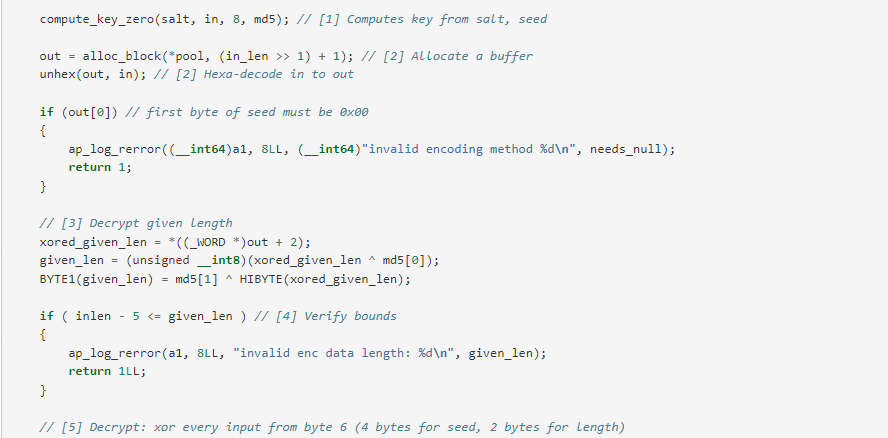
Review on attack to Fortigate VPN (CVE-2023-27997)

# Principle

The attack to Fortigate VPN is based on the bug which is located at comparing the lengths of input buffer and the size field. Using this vulnerability, we can invoke overflow and inject attached exploit payload.

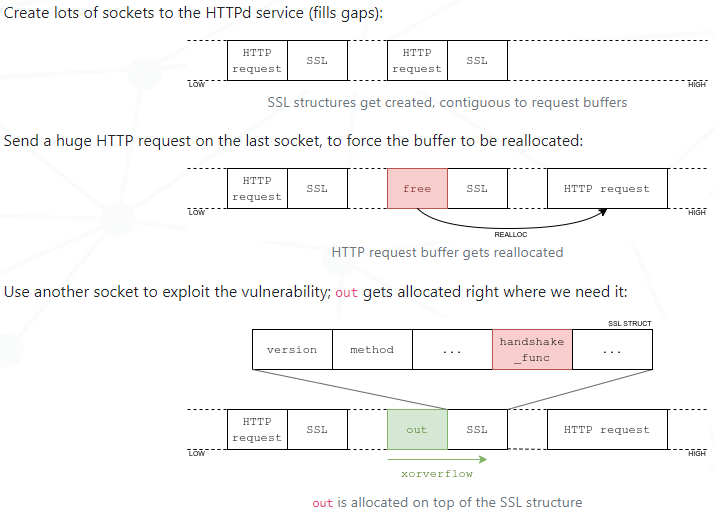


**Here**

So we can attack the vpn with the following flow:

1. Send normal requests
2. Send a huge HTTP request, to force the buffer to be reallocated.
3. Use another socket to exploit the vulnerability.

out gets allocated at the location of old request buffer and SSL structure is modified with the invoked overflow



# 2. Implementation

We wrote Python script to replay attack. Python supports lots of packages so it’s very easy to code.

We implemented our objective in 1 class – Employee. This class has the following functions.

* \_\_init\_\_

This function parses the target and local address. Also parses crash flag.

* run

Acts according to the following scenario:

1. Get salt
2. Send safe payload
3. Connects to server with specified number (NB\_WORKERS) of sockets
4. Send a huge request
5. Send exploit payload

SSL structure is overwritten

1. Check attack result

* \_split\_host\_port

Split address into host and port

* create\_ssl\_socket

Creates SSL socket

* get\_seed\_for\_md5\_byte

Finds a seed which the specified offset (pos) of keystream is equal to value

* single\_write

Writes byte value at the offset pos

* write\_bytes

Write bytes at the offset pos

* compute\_packet\_data\_size

Computes packet data size

* create\_rop\_payload

Creates ROP (Return-Oriented Programming) payload for exploit

* create\_payload

Creates a payload with the specified size using the given seed

To execute, please consult the README.md file in the same directory.

# References

[1] <https://blog.lexfo.fr/xortigate-cve-2023-27997.html>

[2] <https://blog.lexfo.fr/Forensics-xortigate-notice.html>

[3] <https://www.upguard.com/blog/how-to-respond-cve-2023-27997>