Security Evaluation of 'Gate System' by team2

ShinPark Team

0. Introduction

LGE security specialist Team 1 evaluated ‘Gate system’ by team2

Evaluation is done from the security point of view.
The evaluation items are as follows:

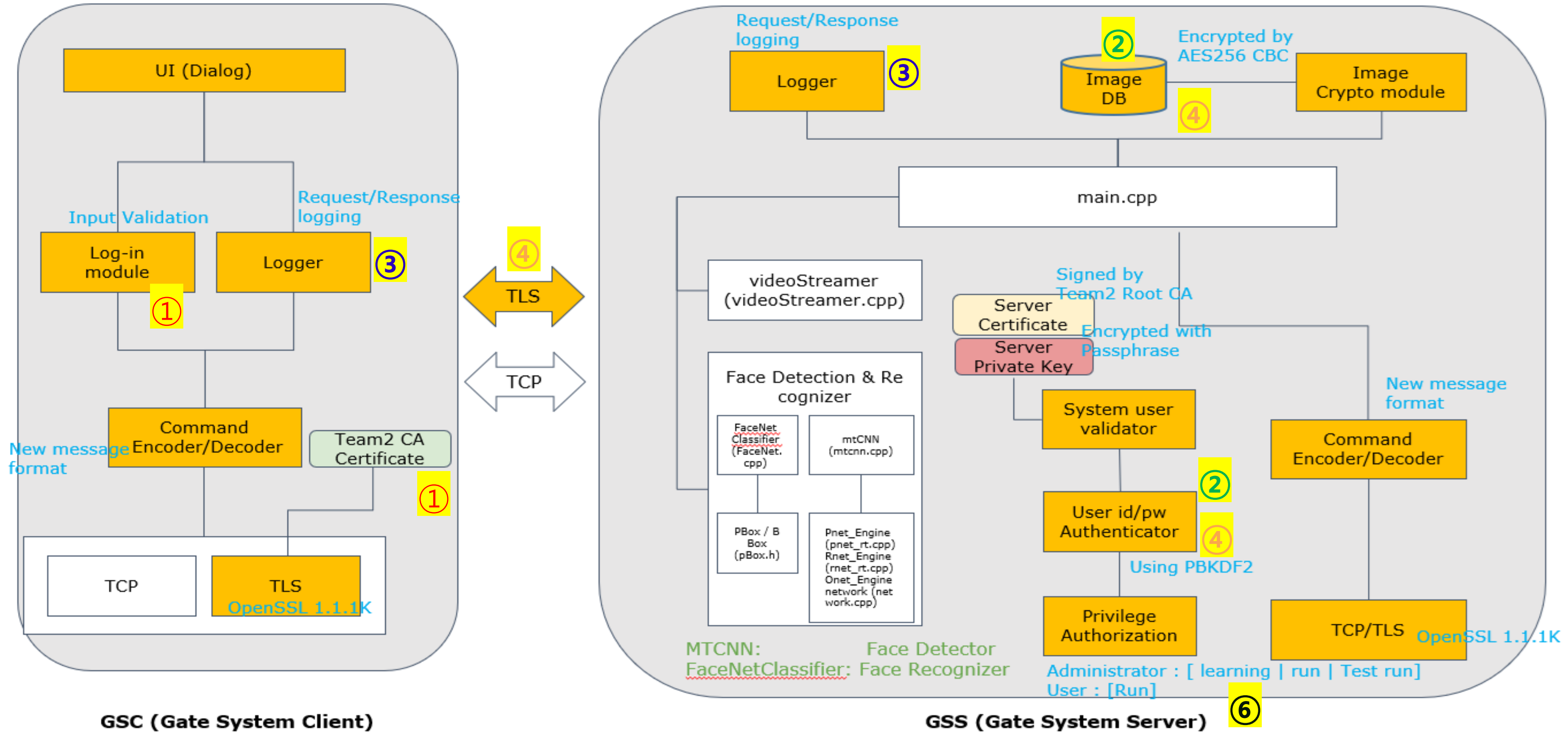
1. Design Analysis
   - ➢ Architecture Review

2. Secure coding
   - ➢ Code Review
   - ➢ Static Analysis
   - ➢ Open-Source Vulnerability Scan

3. Testing
   - ➢ Dynamic Analysis
   - ➢ Fuzz test
   - ➢ Function Test
   - ➢ Penetration test

1. Design Analysis – Architecture Review

Check architecture whether STRIDE is considered

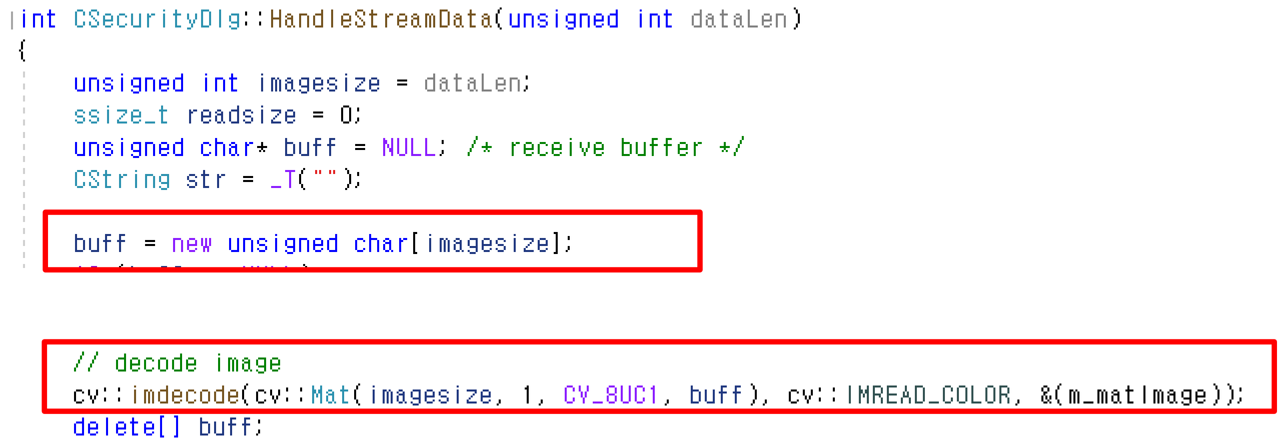


① **Spoofing** - Authenticate via TLS certificates between GSC and GSS
② **Tampering** - Encrypt User information and Image DB
③ **Repudiation** - Logging operation for non-repudiation
④ **Information disclosure** - Encrypt Sensitive data and communicate via TLS
⑤ **Denial of Service** – Could not find the design (ex. Firewall, IDPS, Service manager, log rotation, etc.)
⑥ **Elevation of Privilege** - Permission control by ID

2. Secure Coding – Code Review

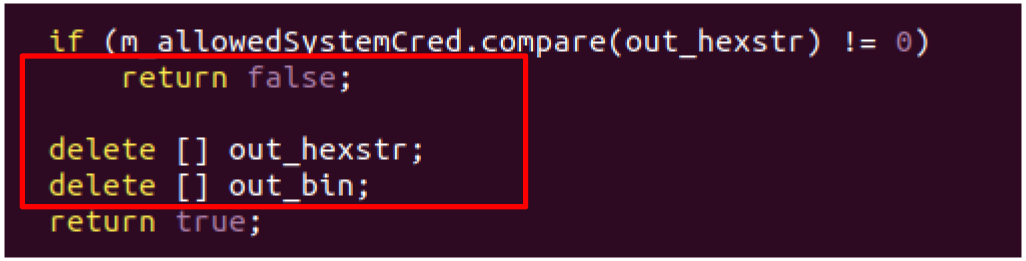
Eye inspection for implementation vulnerabilities in source code

Ex #1. Insufficient size check : A crash occurs when *dataLen* is 0



```
int CSecurityDlg::HandleStreamData(unsigned int dataLen)
{
    unsigned int imagesize = dataLen;
    ssize_t readsize = 0;
    unsigned char* buff = NULL; /* receive buffer */
    CString str = _T("");

    buff = new unsigned char[imagesize];


    // decode image
    cv::imdecode(cv::Mat(imagesize, 1, CV_8UC1, buff), cv::IMREAD_COLOR, &(m_matImage));
    delete[] buff;
```

Ex #2. Memory leak : Missing memory release when if status is *false.*
It doesn't matter as the program ends immediately, but it can cause problems afterwards.



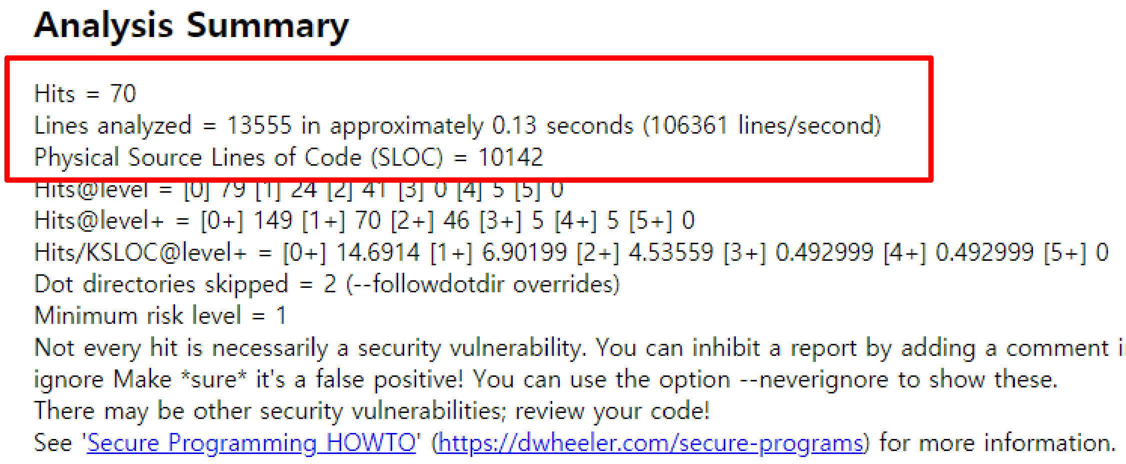
```
if (m_allowedSystemCred.compare(out_hexstr) != 0)
    return false;

delete [] out_hexstr;
delete [] out_bin;
return true;
```

2. Secure Coding – Static Analysis (flaw finder)

Static Analysis with ‘flaw finder’.

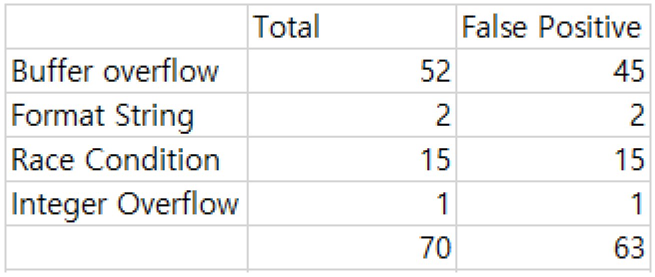
flawfinder_output.html

- ✓ Can check the CWE-based secure coding guide.
- ✓ Out of 70 issues, 7 issues are meaningful flaws, 63 issues are considered as ‘False Positive’.



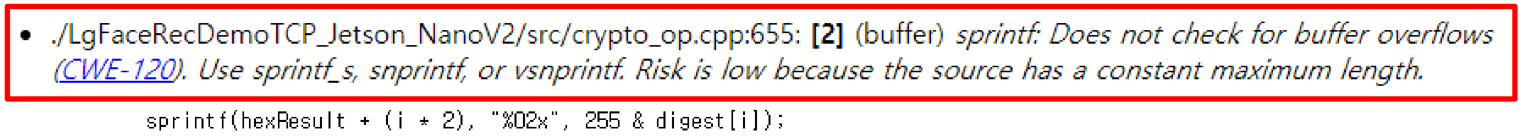
Analysis Summary

```
Hits = 70
Lines analyzed = 13555 in approximately 0.13 seconds (106361 lines/second)
Physical Source Lines of Code (SLOC) = 10142
Hits@level = [0] 79 [1] 24 [2] 41 [3] 0 [4] 5 [5] 0
Hits@level+ = [0+] 149 [1+] 70 [2+] 46 [3+] 5 [4+] 5 [5+] 0
Hits/KSLOC@level+ = [0+] 14.6914 [1+] 6.90199 [2+] 4.53559 [3+] 0.492999 [4+] 0.492999 [5+] 0
Dot directories skipped = 2 (--followdotdir overrides)
Minimum risk level = 1
Not every hit is necessarily a security vulnerability. You can inhibit a report by adding a comment in this form: // flawfinder:
ignore Make *sure* it's a false positive! You can use the option --neverignore to show these.
There may be other security vulnerabilities; review your code!
See 'Secure Programming HOWTO' (https://dwheeler.com/secure-programs) for more information.
```



	Total	False Positive
Buffer overflow	52	45
Format String	2	2
Race Condition	15	15
Integer Overflow	1	1
	70	63

Ex #1. Does not check for buffer overflows with ‘sprintf’.



- ./LgFaceRecDemoTCP_Jetson_NanoV2/src/crypto_op.cpp:655: **[2]** (buffer) *sprintf: Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or vsnprintf. Risk is low because the source has a constant maximum length.*

```
sprintf(hexResult + (i * 2), "%02x", 255 & digest[i]);
```

LG Electronics

Carnegie Mellon University

2. Secure Coding – Static Analysis (SonarQube)

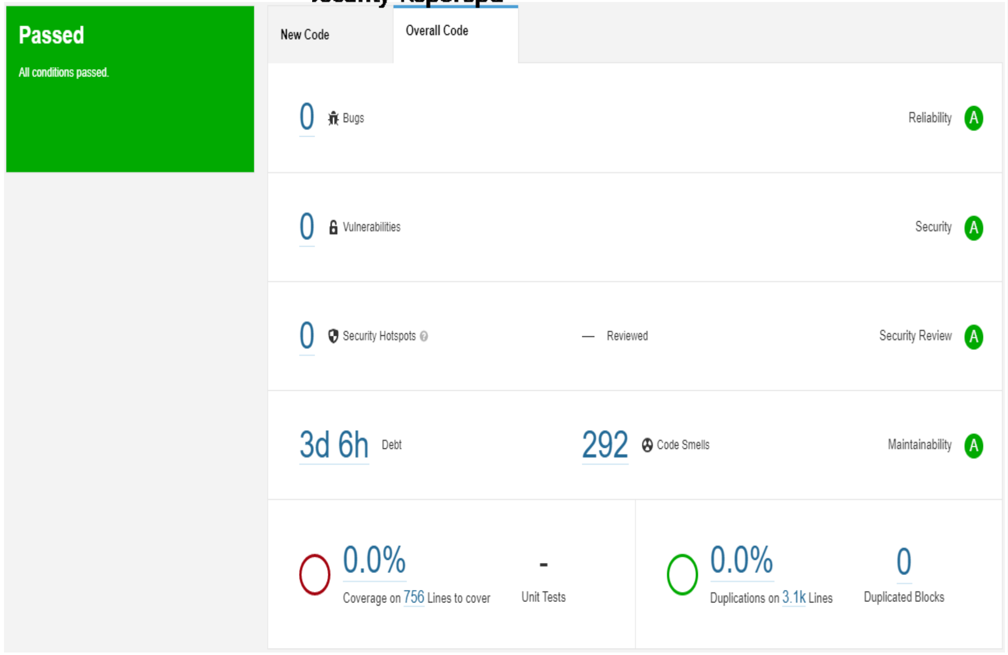
Static Analysis with 'SonarQube'.

- ✓ No issue detected by Client.
- ✓ 1 Bugs, 2 Vulnerabilities, 14 Security Hotspots detected by Server.
- ✓ Can check various standards-based vulnerabilities(CERT, OWASP, Misra C++, etc).
- ✓ Can obtain compliance solutions.

	Total	false positive
Bug	1	1
vulnerabilities	2	0
Security Hospots	14	14

<Client>

team2_test2 - Security Report.pd



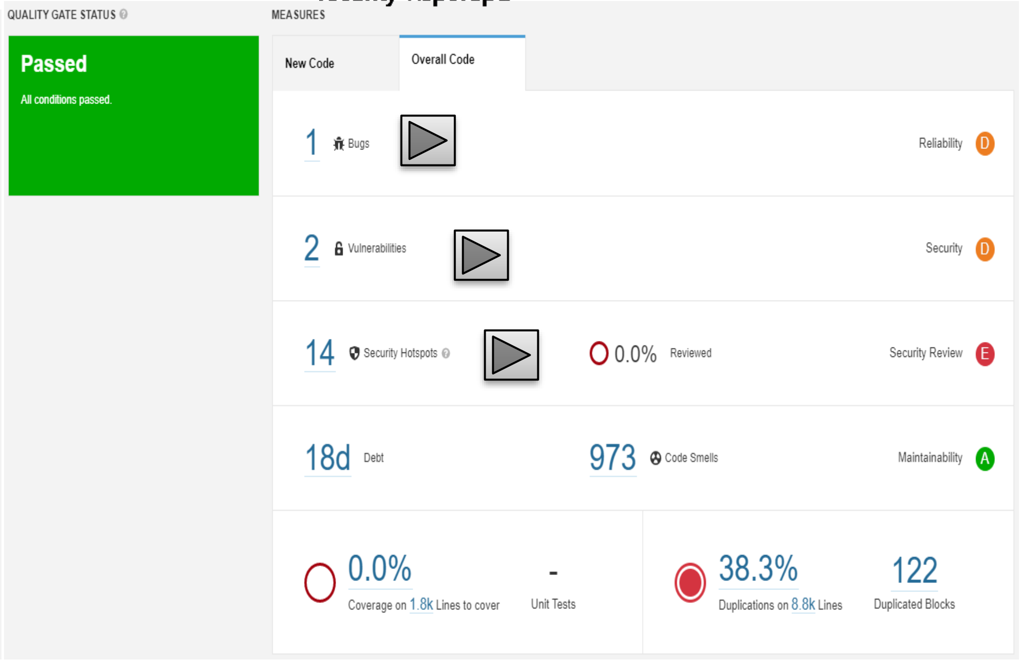
Passed
All conditions passed.

New Code | Overall Code

0 Bugs — Reliability A
0 Vulnerabilities — Security A
0 Security Hotspots — — Reviewed — Security Review A
3d 6h Debt — 292 Code Smells — Maintainability A
0.0% Coverage on 756 Lines to cover — - Unit Tests
0.0% Duplications on 3.1k Lines — 0 Duplicated Blocks

<Server>

team2-server - Security Report.pd



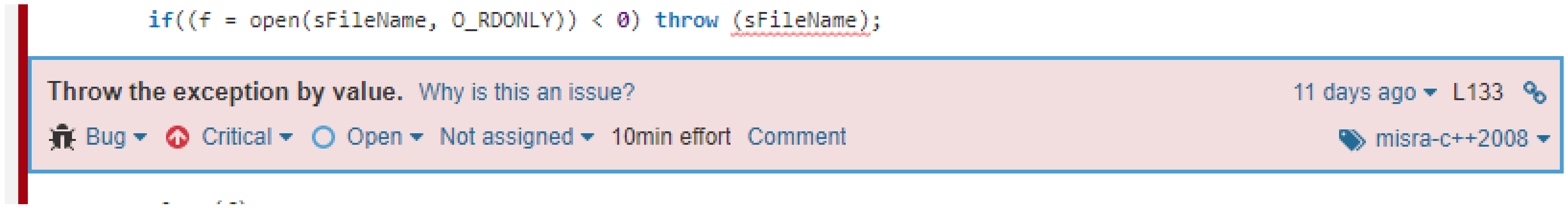
QUALITY GATE STATUS
Passed
All conditions passed.

MEASURES
New Code | Overall Code

1 Bugs — Reliability D
2 Vulnerabilities — Security D
14 Security Hotspots — 0.0% Reviewed — Security Review E
18d Debt — 973 Code Smells — Maintainability A
0.0% Coverage on 1.8k Lines to cover — - Unit Tests
38.3% Duplications on 8.8k Lines — 122 Duplicated Blocks

2. Secure Coding – Static Analysis (SonarQube)

Ex #1. Bug



```
if((f = open(sFileName, O_RDONLY)) < 0) throw (sFileName);
```

Throw the exception by value. Why is this an issue?

Bug · Critical · Open · Not assigned · 10min effort · Comment

11 days ago · L133 · misra-c++2008

If a pointer to an object is used as an exception, the code that will catch the exception may or may not have to delete the pointed-to object. This is even more complex in the exception case than in classical manual memory management, because of the distance between the `throw` statements and the matching `catch`.

Throwing by value is just simpler and less error prone.

Compliant Solution

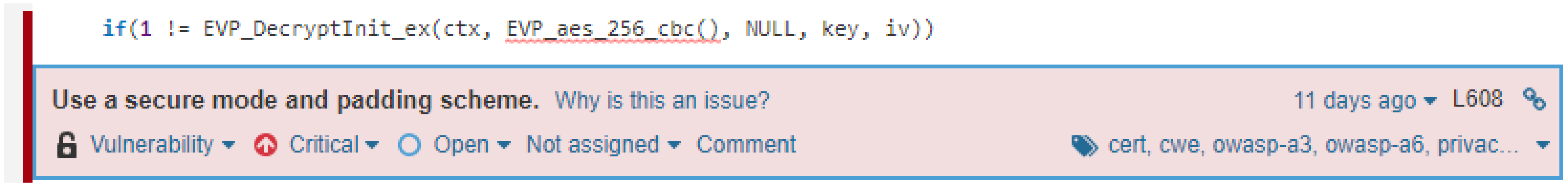
```
class E { /* Implementation */};
E globalException;

void fn ( int i )
{
  if ( i > 10 ) {
    throw ( globalException); // Throws a copy of the global variable
  }
  else {
    throw (E{} ); // Throws a new object
  }
}
```

LG Electronics

Carnegie Mellon University

2. Secure Coding – Static Analysis (SonarQube)

Ex #2. Vulnerability



```
    if(1 != EVP_DecryptInit_ex(ctx, EVP_aes_256_cbc(), NULL, key, iv))
```

Use a secure mode and padding scheme. Why is this an issue? 11 days ago L608

Encryption operation mode and the padding scheme should be chosen appropriately to guarantee data confidentiality, integrity and authenticity:

- For block cipher encryption algorithms (like AES):

the GCM (Galois Counter Mode) mode which works internally with zero/no padding scheme, is recommended, as it is designed to provide both data authenticity (integrity) and confidentiality. Other similar modes are CCM, CWC, EAX, IAPM and OCB.

the CBC (Cipher Block Chaining) mode by itself provides only data confidentiality, it's recommended to use it along with Message Authentication Code or similar to achieve data authenticity (integrity) too and thus to prevent padding oracle attacks.

the ECB (Electronic Codebook) mode doesn't provide serious message confidentiality: under a given key any given plaintext block always gets encrypted to the same ciphertext block. This mode should not be used.

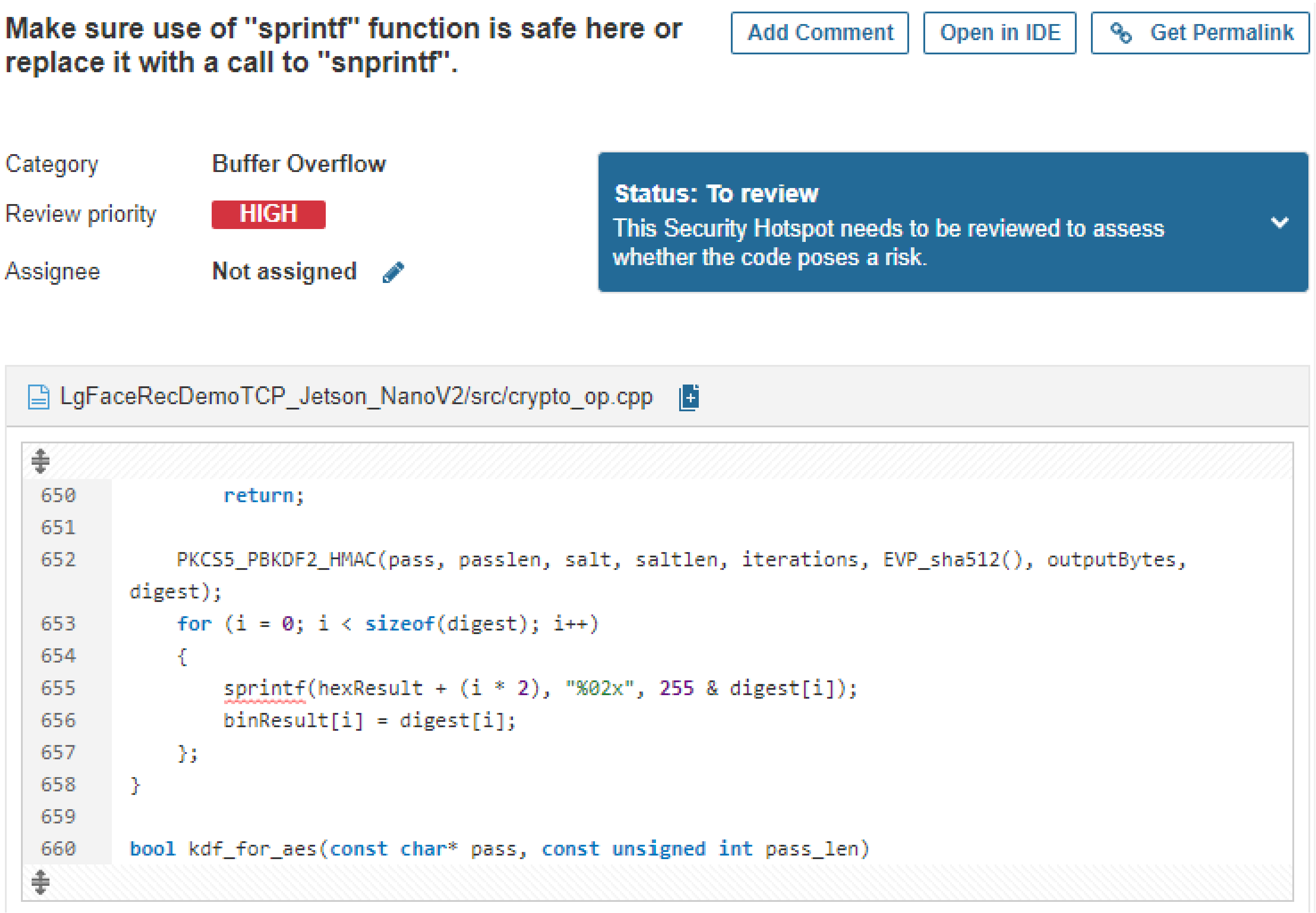
- For RSA encryption algorithm, the recommended padding scheme is OAEP.

OpenSSL

```
#include <openssl/evp.h>

// AES symmetric cipher is recommended to be used with GCM mode
EVP_aes_128_gcm() // Compliant

// RSA asymmetric cipher is recommended be used with OAEP padding
RSA_public_decrypt(flen, from, to, key, RSA_PKCS1_OAEP_PADDING); // Compliant
```

2. Secure Coding – Static Analysis (SonarQube)

Ex #3. Security Hotspots



Make sure use of "sprintf" function is safe here or replace it with a call to "snprintf".

Category	Buffer Overflow
Review priority	HIGH
Assignee	Not assigned

Status: To review
This Security Hotspot needs to be reviewed to assess whether the code poses a risk.

LgFaceRecDemoTCP_Jetson_NanoV2/src/crypto_op.cpp

```
        return;

    PKCS5_PBKDF2_HMAC(pass, passlen, salt, saltlen, iterations, EVP_sha512(), outputBytes,
digest);
    for (i = 0; i < sizeof(digest); i++)
    {
        sprintf(hexResult + (i * 2), "%02x", 255 & digest[i]);
        binResult[i] = digest[i];
    };
}

bool kdf_for_aes(const char* pass, const unsigned int pass_len)
```

2. Secure Coding – OSSVS

‘Gate System’ use some of open source as follows:

- ✓ nvidia cuda (common)
- ✓ tensorrt (common)
- ✓ **OpenSSL (specified)**
- ✓ Opencv (common)

Open Source Security Vulnerability Scan performed with used openSSL version. (except for common open source)

Ex #1. OpenSSL

- ✓ **https://www.openssl.org/news/vulnerabilities-1.1.1.html**
- ✓ **Gate system uses OepnSSL 1.1.1k version (latest version)**
- ✓ **Major changes between OpenSSL 1.1.1j and OpenSSL 1.1.1k**
  - Fixed a problem with verifying a certificate chain when using the X509_V_FLAG_X509_STRICT flag ([CVE-2021-3450])
  - Fixed an issue where an OpenSSL TLS server may crash if sent a m renegotiation ClientHello message from a client ([CVE-2021-3449])

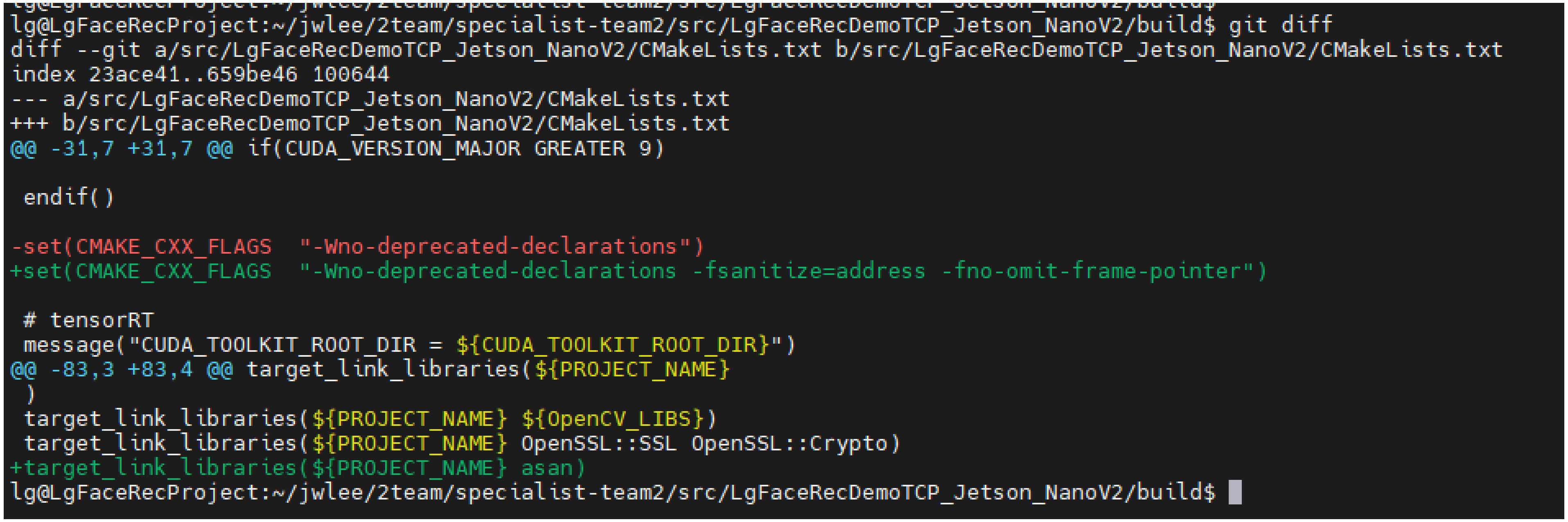
NO PROBLEM

3. Test – Dynamic Analysis

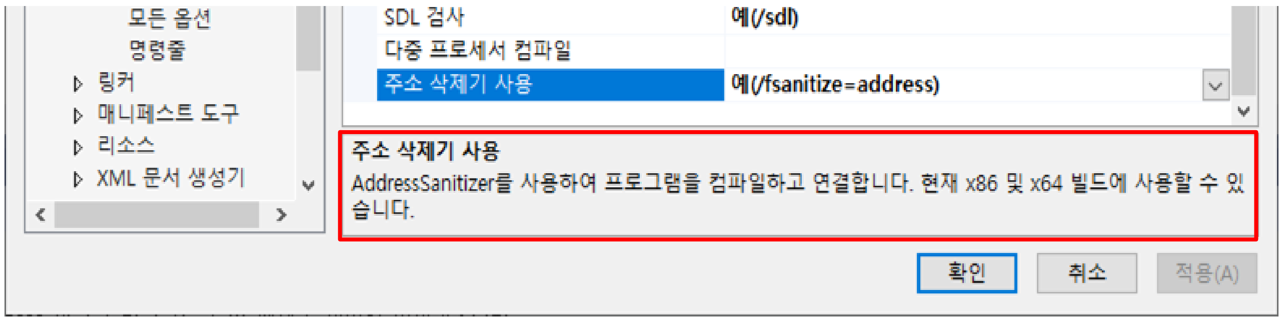
Dynamic Analysis with ‘ASan’.

✓ ‘Address Sanitizer’(ASan) can detect memory bugs

✓ No significant memory-caused crashes during nomal testing on server/client



```
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_NanoV2/build$ git diff
diff --git a/src/LgFaceRecDemoTCP_Jetson_NanoV2/CMakeLists.txt b/src/LgFaceRecDemoTCP_Jetson_NanoV2/CMakeLists.txt
index 23ace41..659be46 100644
--- a/src/LgFaceRecDemoTCP_Jetson_NanoV2/CMakeLists.txt
+++ b/src/LgFaceRecDemoTCP_Jetson_NanoV2/CMakeLists.txt
@@ -31,7 +31,7 @@ if(CUDA_VERSION_MAJOR GREATER 9)

 endif()

-set(CMAKE_CXX_FLAGS  "-Wno-deprecated-declarations")
+set(CMAKE_CXX_FLAGS  "-Wno-deprecated-declarations -fsanitize=address -fno-omit-frame-pointer")

 # tensorRT
 message("CUDA_TOOLKIT_ROOT_DIR = ${CUDA_TOOLKIT_ROOT_DIR}")
@@ -83,3 +83,4 @@ target_link_libraries(${PROJECT_NAME}
 )
 target_link_libraries(${PROJECT_NAME} ${OpenCV_LIBS})
 target_link_libraries(${PROJECT_NAME} OpenSSL::SSL OpenSSL::Crypto)
+target_link_libraries(${PROJECT_NAME} asan)
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_NanoV2/build$
```



SDL 검사	예(/sdl)
다중 프로세서 컴파일	
주소 삭제기 사용	예(/fsanitize=address)

주소 삭제기 사용
AddressSanitizer를 사용하여 프로그램을 컴파일하고 연결합니다. 현재 x86 및 x64 빌드에 사용할 수 있습니다.

3. Test – Fuzz Test (AFL)

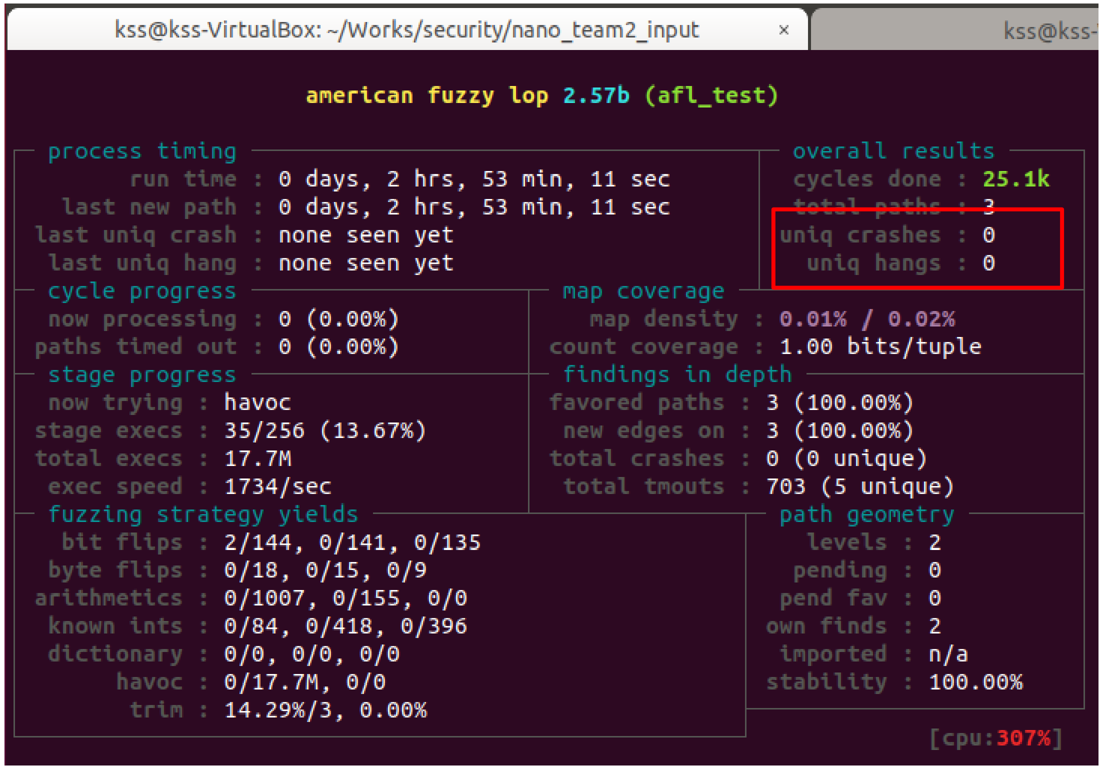
Fuzz testing with 'AFL'.

✓ Input validation is critical for security, so we performed AFL.

✓ The Server receives the port number and secure mode of operation from user.

```
portNum = atoi(argv[1]);
if(!strcmp(argv[2], "0")) bSecureMode = false;
```

✓ No Crashes, No hangs. But…



```
kss@kss-VirtualBox: ~/Works/security/nano_team2_input        ×        kss@kss-

                    american fuzzy lop 2.57b (afl_test)

┌─ process timing ─────────────────────────────────────┬─ overall results ─────┐
│        run time : 0 days, 2 hrs, 53 min, 11 sec      │  cycles done : 25.1k  │
│   last new path : 0 days, 2 hrs, 53 min, 11 sec      │  total paths : 3      │
│ last uniq crash : none seen yet                      │ uniq crashes : 0      │
│  last uniq hang : none seen yet                      │   uniq hangs : 0      │
├─ cycle progress ────────────────────┬─ map coverage ─┴───────────────────────┤
│  now processing : 0 (0.00%)         │    map density : 0.01% / 0.02%        │
│ paths timed out : 0 (0.00%)         │ count coverage : 1.00 bits/tuple      │
├─ stage progress ────────────────────┼─ findings in depth ────────────────────┤
│  now trying : havoc                 │ favored paths : 3 (100.00%)           │
│ stage execs : 35/256 (13.67%)       │  new edges on : 3 (100.00%)           │
│ total execs : 17.7M                 │ total crashes : 0 (0 unique)          │
│  exec speed : 1734/sec              │  total tmouts : 703 (5 unique)        │
├─ fuzzing strategy yields ───────────┴───────────────┬─ path geometry ────────┤
│   bit flips : 2/144, 0/141, 0/135                   │    levels : 2          │
│  byte flips : 0/18, 0/15, 0/9                       │   pending : 0          │
│ arithmetics : 0/1007, 0/155, 0/0                    │  pend fav : 0          │
│  known ints : 0/84, 0/418, 0/396                    │ own finds : 2          │
│  dictionary : 0/0, 0/0, 0/0                         │  imported : n/a        │
│       havoc : 0/17.7M, 0/0                          │ stability : 100.00%    │
│        trim : 14.29%/3, 0.00%                       ├────────────────────────┘
└─────────────────────────────────────────────────────┘          [cpu:307%]
```

SEI CERT C Coding Standard

Use *strtol()* instead of *atoi()*

ERR34-C. Detect errors when converting a string to a number

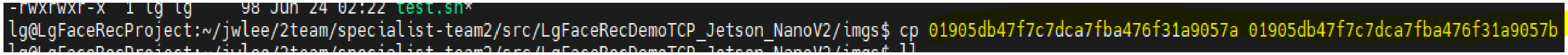
Use one of the C Standard Library strto*() functions to parse an integer or floating-point number from a string. These functions provide more robust error handling than alternative solutions.

3. Test – Fuzz Test (Manual Fuzz)

Fuzz testing with ‘Tampering’.

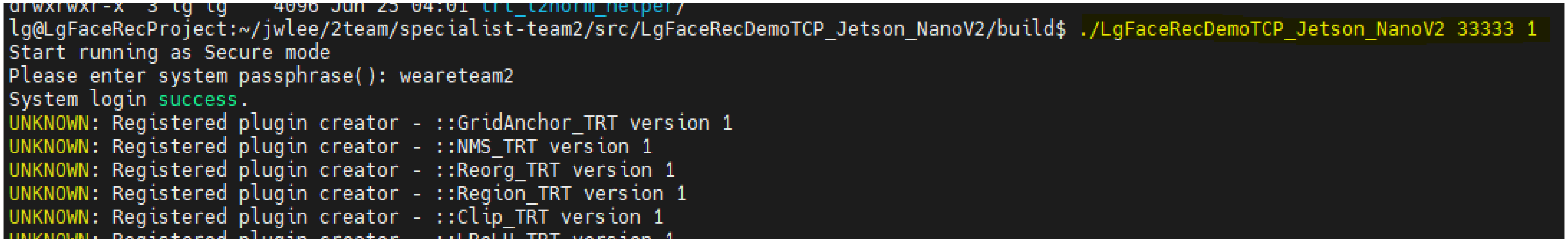
- ✓ The tampered data caused not only errors but also memory leaks.
- ✓ Error handling have to consider Memory leaks.

<just Modify Image file name>



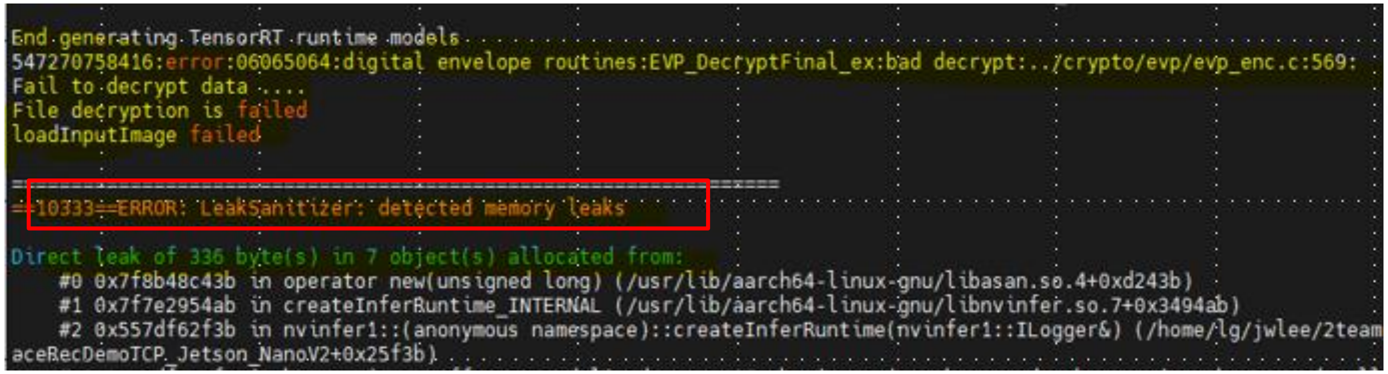
```
-rwxrwxr-x  1 lg lg     98 Jun 24 02:22 test.sh*
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_NanoV2/imgs$ cp 01905db47f7c7dca7fba476f31a9057a 01905db47f7c7dca7fba476f31a9057b
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_NanoV2/imgs$ ll
```

<Run Server>



```
drwxrwxr-x  3 lg lg   4096 Jun 25 04:01 trt_l2norm_helper/
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_NanoV2/build$ ./LgFaceRecDemoTCP_Jetson_NanoV2 33333 1
Start running as Secure mode
Please enter system passphrase(): weareteam2
System login success.
UNKNOWN: Registered plugin creator - ::GridAnchor_TRT version 1
UNKNOWN: Registered plugin creator - ::NMS_TRT version 1
UNKNOWN: Registered plugin creator - ::Reorg_TRT version 1
UNKNOWN: Registered plugin creator - ::Region_TRT version 1
UNKNOWN: Registered plugin creator - ::Clip_TRT version 1
UNKNOWN: Registered plugin creator - ::LReLU_TRT version 1
```

<Result>



```
End generating TensorRT runtime models
547270758416:error:06065064:digital envelope routines:EVP_DecryptFinal_ex:bad decrypt:../crypto/evp/evp_enc.c:569:
Fail to decrypt data ....
File decryption is failed
loadInputImage failed

=================================================================
==10333==ERROR: LeakSanitizer: detected memory leaks

Direct leak of 336 byte(s) in 7 object(s) allocated from:
    #0 0x7f8b48c43b in operator new(unsigned long) (/usr/lib/aarch64-linux-gnu/libasan.so.4+0xd243b)
    #1 0x7f7e2954ab in createInferRuntime_INTERNAL (/usr/lib/aarch64-linux-gnu/libnvinfer.so.7+0x3494ab)
    #2 0x557df62f3b in nvinfer1::(anonymous namespace)::createInferRuntime(nvinfer1::ILogger&) (/home/lg/jwlee/2team
aceRecDemoTCP_Jetson_NanoV2+0x25f3b)
```

1. service dead due to decryption fail

2. service doesn`t free memory when exception occurred

3. Test – Forensic Test

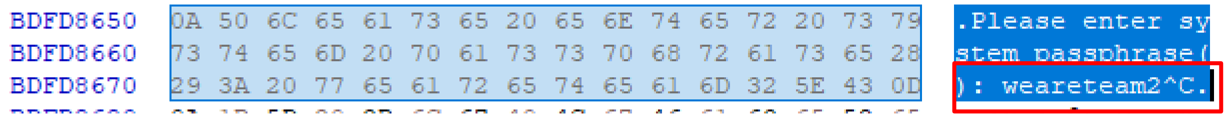
Full Memory Dump in Server

- ✓ Full physical memory dump and analysis
- ✓ Found some credential data in memory (passphrase, user name)

Memory dump using LiME(https://github.com/504ensicsLabs/LiME)

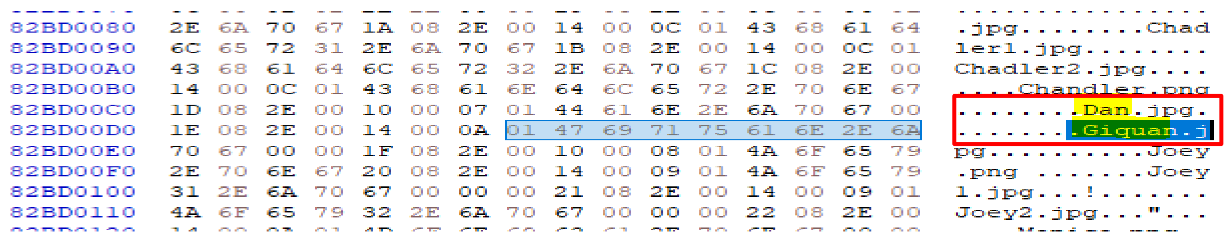
```
sudo insmod ./4.9.201-tegra/updates/dkms/lime.ko "path=/home/lg/jwlee/mem.lime format=lime“

lg@LgFaceRecProject:~/jwlee$ cat /proc/meminfo | grep MemTotal
MemTotal:       4059272 kB     (<- 4GB memory)

lg@LgFaceRecProject:~/jwlee$ ll mem.lime
-r--r--r-- 1 root root 4259315808 Jun 29 05:10 mem.lime     (<- full dumped)
```

Analysis dumped file using HxD(HxD - Freeware Hex Editor and Disk Editor | mh-nexus)



```
BDFD8650  0A 50 6C 65 61 73 65 20 65 6E 74 65 72 20 73 79  .Please enter sy
BDFD8660  73 74 65 6D 20 70 61 73 73 70 68 72 61 73 65 28  stem passphrase(
BDFD8670  29 3A 20 77 65 61 72 65 74 65 61 6D 32 5E 43 0D  ): weareteam2^C.
```

passphrase -> ‘weareteam2’



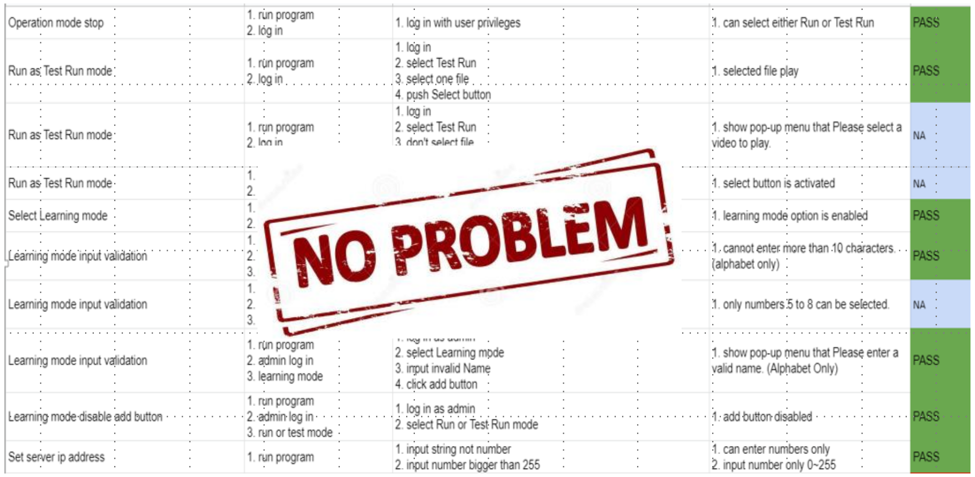
```
82BD0080  2E 6A 70 67 1A 08 2E 00 14 00 0C 01 43 68 61 64  .jpg........Chad
82BD0090  6C 65 72 31 2E 6A 70 67 1B 08 2E 00 14 00 0C 01  ler1.jpg........
82BD00A0  43 68 61 64 6C 65 72 32 2E 6A 70 67 1C 08 2E 00  Chadler2.jpg....
82BD00B0  14 00 0C 01 43 68 61 6E 64 6C 65 72 2E 70 6E 67  ....Chandler.png
82BD00C0  1D 08 2E 00 10 00 07 01 44 61 6E 2E 6A 70 67 00  ........Dan.jpg.
82BD00D0  1E 08 2E 00 14 00 0A 01 47 69 71 75 61 6E 2E 6A  ........Giquan.j
82BD00E0  70 67 00 00 1F 08 2E 00 10 00 08 01 4A 6F 65 79  pg..........Joey
82BD00F0  2E 70 6E 67 20 08 2E 00 14 00 09 01 4A 6F 65 79  .png .......Joey
82BD0100  31 2E 6A 70 67 00 00 00 21 08 2E 00 14 00 09 01  1.jpg...!.......
82BD0110  4A 6F 65 79 32 2E 6A 70 67 00 00 00 22 08 2E 00  Joey2.jpg...."...
```

name as plain text -> ‘Dan’, ‘Giguan’, etc

3. Test – Test Case

Testing with ‘Test Case’.

✓ Original test case by developers was passed all.



Operation mode stop	1. run program 2. log in	1. log in with user privileges	1. can select either Run or Test Run	PASS
Run as Test Run mode	1. run program 2. log in	1. log in 2. select Test Run 3. select one file 4. push Select button	1. selected file play	PASS
Run as Test Run mode	1. run program 2. log in	1. log in 2. select Test Run 3. don't select file	1. show pop-up menu that Please select a video to play.	NA
Run as Test Run mode	1. 2.	[illegible]	1. select button is activated	NA
Select Learning mode	1. 2.	[illegible]	1. learning mode option is enabled	PASS
Learning mode input validation	1. 2. 3.	[illegible]	1. cannot enter more than 10 characters. (alphabet only)	PASS
Learning mode input validation	1. 2. 3.	[illegible]	1. only numbers 5 to 8 can be selected.	NA
Learning mode input validation	1. run program 2. admin log in 3. learning mode	[illegible] 2. select Learning mode 3. input invalid Name 4. click add button	1. show pop-up menu that Please enter a valid name. (Alphabet Only)	PASS
Learning mode disable add button	1. run program 2. admin log in 3. run or test mode	1. log in as admin 2. select Run or Test Run mode	1. add button disabled	PASS
Set server ip address	1. run program	1. input string not number 2. input number bigger than 255	1. can enter numbers only 2. input number only 0~255	PASS

✓ Some of Test Case are added to meet the security requirements as follows:

Requirements

- Learning Mode - User images can be added to the image database. In this mode the interface should query for the name of the person in front of the camera and the number of samples to be collected.
- Proper fault/error detection, recovery, and reporting.

No	Test case	Result	Description
24	In Learning Mode the interface should query for the name of the person in front of the camera and the number of samples to be collected.	Fail	The interface does not query the number of samples to be collected. **→ Failed to meet system requirements**
25	If the server is forcibly terminated, it should restart again.	Fail	If the server is forcibly terminated with 'sudo pkill' command, it was not restarted. **→ Insufficient system resiliency / robustness**

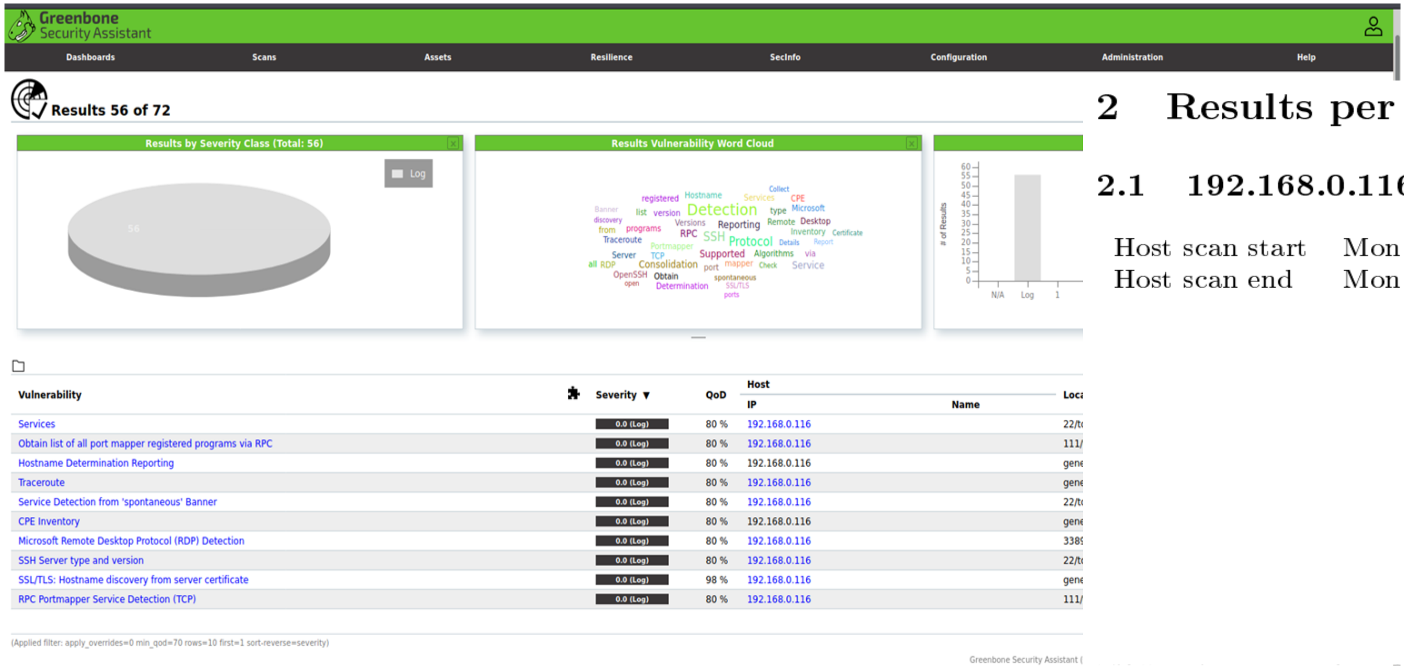
3. Test – OpenVAS

Open Vulnerability Assessment Scan performed with 'OpenVAS '.

report-openvas.pdf

- ✓ OpenVAS can check network vulnerability.
- ✓ Detected 2 medium level vulnerabilities

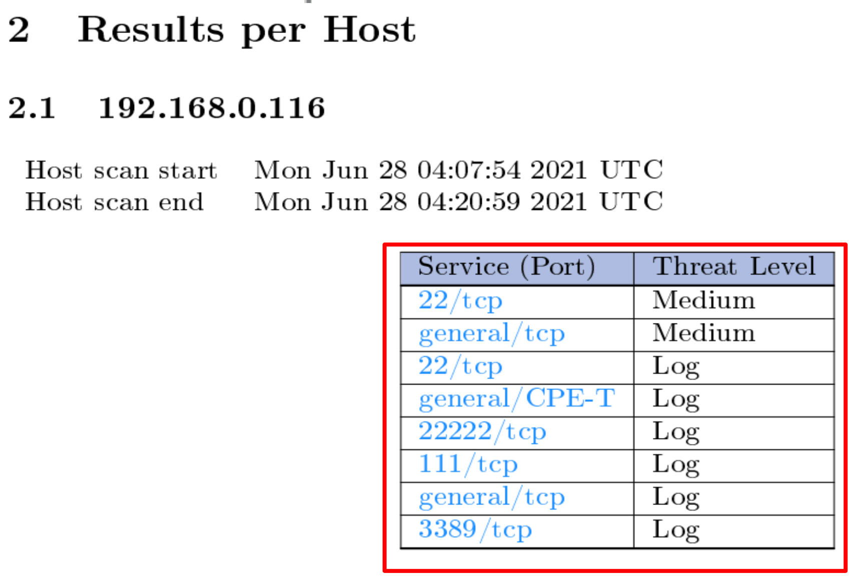
Service(port)	Subject	Vulnerability insight	CVE
SSH	OpenSSH <= 8.3p1 Command Injection Vulnerability	scp of OpenSSH allows command injection in spc.c via backtick characters in the destination argument.	CVE-2020-15778
General TCP	TCP Sequence Number Approximation Reset Denial of Service Vulnerability	The flaw is triggered when spoofed TCP Reset packets are received by the targeted TCP stack and will result in loss of availability for the attacked TCP services.	CVE-2004-0230



Vulnerability	Severity ▼	QoD	Host IP	Name	Loca
Services	0.0 (Log)	80 %	192.168.0.116		22/t
Obtain list of all port mapper registered programs via RPC	0.0 (Log)	80 %	192.168.0.116		111/
Hostname Determination Reporting	0.0 (Log)	80 %	192.168.0.116		gene
Traceroute	0.0 (Log)	80 %	192.168.0.116		gene
Service Detection from 'spontaneous' Banner	0.0 (Log)	80 %	192.168.0.116		22/t
CPE Inventory	0.0 (Log)	80 %	192.168.0.116		gene
Microsoft Remote Desktop Protocol (RDP) Detection	0.0 (Log)	80 %	192.168.0.116		3389
SSH Server type and version	0.0 (Log)	80 %	192.168.0.116		22/t
SSL/TLS: Hostname discovery from server certificate	0.0 (Log)	98 %	192.168.0.116		gene
RPC Portmapper Service Detection (TCP)	0.0 (Log)	80 %	192.168.0.116		111/

(Applied filter: apply_overrides=0 min_qod=70 rows=10 first=1 sort-reverse=severity)

Greenbone Security Assistant (



2 Results per Host

2.1 192.168.0.116

Host scan start	Mon Jun 28 04:07:54 2021 UTC
Host scan end	Mon Jun 28 04:20:59 2021 UTC

Service (Port)	Threat Level
22/tcp	Medium
general/tcp	Medium
22/tcp	Log
general/CPE-T	Log
22222/tcp	Log
111/tcp	Log
general/tcp	Log
3389/tcp	Log

3. Test – Penetration Test

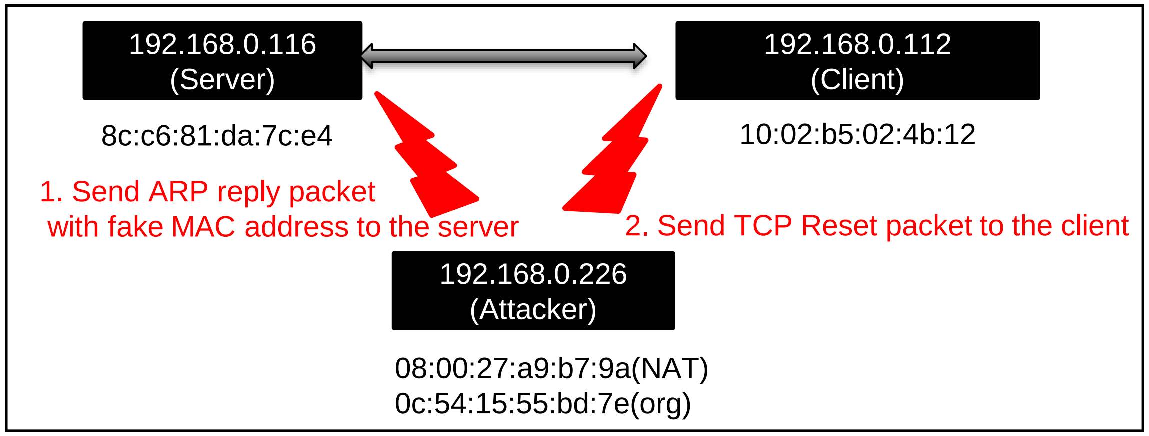
Penetration testing with 'Dos Attack'.

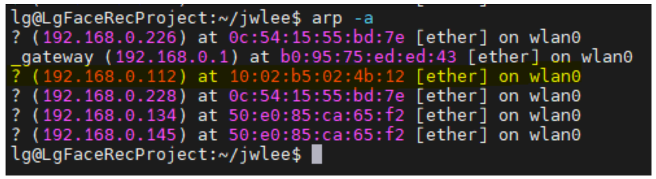
Service(port)	Subject	Vulnerability insight	CVE
General TCP	TCP Sequence Number Approximation Reset Denial of Service Vulnerability	The flaw is triggered when spoofed TCP Reset packets are received by the targeted TCP stack and will result in loss of availability for the attacked TCP services.	CVE-2004-0230

✔ Attempted ARP(Address Resolution Protocol) spoofing and Send TCP Reset packet to Client

➔ Client can`t receive Server`s data(face img,etc…) and closed TCP connection by attacker

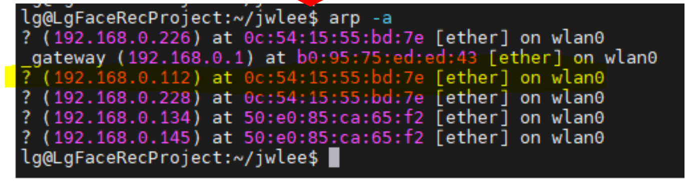
✔ To avoid MITM (Man In The Middle Attack), firewall/IDS have to be considered.(detect spoofing)





```
lg@LgFaceRecProject:~/jwlee$ arp -a
? (192.168.0.226) at 0c:54:15:55:bd:7e [ether] on wlan0
_gateway (192.168.0.1) at b0:95:75:ed:ed:43 [ether] on wlan0
? (192.168.0.112) at 10:02:b5:02:4b:12 [ether] on wlan0
? (192.168.0.228) at 0c:54:15:55:bd:7e [ether] on wlan0
? (192.168.0.134) at 50:e0:85:ca:65:f2 [ether] on wlan0
? (192.168.0.145) at 50:e0:85:ca:65:f2 [ether] on wlan0
lg@LgFaceRecProject:~/jwlee$
```

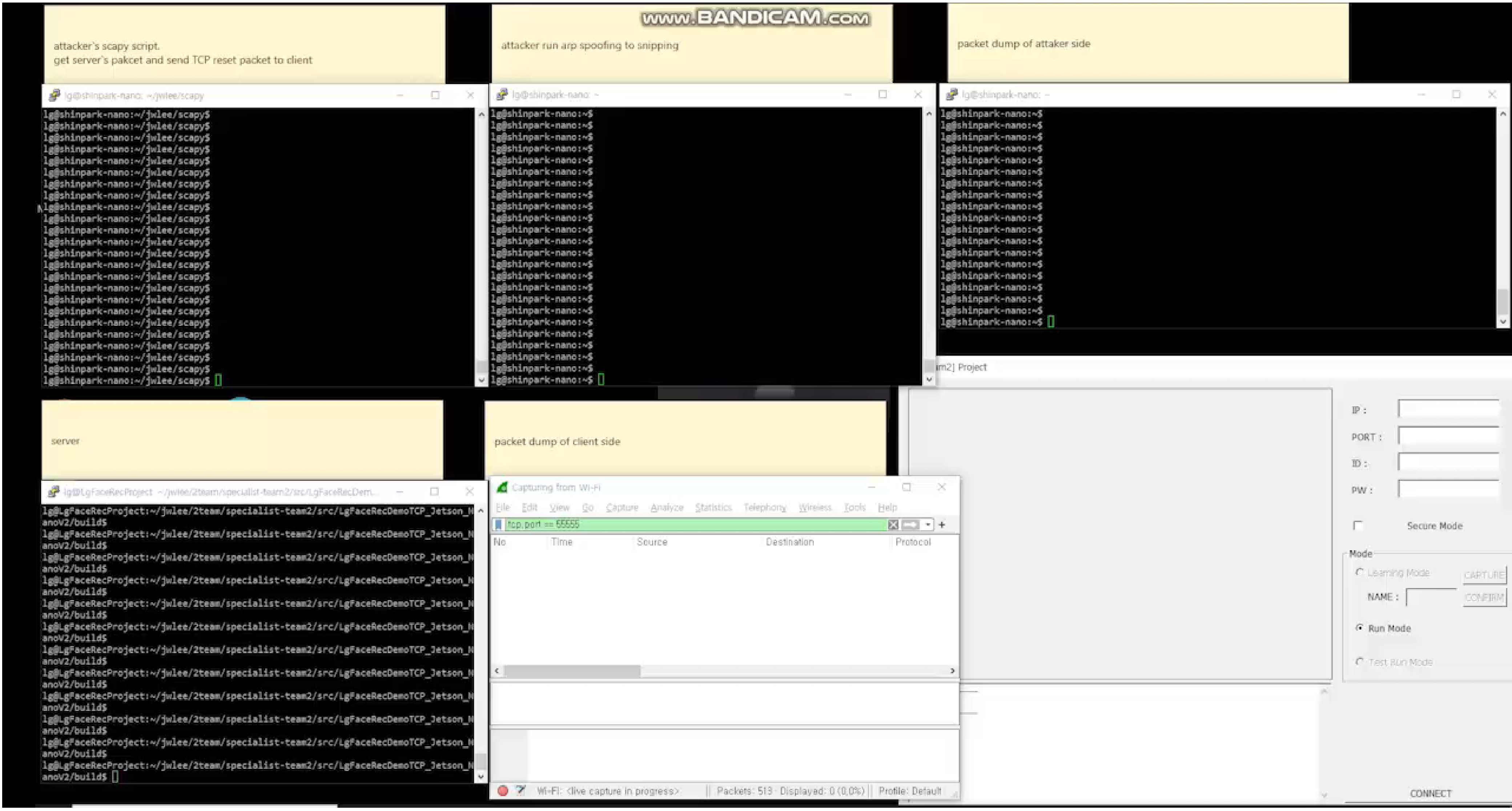
Normal ARP cache of Server



```
lg@LgFaceRecProject:~/jwlee$ arp -a
? (192.168.0.226) at 0c:54:15:55:bd:7e [ether] on wlan0
_gateway (192.168.0.1) at b0:95:75:ed:ed:43 [ether] on wlan0
? (192.168.0.112) at 0c:54:15:55:bd:7e [ether] on wlan0
? (192.168.0.228) at 0c:54:15:55:bd:7e [ether] on wlan0
? (192.168.0.134) at 50:e0:85:ca:65:f2 [ether] on wlan0
? (192.168.0.145) at 50:e0:85:ca:65:f2 [ether] on wlan0
lg@LgFaceRecProject:~/jwlee$
```

Spoofed ARP cache of Server

3. Test – Demo



attacker's scapy script.
get server's pakcet and send TCP reset packet to client

lg@shinpark-nano: ~/jwlee/scapy

```
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
lg@shinpark-nano:~/jwlee/scapy$
```

attacker run arp spoofing to snipping

lg@shinpark-nano: ~

```
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
```

packet dump of attaker side

lg@shinpark-nano: ~

```
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
lg@shinpark-nano:~$
```

server

lg@LgFaceRecProject: ~/jwlee/2team/specialist-team2/src/LgFaceRecDem...

```
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_N
anoV2/build$
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_N
anoV2/build$
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_N
anoV2/build$
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_N
anoV2/build$
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_N
anoV2/build$
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_N
anoV2/build$
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_N
anoV2/build$
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_N
anoV2/build$
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_N
anoV2/build$
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_N
anoV2/build$
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_N
anoV2/build$
lg@LgFaceRecProject:~/jwlee/2team/specialist-team2/src/LgFaceRecDemoTCP_Jetson_N
anoV2/build$
```

packet dump of client side

Capturing from Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 55555

No | Time | Source | Destination | Protocol

Wi-Fi: <live capture in progress> | Packets: 513 · Displayed: 0 (0.0%) | Profile: Default

m2] Project

IP :

PORT :

ID :

PW :

[ ] Secure Mode

Mode

( ) Learning Mode — NAME :

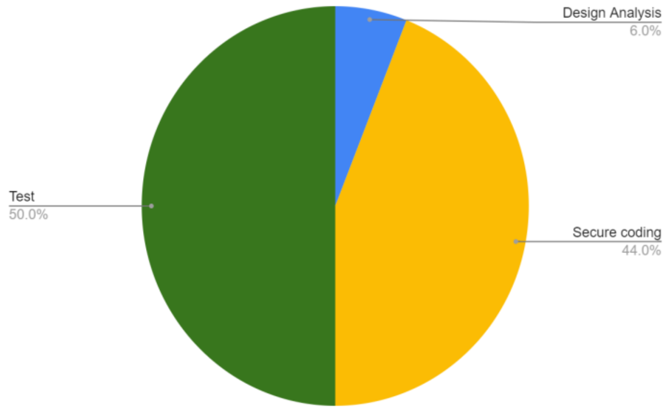
[✓ Run Mode]

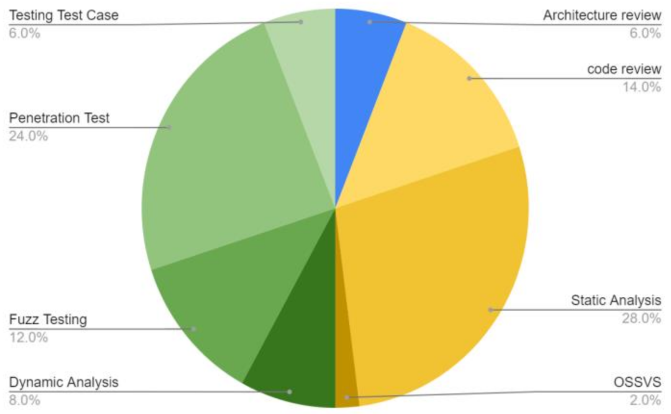
( ) Test Run Mode

4. Evaluation summary

Time distribution

- ✓ Total : Approximately 90 hours
- ✓ Test : 50%, Secure code review : 44%, Design Analysis : 6%





Identified Outcomes

Lesson	Activities	Vulnerabilities/Issues
Design Analysis	Architecture review	0
Secure coding	Eye inspection	3
	Flow finder	70 (but 63 is false positive)
	Sonar qube	17 (but 15 is false positive)
	OSSVS	0
	OpenVAS	2(but 1 is out of scope)
Test	Dynamic Analysis (ASAN)	0
	Fuzz Testing (AFL, Manual)	1
	Penetration Test (DoS, Memory)	2
	Testing Test Case	2

LG Electronics Carnegie Mellon University

5. Lessons and Learned

Lesson	Activities	Learned
Design Analysis	Architecture review	• Architecture should be designed considering all STRIDE.
Secure coding	code review	• Eye inspection code review are meaningful but have limited issue detection.
	Static Analysis	• Static analysis tool should consider several coding rule standards • Analysis process should be systematically managed.
	OSSVS	• The way to avoid open-source vulnerabilities is to always have the latest version.
Test	Dynamic Analysis	• Dynamic analysis can detect subtle flaws or vulnerabilities that cannot be detected by static analysis. • Dynamic and static analysis are complementary because a single approach cannot find all errors.
	Fuzz Testing	• Error handling have to consider Memory leaks.
	Penetration Test	• Trying to attack a system similarly to a malicious hacker can help you understand the importance of security.
	Testing Test Case	• A properly written requirement can create an accurate test case. • Vulnerabilities that could not be found through static analysis may be discovered through testing.

Appendix

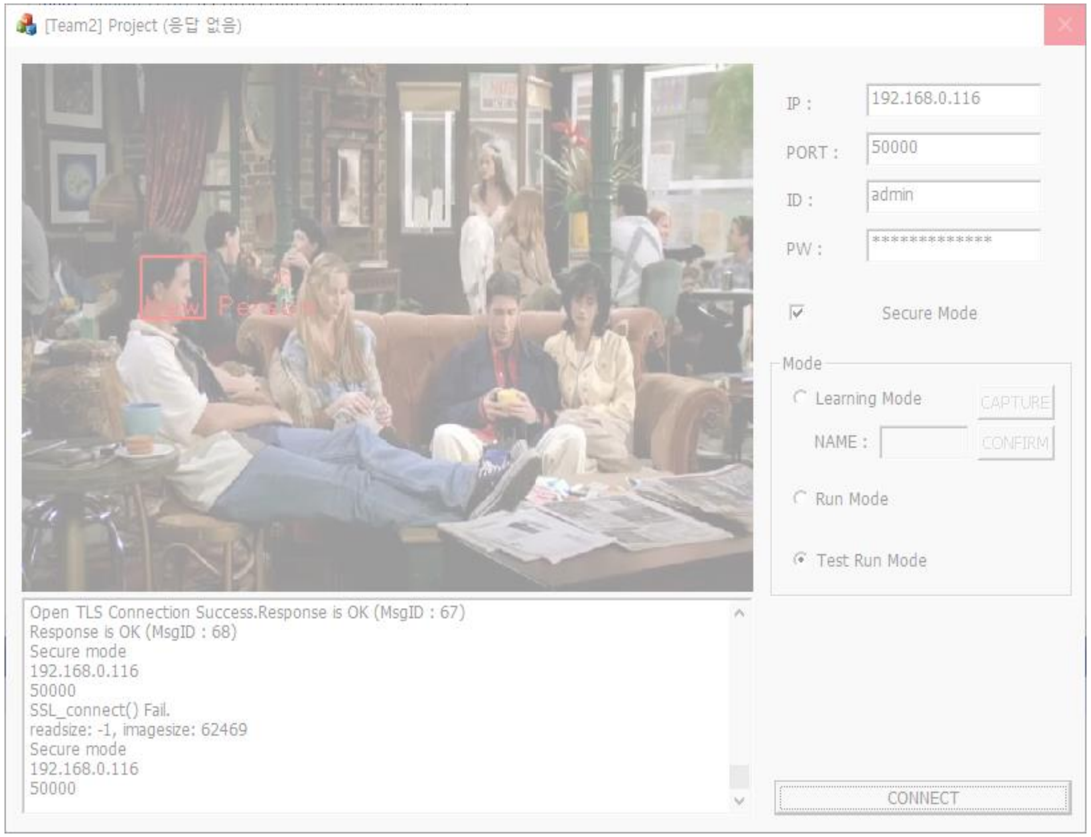
Appendix – Functional Test

Application operation test for functional verification.

- ✓ Although it was excluded from the security evaluation item, the verification of the basic function was also performed.
- ✓ These errors are occurred in **Secure Mode** only.

Case #1. No response in Test Run Mode

1. **Select Secure Mode**
2. **Select Test Run Mode**
3. **Repeat connect and disconnect**
4. **Client program has stopped working**



IP : 192.168.0.116
PORT : 50000
ID : admin
PW : *************

[✓ Secure Mode]

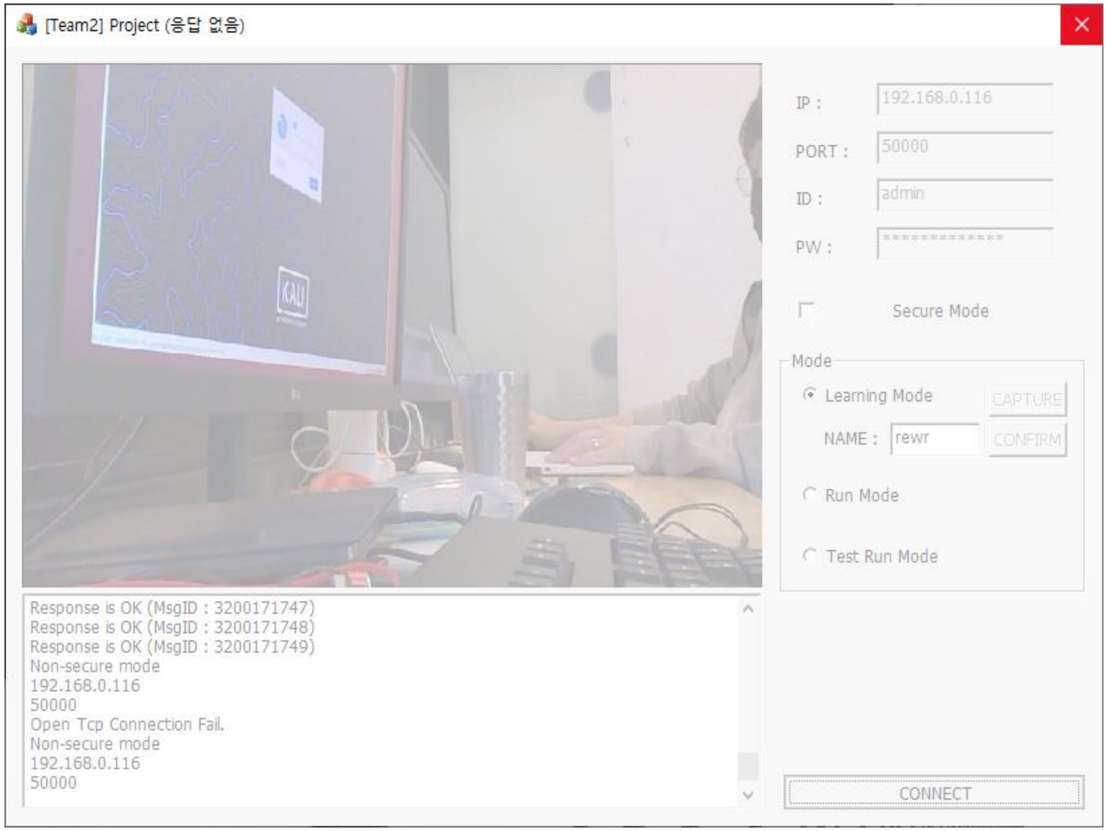
Mode
Learning Mode
NAME :
Run Mode
[✓ Test Run Mode]

```
Open TLS Connection Success.Response is OK (MsgID : 67)
Response is OK (MsgID : 68)
Secure mode
192.168.0.116
50000
SSL_connect() Fail.
readsize: -1, imagesize: 62469
Secure mode
192.168.0.116
50000
```

Appendix – Functional Test

Case #2. No response in Learning Mode

1. Select Secure Mode
2. Select Learning Mode
3. Press CONFIRM button repeatedly.
4. Select DISCONNECT button
5. Client program has stopped working



IP : 192.168.0.116
PORT : 50000
ID : admin
PW : *************
NAME : rewr

```
Response is OK (MsgID : 3200171747)
Response is OK (MsgID : 3200171748)
Response is OK (MsgID : 3200171749)
Non-secure mode
192.168.0.116
50000
Open Tcp Connection Fail.
Non-secure mode
192.168.0.116
50000
```

Thank you!