root:4]SSL state:fatal decode error (10.0.1.3)
[2000:root:4]SSL state:error:(null)(10.0.1.3)
[2000:root:4]SSL_accept failed, 1:unexpected eof while reading
[2000:root:5]SSL state:before SSL initialization (10.0.1.3)
[2000:root:5]SSL state:fatal decode error (10.0.1.3)
[2000:root:5]SSL state:error:(null)(10.0.1.3)
[2000:root:5]SSL_accept failed, 1:unexpected eof while reading
[2000:root:6]SSL state:before SSL initialization (10.0.1.3)
[2000:root:6]SSL state:before SSL initialization (10.0.1.3)
[2000:root:6]no SNI received
[2000:root:6]SSL state:SSLv3/TLS read client hello (10.0.1.3)
[2000:root:6]SSL state:SSLv3/TLS write server hello (10.0.1.3)
[2000:root:6]SSL state:SSLv3/TLS write certificate (10.0.1.3)
[2000:root:6]SSL state:SSLv3/TLS write key exchange (10.0.1.3)
[2000:root:6]SSL state:SSLv3/TLS write server done (10.0.1.3)
[2000:root:6]SSL state:SSLv3/TLS write server done (10.0.1.3)
[2000:root:6]SSL state:SSLv3/TLS read client key exchange (10.0.1.3)
[2000:root:6]SSL state:SSLv3/TLS read change cipher spec (10.0.1.3)
[2000:root:6]SSL state:SSLv3/TLS read finished (10.0.1.3)
[2000:root:6]SSL state:SSLv3/TLS write session ticket (10.0.1.3)
[2000:root:6]SSL state:SSLv3/TLS write change cipher spec (10.0.1.3)
[2000:root:6]SSL state:SSLv3/TLS write finished (10.0.1.3)
[2000:root:6]SSL state:SSL negotiation finished successfully (10.0.1.3)
[2000:root:6]SSL established: TLSv1 ECDHE-RSA-AES256-SHA
```

32bit: first evidence

At the very beginning of the following capture, we noticed the `/remote/error` request, which is a way found to force the reallocation of the buffer for the HTTP response, in order for "out" to take its place in the heap. "out" is then allocated on the top of the SSL structure.

Next, the following requests are related to the "xorverflow" which rewrites the SSL structure. It is important to note that, depending on the payload used, the volume of these requests may vary in the logs. This means that depending on the opponent you are looking for, the behavior can vary in the logs:

- **Standard threat actor:** For overflow to be optimal and functional, these requests need to be sent at extremely short intervals. The timing of these requests is therefore more important than their volume (from a forensics/detection perspective).

- **Advanced Threat Actor:** If the heap is set up correctly, there is no need to send successive requests. The rewriting of the SSL structure can be perfectly spaced out over time.

32bit second evidence

One step that is specific to the CVE-2023-27997 exploitation on 32bit is the queries that are made to the web page that echoes back some inputs. It is these requests that we found in the last section of the screen.

The URL is not described here (as in the original blog), but it should be noted that different approaches (or URL requested) may work.

64-bit environment

The 64-bit environment broadly follows the same logic when we look only at the existing traces. First, the salt is retrieved and the `/remote/sam1/logout` that appears after corresponds to the set up of the heap.

```
[1800:root:2]SSL state:before SSL initialization (10.0.1.3)
[1800:root:2]no SNI received
[1800:root:2]SSL state:SSLv3/TLS read client hello (10.0.1.3)
[1800:root:2]SSL state:SSLv3/TLS write server hello (10.0.1.3)
[1800:root:2]SSL state:SSLv3/TLS write certificate (10.0.1.3)
[1800:root:2]SSL state:SSLv3/TLS write key exchange (10.0.1.3)
[1800:root:2]SSL state:SSLv3/TLS write server done (10.0.1.3)
[1800:root:2]SSL state:SSLv3/TLS write server done:(null)(10.0.1.3)
[1800:root:2]SSL state:SSLv3/TLS write server done (10.0.1.3)
[1800:root:2]SSL state:SSLv3/TLS read client key exchange (10.0.1.3)
[1800:root:2]SSL state:SSLv3/TLS read change cipher spec (10.0.1.3)
[1800:root:2]SSL state:SSLv3/TLS read finished (10.0.1.3)
[1800:root:2]SSL state:SSLv3/TLS write session ticket (10.0.1.3)
[1800:root:2]SSL state:SSLv3/TLS write change cipher spec (10.0.1.3)
[1800:root:2]SSL state:SSLv3/TLS write finished (10.0.1.3)
[1800:root:2]SSL state:SSL negotiation finished successfully (10.0.1.3)
[1800:root:2]SSL established: TLSv1 ECDHE-RSA-AES256-SHA
[1800:root:2]req: /remote/info
[1800:root:2]capability flags: 0x1cdf
[1800:root:2]SSL state:warning close notify (10.0.1.3)
[1800:root:3]SSL state:before SSL initialization (10.0.1.3)
[1800:root:3]SSL state:before SSL initialization (10.0.1.3)
[1800:root:3]no SNI received
[1800:root:3]SSL state:SSLv3/TLS read client hello (10.0.1.3)
[1800:root:3]SSL state:SSLv3/TLS write server hello (10.0.1.3)
[1800:root:3]SSL state:SSLv3/TLS write certificate (10.0.1.3)
[1800:root:3]SSL state:SSLv3/TLS write server done (10.0.1.3)
[1800:root:3]SSL state:SSLv3/TLS write server done:(null)(10.0.1.3)
[1800:root:3]SSL state:SSLv3/TLS write server done (10.0.1.3)
[1800:root:3]SSL state:SSLv3/TLS read client key exchange (10.0.1.3)
[1800:root:3]SSL state:SSLv3/TLS read change cipher spec (10.0.1.3)
[1800:root:3]SSL state:SSLv3/TLS read finished (10.0.1.3)
[1800:root:3]SSL state:SSLv3/TLS write session ticket (10.0.1.3)
[1800:root:3]SSL state:SSLv3/TLS write change cipher spec (10.0.1.3)
[1800:root:3]SSL state:SSLv3/TLS write finished (10.0.1.3)
[1800:root:3]SSL state:SSL negotiation finished successfully (10.0.1.3)
[1800:root:3]SSL established: TLSv1 ECDHE-RSA-AES256-SHA
[1800:root:3]req: /remote/saml/logout
[1800:root:3]fsv_rmt_saml_logout_cb:35 got SAML logout request.
[1800:root:3]rmt_web_auth_info_parser_common:504 no session id in auth info
[1800:root:3]rmt_web_access_check:772 access failed, uri=[/remote/logout],ret=4103,
[1800:root:5]SSL state:before SSL initialization (10.0.1.3)
[1800:root:5]SSL state:before SSL initialization (10.0.1.3)
[1800:root:5]no SNI received
[1800:root:5]SSL state:SSLv3/TLS read client hello (10.0.1.3)
[1800:root:5]SSL state:SSLv3/TLS write server hello (10.0.1.3)
[1800:root:5]SSL state:SSLv3/TLS write certificate (10.0.1.3)
[1800:root:5]SSL state:SSLv3/TLS write key exchange (10.0.1.3)
[1800:root:5]SSL state:SSLv3/TLS write server done (10.0.1.3)
[1800:root:5]SSL state:SSLv3/TLS write server done (10.0.1.3)
[1800:root:5]SSL state:SSLv3/TLS read client key exchange (10.0.1.3)
[1800:root:5]SSL state:SSLv3/TLS read change cipher spec (10.0.1.3)
[1800:root:5]SSL state:SSLv3/TLS read finished (10.0.1.3)
[1800:root:5]SSL state:SSLv3/TLS write session ticket (10.0.1.3)
[1800:root:5]SSL state:SSLv3/TLS write change cipher spec (10.0.1.3)
[1800:root:5]SSL state:SSLv3/TLS write finished (10.0.1.3)
```

64bit, first evidence

As expected, a rewriting of the SSL structure follows with a lot of requests for this payload (364 requests).

```
[1800:root:5]req: /remote/hostcheck_validate?enc=0000000005
[1800:root:5]hostcheck validate failed
[1800:root:5]rmt_web_auth_info_parser_common:504 no session id in auth info
[1800:root:5]rmt_web_access_check:772 access failed, uri=[/remote/logout],ret=4103,
[1800:root:5]req: /remote/hostcheck_validate?enc=0040000007
[1800:root:5]hostcheck validate failed
[1800:root:5]rmt_web_auth_info_parser_common:504 no session id in auth info
[1800:root:5]rmt_web_access_check:772 access failed, uri=[/remote/logout],ret=4103,
[1800:root:5]req: /remote/hostcheck_validate?enc=00be00007
[1800:root:5]hostcheck validate failed
[1800:root:5]rmt_web_auth_info_parser_common:504 no session id in auth info
[1800:root:5]rmt_web_access_check:772 access failed, uri=[/remote/logout],ret=4103,
[1800:root:5]req: /remote/hostcheck_validate?enc=007400000f
[1800:root:5]hostcheck validate failed
[1800:root:5]rmt_web_auth_info_parser_common:504 no session id in auth info
[1800:root:5]rmt_web_access_check:772 access failed, uri=[/remote/logout],ret=4103,
[1800:root:5]req: /remote/hostcheck_validate?enc=004500001
[1800:root:5]hostcheck validate failed
[1800:root:5]rmt_web_auth_info_parser_common:504 no session id in auth info
[1800:root:5]rmt_web_access_check:772 access failed, uri=[/remote/logout],ret=4103,
[1800:root:5]req: /remote/hostcheck_validate?enc=009f00000
[1800:root:5]hostcheck validate failed
[1800:root:5]rmt_web_auth_info_parser_common:504 no session id in auth info
[1800:root:5]rmt_web_access_check:772 access failed, uri=[/remote/logout],ret=4103,
[1800:root:5]req: /remote/hostcheck_validate?enc=00da01006
[1800:root:5]hostcheck validate failed
[1800:root:5]rmt_web_auth_info_parser_common:504 no session id in auth info
[1800:root:5]rmt_web_access_check:772 access failed, uri=[/remote/logout],ret=4103,
[1800:root:5]req: /remote/hostcheck_validate?enc=00ff00007
[1800:root:5]hostcheck validate failed
[1800:root:5]rmt_web_auth_info_parser_common:504 no session id in auth info
[1800:root:5]rmt_web_access_check:772 access failed, uri=[/remote/logout],ret=4103,
[1800:root:5]req: /remote/hostcheck_validate?enc=005901006
[1800:root:5]hostcheck validate failed
[1800:root:5]rmt_web_auth_info_parser_common:504 no session id in auth info
[1800:root:5]rmt_web_access_check:772 access failed, uri=[/remote/logout],ret=4103,
[1800:root:5]req: /remote/hostcheck_validate?enc=00f200008
[1800:root:5]hostcheck validate failed
[1800:root:5]rmt_web_auth_info_parser_common:504 no session id in auth info
[1800:root:5]rmt_web_access_check:772 access failed, uri=[/remote/logout],ret=4103,
[1800:root:5]req: /remote/hostcheck_validate?enc=001800006
[1800:root:5]hostcheck validate failed
[1800:root:5]rmt_web_auth_info_parser_common:504 no session id in auth info
[1800:root:5]rmt_web_access_check:772 access failed, uri=[/remote/logout],ret=4103,
[1800:root:5]req: /remote/hostcheck_validate?enc=00a903000
```

64bit: second evidence

Note: I would like to emphasize once again the existence of certain payloads that could, for example, be successful with just 5 requests. Here, the sequence of requests is therefore the point to pay attention to, not the number of requests.

We also noted the presence of the "enc" variable in the request `/remote/hostcheck_validate?enc=value`, which is targeted in this vulnerability. However, this value is truncated. The hexadecimal value displayed corresponds to the value of the seed. The seed value generally starts with "0x00" and must be 8 hexadecimal character long. No trace of payloads has been identified with this approach.

The "enc" variable is encrypted with a keystream generated from a constant, a value provided by the server (the salt, which changes at runtime), and a value provided by the attacker (the seed). As a result, you cannot always "decrypt" the payload, as if the main sslvpnd process had crashed, the salt would have changed.

Conclusion

In addition to the "blue teamers" elements mentioned in the original blog, it is possible to identify the execution of the CVE-2023-27997 in our test environment. From a defensive point of view, the first thing to remember is **to apply the editor's patch**.

If you need to carry out a forensic analysis, in the hope that the integrity of evidence has been preserved, we advise the following:

- Do not rely solely on the application crash of the `/bin/sslvpnd` process;
- Take an interest in the sequencing of the requests used in the logs;
- Question the presence of `/remote/logincheck` and `/remote/hostcheck_validate` requests;
- Investigate the presence of HTTP requests containing the variable "enc" and do not forget the possibility for the attacker to use the POST request;
- Pay attention to the size of "enc";
- Read how this vulnerability works in detail;

- And most importantly, stay proactive in monitoring the news concerning the investigation of this CVE. This leaflet may be incomplete. Additions and adjustments will be made as real cases come to light.

References

- https://www.fortiguard.com/psirt/FG-IR-23-097
- https://www.fortinet.com/blog/psirt-blogs/analysis-of-cve-2023-27997-and-clarifications-on-volt-typhoon-campaign
- https://blog.lexfo.fr/xortigate-cve-2023-27997.html