Project 1 Spot the Holes

Group Member Names: Nimoshika Jayaraman, Sukanya Sravasti

Note: In the report below, the github path to the vulnerabilities are mentioned in the first line ("/app/src/main/cpp/skyline/vfs/os_filesystem.cpp" is the github file link for the vulnerability described in this point). The main git rep where these files are located is: https://github.com/skyline-emu/skyline. The first paragraph outputs the output from flawfinder that flags the vulnerability.

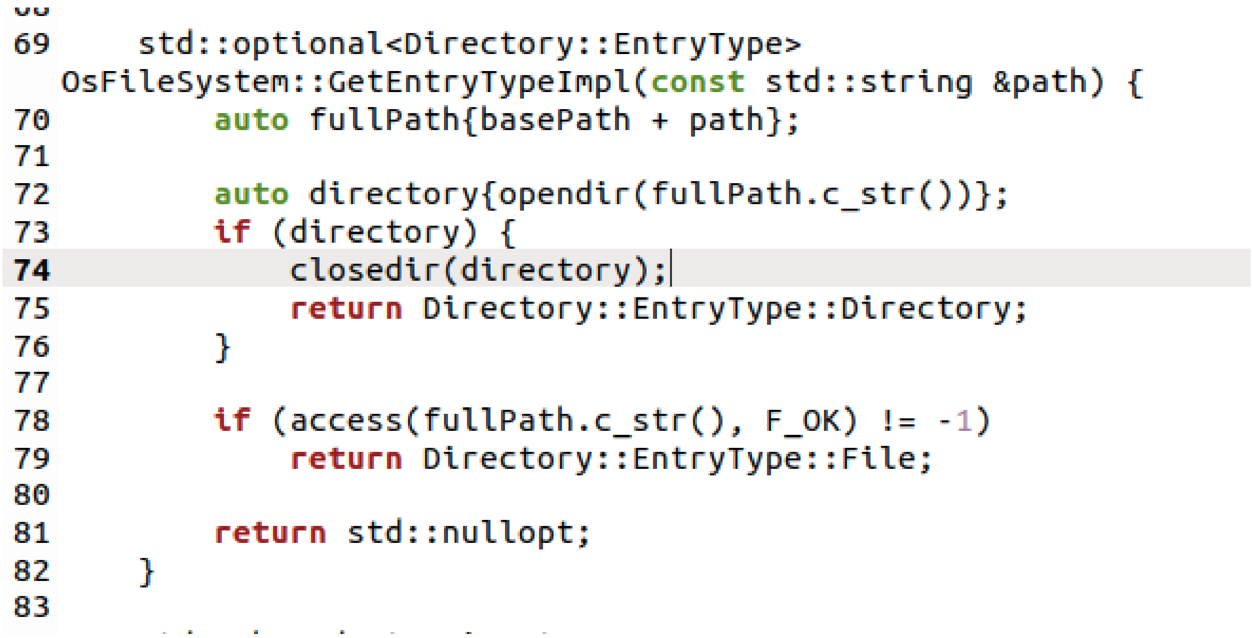
1./app/src/main/cpp/skyline/vfs/os_filesystem.cpp:78: [4] (race) access:
This usually indicates a security flaw. If an attacker can change anything
along the path between the call to access() and the file's actual use
(e.g., by moving files), the attacker can exploit the race condition
(CWE-362/CWE-367!). Set up the correct permissions (e.g., using setuid())
and try to open the file directly.

- **Vulnerability Description:** In this case, it is a vulnerability because the access function checks for a particular file name inside a directory, given the directory and file name. In doing so, there might be a "time to check and time to use" vulnerability because there might be a lag between accessing the file and using that file. For example, between the time when the access method is called using the filename and the file is returned, the attacker might change the interpretation of the filename.

  Other classes in this program are calling the same function "GetEntryTypeImpl" where this access method is defined, and this might lead to concurrent execution and race condition.

- **Categories:** Concurrency Attacks : Race conditions , time to check to time to use

- **Source Code Snippet:** In line 78, when the access command checks for the filename in the directory, a time to check and time to use vulnerability might occur. Since other classes in this software are also making multiple calls to this function "GetEntryTypeImpl", it might lead to a race condition and concurrent execution.

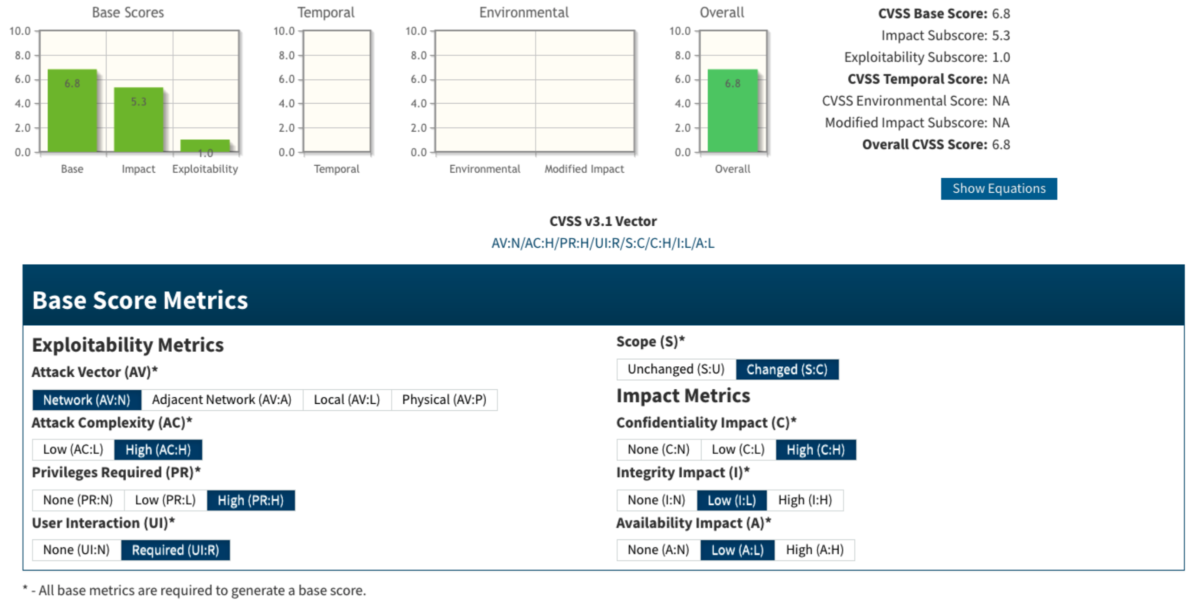


```
    std::optional<Directory::EntryType>
OsFileSystem::GetEntryTypeImpl(const std::string &path) {
        auto fullPath{basePath + path};

        auto directory{opendir(fullPath.c_str())};
        if (directory) {
            closedir(directory);
            return Directory::EntryType::Directory;
        }

        if (access(fullPath.c_str(), F_OK) != -1)
            return Directory::EntryType::File;

        return std::nullopt;
    }
```

- **Discovery Process:**

We ran the FlawFinder tool on the git rep of this software in Ubuntu.

- **Assessment of the Severity and Impact:**



CVSS Base Score: 6.8
Impact Subscore: 5.3
Exploitability Subscore: 1.0
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 6.8

CVSS v3.1 Vector
AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:L/A:L

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*
[✓ Network (AV:N)] Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*
Low (AC:L) [✓ High (AC:H)]

Privileges Required (PR)*
None (PR:N) Low (PR:L) [✓ High (PR:H)]

User Interaction (UI)*
None (UI:N) [✓ Required (UI:R)]

Scope (S)*
Unchanged (S:U) [✓ Changed (S:C)]

Impact Metrics

Confidentiality Impact (C)*
None (C:N) Low (C:L) [✓ High (C:H)]

Integrity Impact (I)*
None (I:N) [✓ Low (I:L)] High (I:H)

Availability Impact (A)*
None (A:N) [✓ Low (A:L)] High (A:H)

* - All base metrics are required to generate a base score.

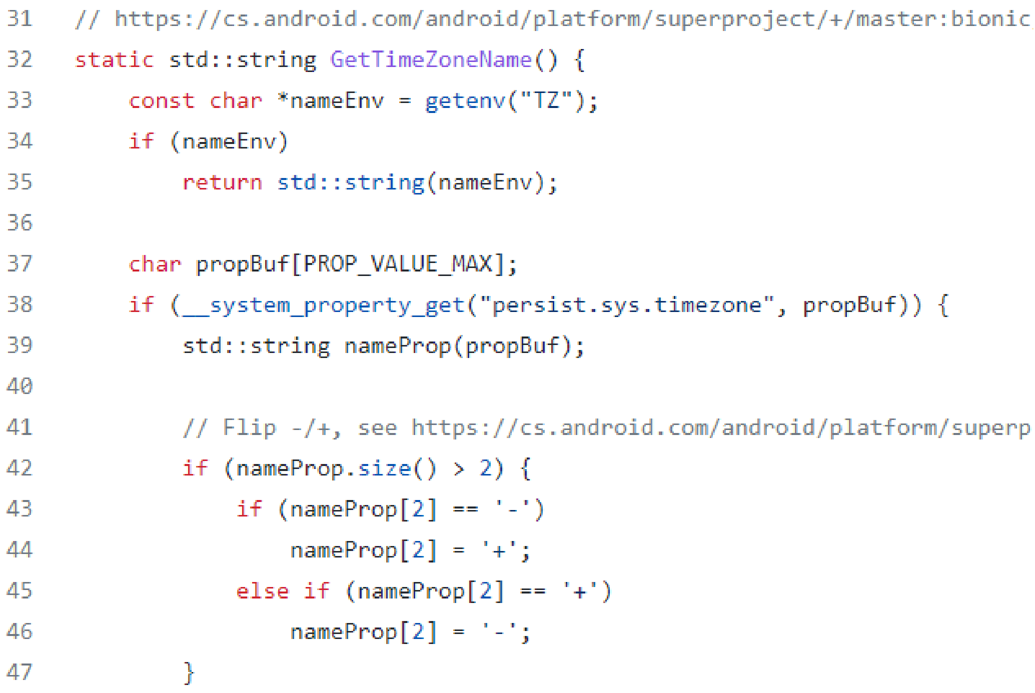
Overall Score: 6.8
Race conditions cause a very high confidentiality impact because the disclosed information presents a direct and serious impact. The integrity impact is low because the attacker does not have total control of modification. Availability impact is low as it does not have the ability to completely destroy the service to other users by making it unavailable.

2. ./app/src/main/cpp/emu_jni.cpp:33: [3] (buffer) getenv:
Environment variables are untrustable input if they can be set by an attacker. They can have any content and length, and the same variable can be set more than once (CWE-807, CWE-20). Check environment variables carefully before using them.

- **Vulnerability Description:** In this case, it is vulnerable because the program takes the input from the environment variables which can also be set by an attacker who is untrustable. The environmental variable thus inputted can have any content , could be of any length and can be set any number of times. The input data thus received is not validated or incorrectly validated. This condition could make the buffer overflow as more data is put into a fixed-length buffer than the buffer can handle.

- **Categories:** Spatial Memory Attacks: Out-of-bound access, buffer overflow.

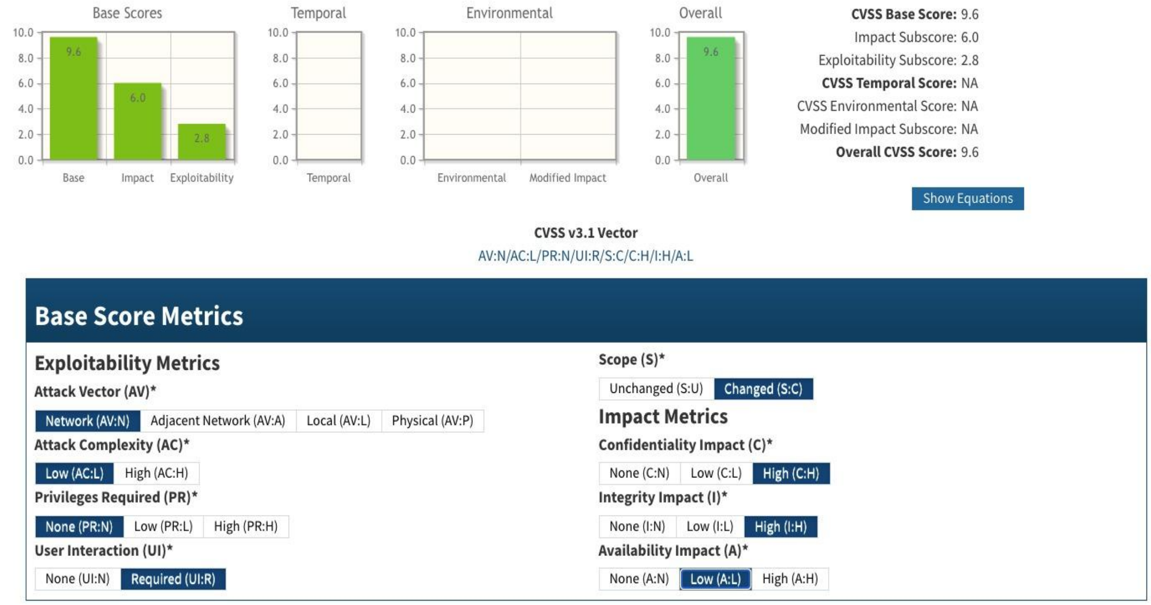
- **Source Code Snippet:** In line 33, the program gets the environment variable" Time Zone" which can also be set by an attacker who is untrustable. The environmental variable thus inputted can have any content , could be of any length and can be set any number of times. The input data thus received is not validated or incorrectly validated. This condition could make the buffer overflow as more data is put into a fixed-length buffer than the buffer can handle.



```
31  // https://cs.android.com/android/platform/superproject/+/master:bionic
32  static std::string GetTimeZoneName() {
33      const char *nameEnv = getenv("TZ");
34      if (nameEnv)
35          return std::string(nameEnv);
36
37      char propBuf[PROP_VALUE_MAX];
38      if (__system_property_get("persist.sys.timezone", propBuf)) {
39          std::string nameProp(propBuf);
40
41          // Flip -/+, see https://cs.android.com/android/platform/superp
42          if (nameProp.size() > 2) {
43              if (nameProp[2] == '-')
44                  nameProp[2] = '+';
45              else if (nameProp[2] == '+')
46                  nameProp[2] = '-';
47          }
```

- **Discovery Process:** We ran the FlawFinder tool on the git rep of this software in Ubuntu.

Assessment of the Severity and Impact:



CVSS Base Score: 9.6
Impact Subscore: 6.0
Exploitability Subscore: 2.8
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 9.6

CVSS v3.1 Vector
AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:L

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*
[✓ Network (AV:N)] Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*
[✓ Low (AC:L)] High (AC:H)

Privileges Required (PR)*
[✓ None (PR:N)] Low (PR:L) High (PR:H)

User Interaction (UI)*
None (UI:N) [✓ Required (UI:R)]

Scope (S)*
Unchanged (S:U) [✓ Changed (S:C)]

Impact Metrics

Confidentiality Impact (C)*
None (C:N) Low (C:L) [✓ High (C:H)]

Integrity Impact (I)*
None (I:N) Low (I:L) [✓ High (I:H)]

Availability Impact (A)*
None (A:N) [✓ Low (A:L)] High (A:H)

Overall Score: 8.8
The confidentiality impact is high as the impacted component is being divulged to the attacker. Integrity impact is high as it is a complete loss of protection. Availability impact is low as it does not have the ability to completely destroy the service to other users by making it unavailable.

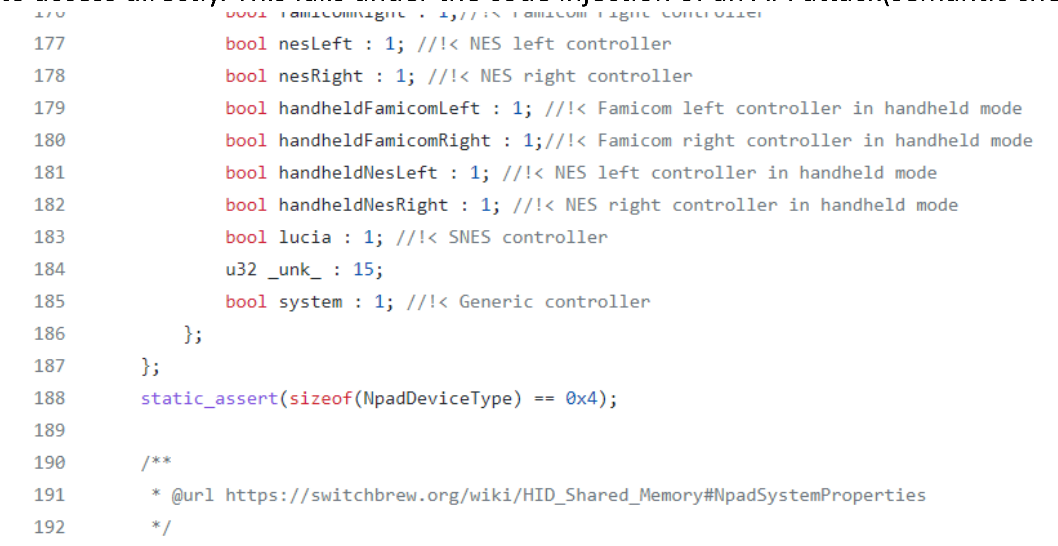
3. ./app/src/main/cpp/skyline/input/sections/Npad.h:185: [4] (shell) system:
This causes a new program to execute and is difficult to use safely
(CWE-78). try using a library call that implements the same functionality if available.

- **Vulnerability Description:** In this case, the application intends to execute a single, fixed program that is under its own control. It intends to use externally-supplied inputs as arguments to that program. The generic controller allows the user to give input which is taken as an argument. Attackers could place the separators into the arguments, which allows them to execute their own program after the controller has finished executing**.** Attackers could execute unauthorized commands, which could then be used to disable the software, or read and modify data for which the attacker does not have permissions to access directly.

- **Categories:** API attack : semantic check bypass

- **Source Code Snippet:**

In line 185, the generic controller gets the input from the player from the API (Application Program Interface) which leads to the interface attacks. The generic controller allows the user to give input which is taken as an argument. attackers could place the separators into the arguments, which allows them to execute their own program after the controller has finished executing. Attackers could execute unauthorized commands, which could then be used to disable the software, or read and modify data for which the attacker does not have permissions to access directly. This falls under the code injection of an API attack(semantic check bypass).

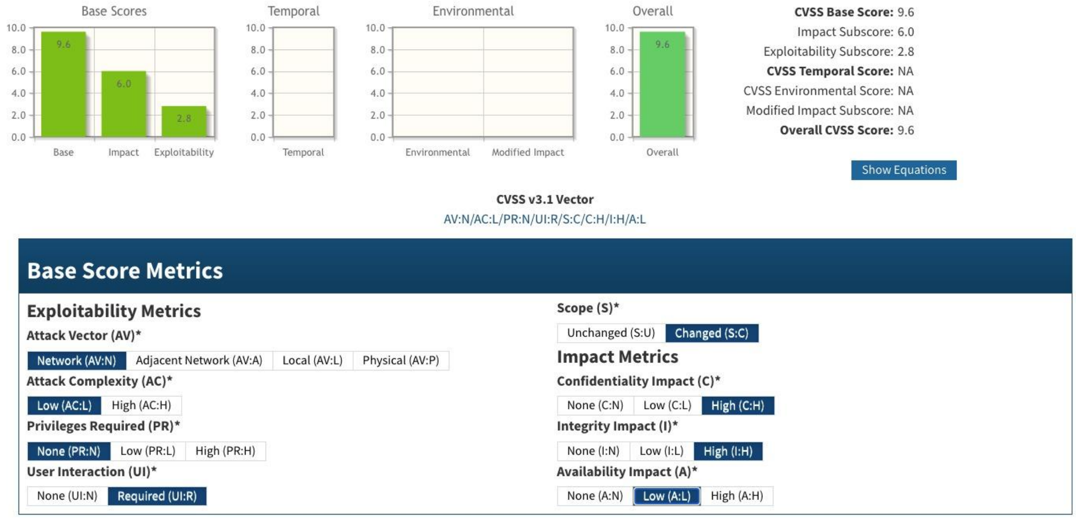


```
                bool famicomRight : 1;//!< Famicom right controller
                bool nesLeft : 1; //!< NES left controller
                bool nesRight : 1; //!< NES right controller
                bool handheldFamicomLeft : 1; //!< Famicom left controller in handheld mode
                bool handheldFamicomRight : 1;//!< Famicom right controller in handheld mode
                bool handheldNesLeft : 1; //!< NES left controller in handheld mode
                bool handheldNesRight : 1; //!< NES right controller in handheld mode
                bool lucia : 1; //!< SNES controller
                u32 _unk_ : 15;
                bool system : 1; //!< Generic controller
            };
        };
        static_assert(sizeof(NpadDeviceType) == 0x4);

        /**
         * @url https://switchbrew.org/wiki/HID_Shared_Memory#NpadSystemProperties
         */
```

- **Discovery Process:**

We ran the FlawFinder tool on the git rep of this software in Ubuntu.

- **Assessment of the Severity and Impact:**



CVSS Base Score: 9.6
Impact Subscore: 6.0
Exploitability Subscore: 2.8
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 9.6

CVSS v3.1 Vector
AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:L

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*
[✓ Network (AV:N)] Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*
[✓ Low (AC:L)] High (AC:H)

Privileges Required (PR)*
[✓ None (PR:N)] Low (PR:L) High (PR:H)

User Interaction (UI)*
None (UI:N) [✓ Required (UI:R)]

Scope (S)*
Unchanged (S:U) [✓ Changed (S:C)]

Impact Metrics

Confidentiality Impact (C)*
None (C:N) Low (C:L) [✓ High (C:H)]

Integrity Impact (I)*
None (I:N) Low (I:L) [✓ High (I:H)]

Availability Impact (A)*
None (A:N) [✓ Low (A:L)] High (A:H)

Overall Score: 9.6
The confidentiality impact is high as the impacted component is being divulged to the attacker. Integrity impact is high as it is a complete loss of protection. Availability impact is low as it does not have the ability to completely destroy the service to other users by making it unavailable.

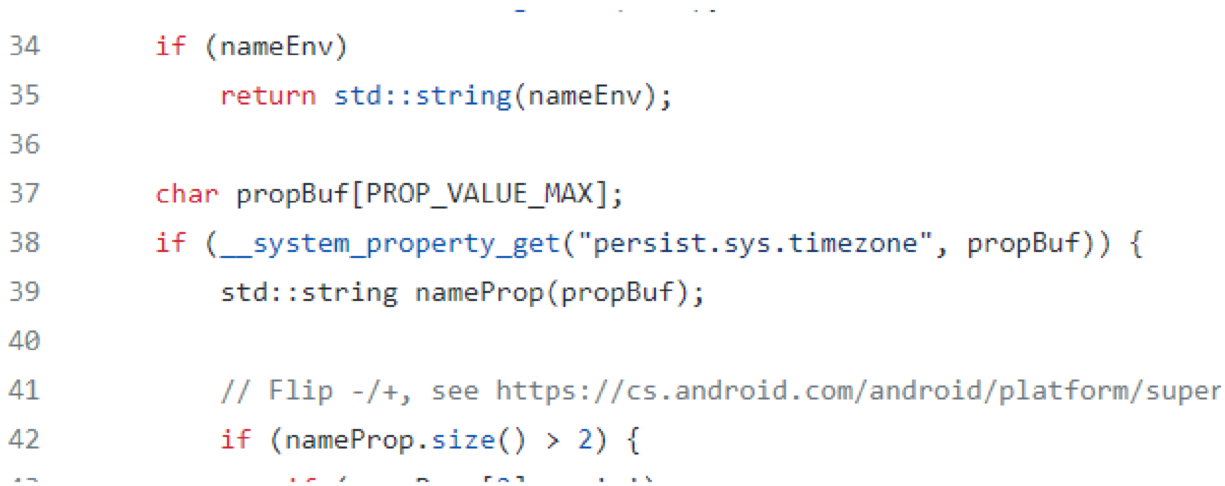
4. ./app/src/main/cpp/emu_jni.cpp:37: [2] (buffer) char:
Statically-sized arrays can be improperly restricted, leading to potential overflows or other issues (CWE-119!/CWE-120). Perform bounds checking, use

functions that limit length, or ensure that the size is larger than the maximum possible length.

- **Vulnerability Description:** While looking for a time zone as input, it can have any content, could be of any length and can be set any number of times which leads to buffer overflow. It goes beyond the property maximum value set in the property buffer. Through this an attacker may be able to execute arbitrary code, alter the intended control flow, read sensitive information, or cause the system to crash.

- **Categories:** Spatial Memory Attacks : buffer overflow

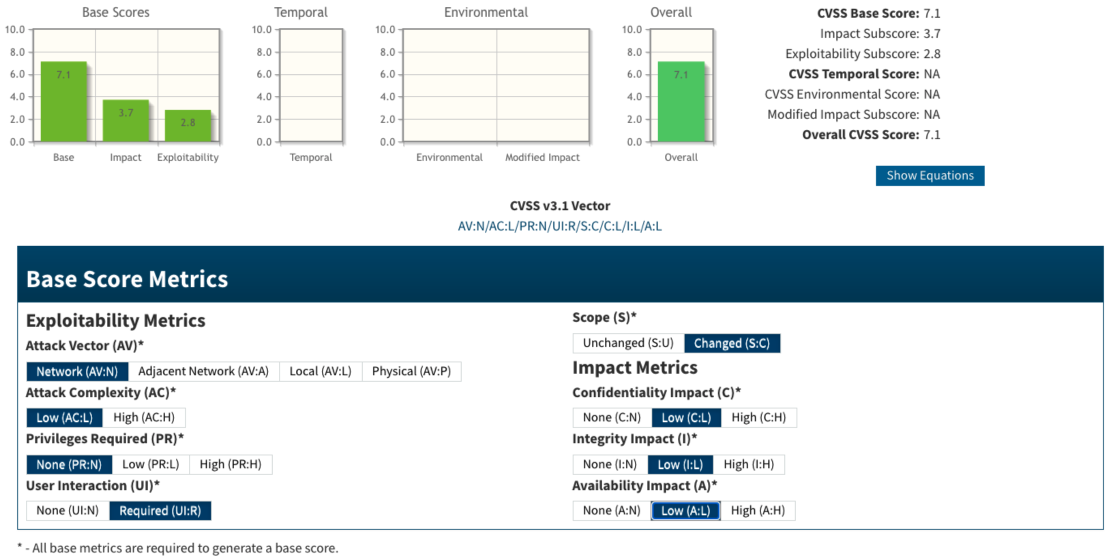
- **Source Code Snippet:** In line 37, “PROP_VALUE_MAX” looks up a system property by name, copying its value and a \0 terminator to the provided pointer. The total bytes copied will be no greater than PROP_VALUE_MAX. It then returns the string length of the value. While looking for a time zone in line 38 as input, it can have any content , could be of any length and can be set any number of times which leads to buffer overflow. It goes beyond the property maximum value set in the property buffer. Through this an attacker may be able to execute arbitrary code, alter the intended control flow, read sensitive information, or cause the system to crash.



```
34        if (nameEnv)
35            return std::string(nameEnv);
36
37        char propBuf[PROP_VALUE_MAX];
38        if (__system_property_get("persist.sys.timezone", propBuf)) {
39            std::string nameProp(propBuf);
40
41            // Flip -/+, see https://cs.android.com/android/platform/super
42            if (nameProp.size() > 2) {
```

- **Discovery Process:** We ran the FlawFinder tool on the git rep of this software in Ubuntu.

- **Assessment of the Severity and Impact:**



CVSS Base Score: 7.1
Impact Subscore: 3.7
Exploitability Subscore: 2.8
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 7.1

CVSS v3.1 Vector
AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:L

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*
[✓ Network (AV:N)] Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*
[✓ Low (AC:L)] High (AC:H)

Privileges Required (PR)*
[✓ None (PR:N)] Low (PR:L) High (PR:H)

User Interaction (UI)*
None (UI:N) [✓ Required (UI:R)]

Scope (S)*
Unchanged (S:U) [✓ Changed (S:C)]

Impact Metrics

Confidentiality Impact (C)*
None (C:N) [✓ Low (C:L)] High (C:H)

Integrity Impact (I)*
None (I:N) [✓ Low (I:L)] High (I:H)

Availability Impact (A)*
None (A:N) [✓ Low (A:L)] High (A:H)

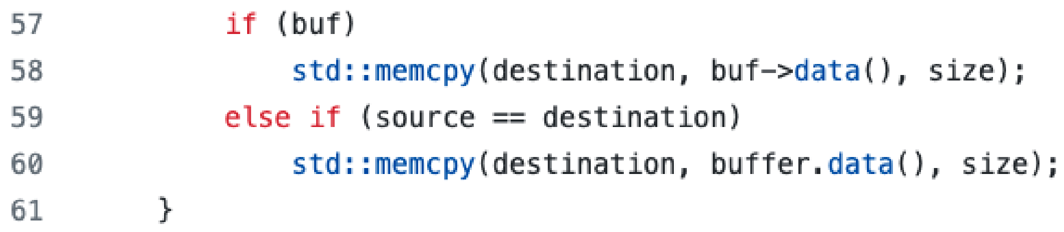
* - All base metrics are required to generate a base score.

The overall CVSS score is 7.1, which is somewhat high. This vulnerability impacts confidentiality, integrity and availability. Confidentiality impact is low since the vulnerability might provide access to some restricted information but the attacker does not have control over the information obtained. Integrity impact is low because the attacker might be able to modify some data, but the attacker does not necessarily have control over the amount of modification. Availability impact is low since buffer overflow might cause the game to crash and have reduced performance, but the attacker does not have the ability to completely deny service to legitimate users.

5. ./app/src/main/cpp/skyline/crypto/aes_cipher.cpp:58: [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure the destination can always hold the source data.

./app/src/main/cpp/skyline/crypto/aes_cipher.cpp:60: [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure the destination can always hold the source data.

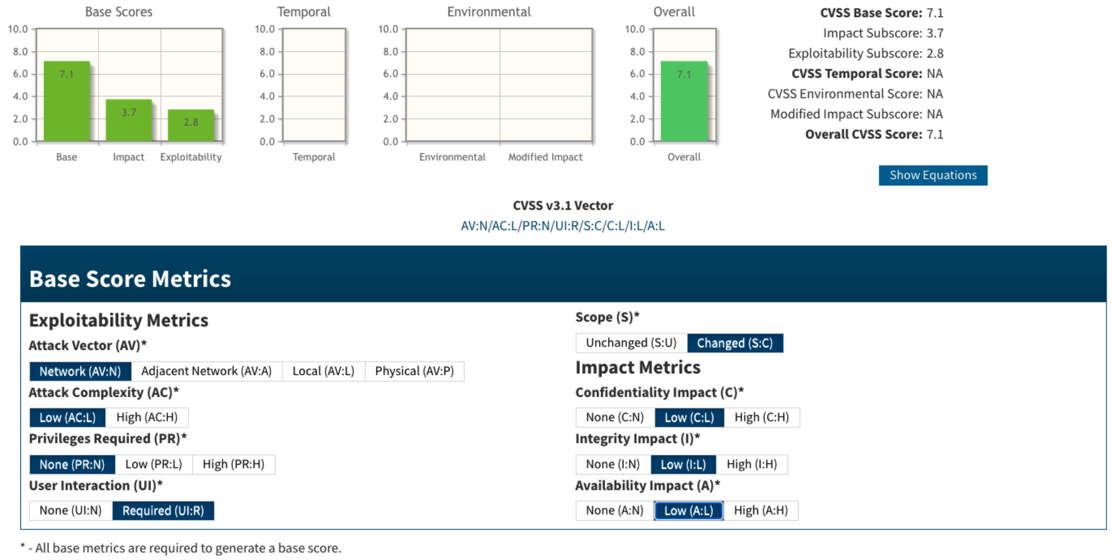
- **Vulnerability Description:** The vulnerability is a buffer overflow. Without verifying the sizes of the input and output buffers, the program copies the input buffer to the output buffer. This leads to a buffer overflow and this overflow can corrupt adjacent storage. Such a vulnerability can cause the program to crash and it creates entry points for attacks.
- **Categories:** Spatial Memory Attacks: Buffer Overflow, Out-of-Bound Access
- **Source Code Snippet:**



```
57            if (buf)
58                std::memcpy(destination, buf->data(), size);
59            else if (source == destination)
60                std::memcpy(destination, buffer.data(), size);
61        }
```

In this program, the command memcpy in lines 58 and 60 copies data from the sources (variables “buf” and “buffer”) to the destination. Within these “if” and “else if” statements, the code is not checking for buffer overflow in case the size of the source variables “buf” or “buffs” exceeds the destination size.

- **Discovery Process:** We ran the FlawFinder tool on the git rep of this software in Ubuntu.
- **Assessment of the severity and impact:** The overall CVSS score is 7.1, which is somewhat high. This vulnerability impacts confidentiality, integrity and availability. Confidentiality impact is low since the vulnerability might provide access to some restricted information but the attacker does not have control over the information obtained. Integrity impact is low because the attacker might be able to modify some data, but the attacker does not necessarily have control over the amount of modification. Availability impact is low since buffer overflow might cause the game to crash and have reduced performance, but the attacker does not have the ability to completely deny service to legitimate users.



CVSS Base Score: 7.1
Impact Subscore: 3.7
Exploitability Subscore: 2.8
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 7.1

CVSS v3.1 Vector
AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:L

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*
[✓ Network (AV:N)] Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*
[✓ Low (AC:L)] High (AC:H)

Privileges Required (PR)*
[✓ None (PR:N)] Low (PR:L) High (PR:H)

User Interaction (UI)*
None (UI:N) [✓ Required (UI:R)]

Scope (S)*
Unchanged (S:U) [✓ Changed (S:C)]

Impact Metrics

Confidentiality Impact (C)*
None (C:N) [✓ Low (C:L)] High (C:H)

Integrity Impact (I)*
None (I:N) [✓ Low (I:L)] High (I:H)

Availability Impact (A)*
None (A:N) [✓ Low (A:L)] High (A:H)

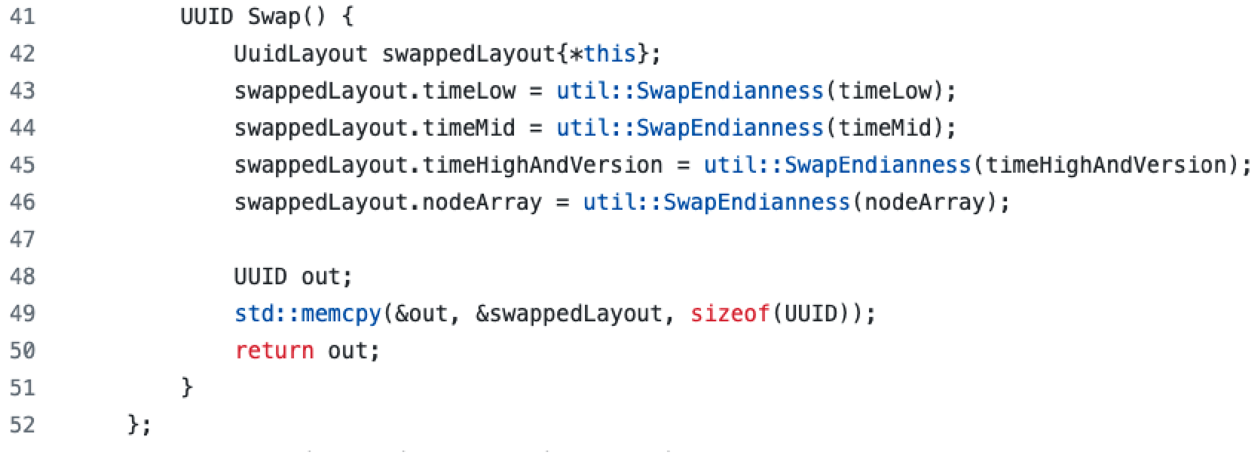
* - All base metrics are required to generate a base score.

6. ./app/src/main/cpp/skyline/common/uuid.cpp:49: [2] (buffer) memcpy:
Does not check for buffer overflows when copying to destination (CWE-120).
Make sure the destination can always hold the source data.

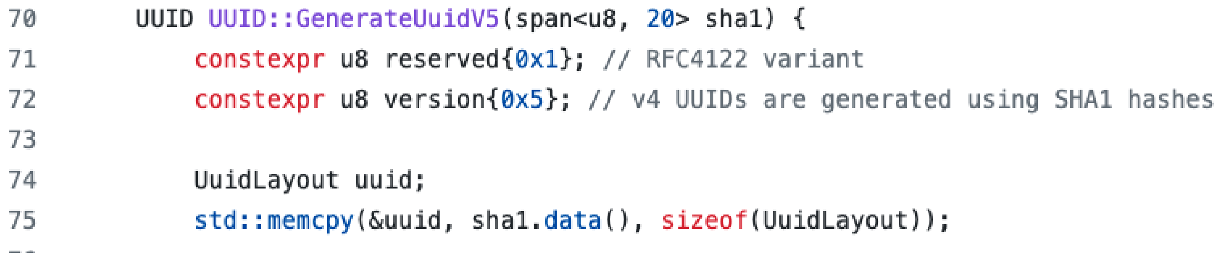
./app/src/main/cpp/skyline/common/uuid.cpp:75: [2] (buffer) memcpy:

Does not check for buffer overflows when copying to destination (CWE-120).
Make sure the destination can always hold the source data.

- **Vulnerability Description:** The vulnerability is a buffer overflow. Without verifying the sizes of the input and output buffers, the program copies the input buffer to the output buffer. This leads to a buffer overflow and this overflow can corrupt adjacent storage. Such a vulnerability can cause the program to crash and it creates entry points for attacks.
- **Categories:** Spatial Memory Attacks: Buffer Overflow, Out-of-Bound Access
- **Source Code Snippet:**



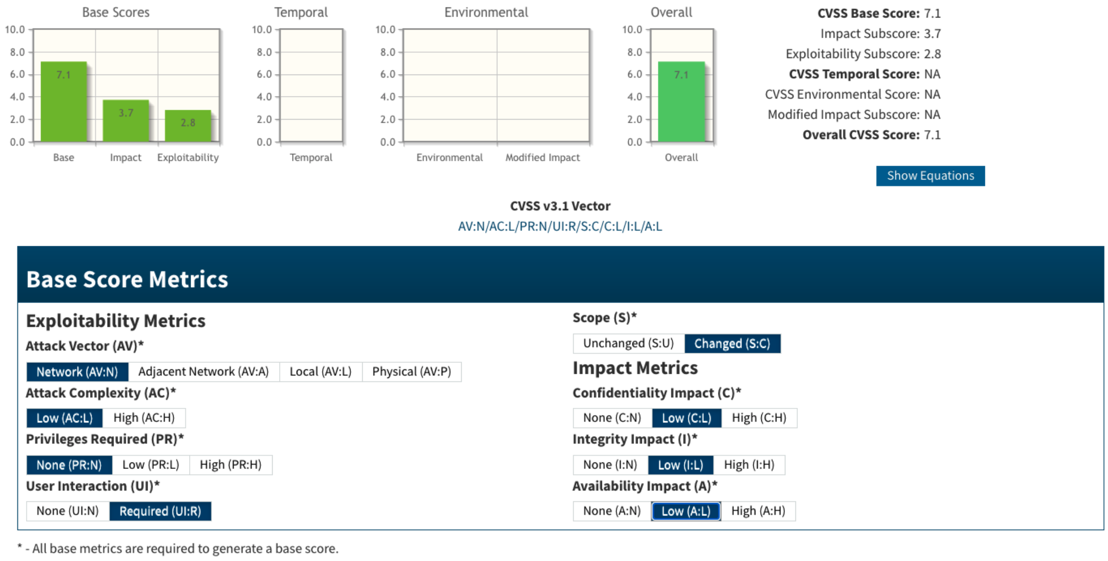
```
41          UUID Swap() {
42              UuidLayout swappedLayout{*this};
43              swappedLayout.timeLow = util::SwapEndianness(timeLow);
44              swappedLayout.timeMid = util::SwapEndianness(timeMid);
45              swappedLayout.timeHighAndVersion = util::SwapEndianness(timeHighAndVersion);
46              swappedLayout.nodeArray = util::SwapEndianness(nodeArray);
47
48              UUID out;
49              std::memcpy(&out, &swappedLayout, sizeof(UUID));
50              return out;
51          }
52      };
```



```
70      UUID UUID::GenerateUuidV5(span<u8, 20> sha1) {
71          constexpr u8 reserved{0x1}; // RFC4122 variant
72          constexpr u8 version{0x5}; // v4 UUIDs are generated using SHA1 hashes
73
74          UuidLayout uuid;
75          std::memcpy(&uuid, sha1.data(), sizeof(UuidLayout));
```

In this program, the command memcpy in lines 49 and 75 copies data from the sources (variables “swappedLayout” and “sha1”) to the destinations (variables “out” and “uuid”). In both instances, the code is not checking for buffer overflow in case the size of the source variables exceeds that of the destination variables.

- **Discovery Process:** We ran the FlawFinder tool on the git rep of this software in Ubuntu.
- **Assessment of the severity and impact:** The overall CVSS score is 7.1, which is somewhat high. This vulnerability impacts confidentiality, integrity and availability. Confidentiality impact is low since the vulnerability might provide access to some restricted information but the attacker does not have control over the information obtained. Integrity impact is low because the attacker might be able to modify some data, but the attacker does not necessarily have control over the amount of modification. Availability impact is low since buffer overflow might cause the game to crash and have reduced performance, but the attacker does not have the ability to completely deny service to legitimate users.



CVSS Base Score: 7.1
Impact Subscore: 3.7
Exploitability Subscore: 2.8
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 7.1

CVSS v3.1 Vector
AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:L

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*
[✓ Network (AV:N)] Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*
[✓ Low (AC:L)] High (AC:H)

Privileges Required (PR)*
[✓ None (PR:N)] Low (PR:L) High (PR:H)

User Interaction (UI)*
None (UI:N) [✓ Required (UI:R)]

Scope (S)*
Unchanged (S:U) [✓ Changed (S:C)]

Impact Metrics

Confidentiality Impact (C)*
None (C:N) [✓ Low (C:L)] High (C:H)

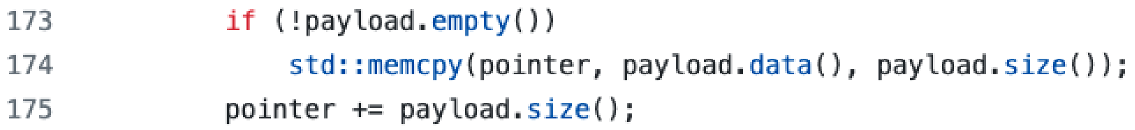
Integrity Impact (I)*
None (I:N) [✓ Low (I:L)] High (I:H)

Availability Impact (A)*
None (A:N) [✓ Low (A:L)] High (A:H)

* - All base metrics are required to generate a base score.

7. ./app/src/main/cpp/skyline/kernel/ipc.cpp:174: [2] (buffer) memcpy:
Does not check for buffer overflows when copying to destination (CWE-120).
Make sure the destination can always hold the source data.

- **Vulnerability Description:** The vulnerability is a buffer overflow. Without verifying the sizes of the input and output buffers, the program copies the input buffer to the output buffer. This leads to a buffer overflow and this overflow can corrupt adjacent storage. Such a vulnerability can cause the program to crash and it creates entry points for attacks.
- **Categories:** Spatial Memory Attacks: Buffer Overflow, Out-of-Bound Access
- **Source Code Snippet:**

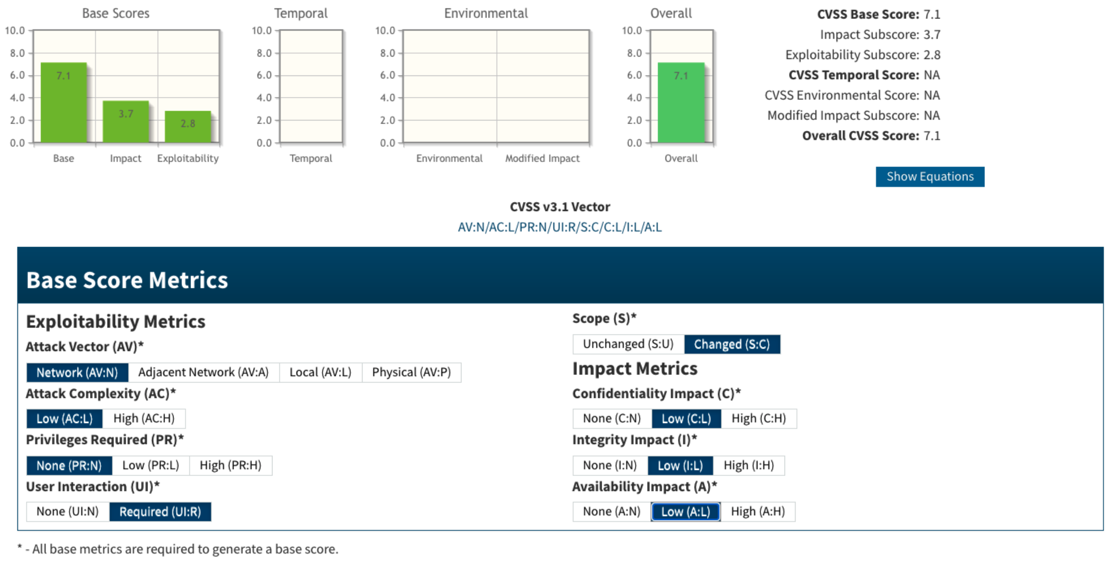


```
173            if (!payload.empty())
174                std::memcpy(pointer, payload.data(), payload.size());
175            pointer += payload.size();
```

In this program, the command memcpy in line 174 copies data from the sources (variable “payload”) to the destination (variable “pointer”). In this if statement, the code is not checking for buffer overflow in case the size of the source exceeds that of the destination.

- **Discovery Process:** We ran the FlawFinder tool on the git rep of this software in Ubuntu.
- **Assessment of the severity and impact:** The overall CVSS score is 7.1, which is somewhat high. This vulnerability impacts confidentiality, integrity and availability. Confidentiality impact is low since the vulnerability might provide access to some restricted information but the attacker does not have control over the information obtained. Integrity impact is low because the attacker might be able to modify some data, but the attacker does not necessarily have control over the amount of modification. Availability impact is low since buffer overflow might cause the game to

crash and have reduced performance, but the attacker does not have the ability to completely deny service to legitimate users.



CVSS Base Score: 7.1
Impact Subscore: 3.7
Exploitability Subscore: 2.8
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 7.1

CVSS v3.1 Vector
AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:L

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*
[✓ Network (AV:N)] Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*
[✓ Low (AC:L)] High (AC:H)

Privileges Required (PR)*
[✓ None (PR:N)] Low (PR:L) High (PR:H)

User Interaction (UI)*
None (UI:N) [✓ Required (UI:R)]

Scope (S)*
Unchanged (S:U) [✓ Changed (S:C)]

Impact Metrics

Confidentiality Impact (C)*
None (C:N) [✓ Low (C:L)] High (C:H)

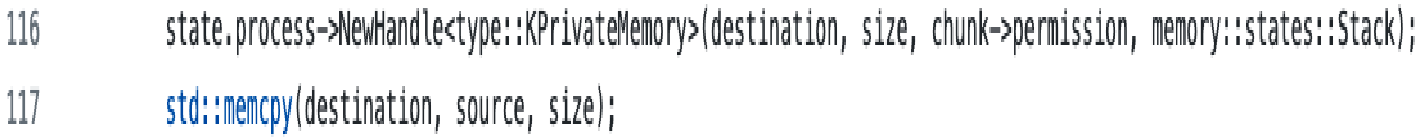
Integrity Impact (I)*
None (I:N) [✓ Low (I:L)] High (I:H)

Availability Impact (A)*
None (A:N) [✓ Low (A:L)] High (A:H)

* - All base metrics are required to generate a base score.

8. ./app/src/main/cpp/skyline/kernel/svc.cpp:117: [2] (buffer) memcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Make sure destination can always hold the source data.

- **Vulnerability Description:** The vulnerability is a buffer overflow. Without verifying the sizes of the input and output buffers, the program copies the input buffer to the output buffer. This leads to a buffer overflow and this overflow can corrupt adjacent storage. Such a vulnerability can cause the program to crash and it creates entry points for attacks.
- **Categories:** Spatial Memory Attacks: Buffer Overflow, Out-of-Bound Access
- **Source Code Snippet:**

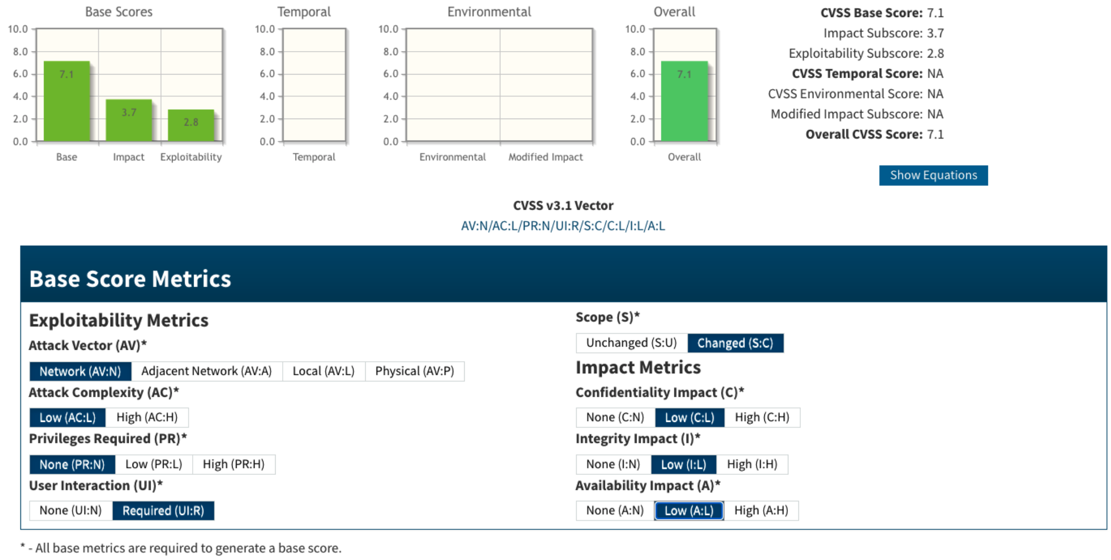


```
116         state.process->NewHandle<type::KPrivateMemory>(destination, size, chunk->permission, memory::states::Stack);
117         std::memcpy(destination, source, size);
```

In this program, the command memcpy in line 117 copies data from the source to the destination. The code is not checking for buffer overflow in case the size of the source exceeds that of the destination.

- **Discovery Process:** We ran the FlawFinder tool on the git rep of this software in Ubuntu.
- **Assessment of the severity and impact:** The overall CVSS score is 7.1, which is somewhat high. This vulnerability impacts confidentiality, integrity and availability. Confidentiality impact is low since the vulnerability might provide access to some restricted information but the attacker does not have control over the information obtained. Integrity impact is low because the attacker might be able to modify some

data, but the attacker does not necessarily have control over the amount of modification. Availability impact is low since buffer overflow might cause the game to crash and have reduced performance, but the attacker does not have the ability to completely deny service to legitimate users.



CVSS Base Score: 7.1
Impact Subscore: 3.7
Exploitability Subscore: 2.8
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 7.1

CVSS v3.1 Vector
AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:L

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*
[✓ Network (AV:N)] Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*
[✓ Low (AC:L)] High (AC:H)

Privileges Required (PR)*
[✓ None (PR:N)] Low (PR:L) High (PR:H)

User Interaction (UI)*
None (UI:N) [✓ Required (UI:R)]

Scope (S)*
Unchanged (S:U) [✓ Changed (S:C)]

Impact Metrics

Confidentiality Impact (C)*
None (C:N) [✓ Low (C:L)] High (C:H)

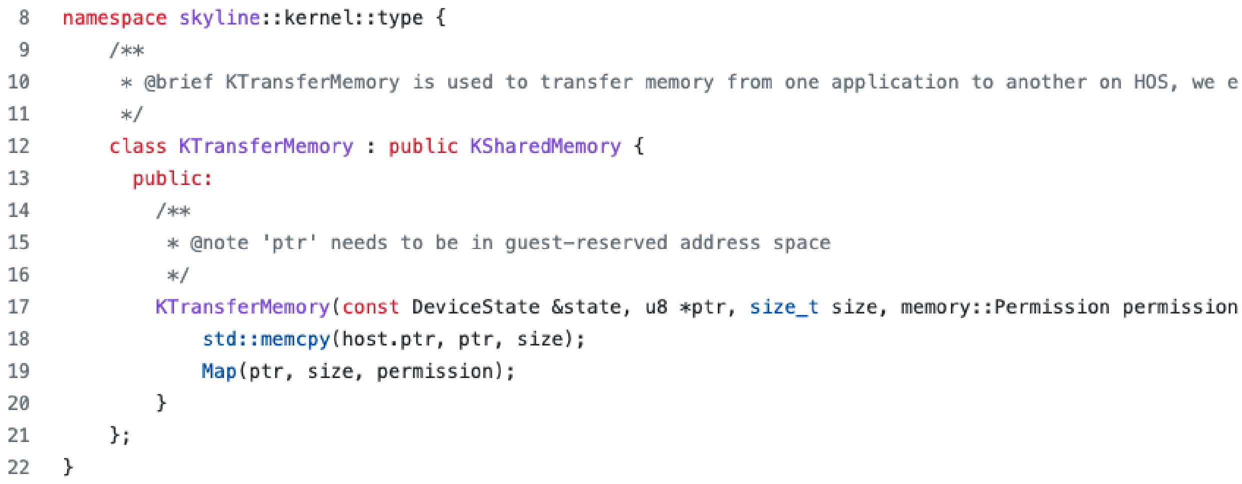
Integrity Impact (I)*
None (I:N) [✓ Low (I:L)] High (I:H)

Availability Impact (A)*
None (A:N) [✓ Low (A:L)] High (A:H)

* - All base metrics are required to generate a base score.

9. ./app/src/main/cpp/skyline/kernel/types/KTransferMemory.h:18: [2] (buffer) memcpy:
Does not check for buffer overflows when copying to destination (CWE-120).
Make sure destination can always hold the source data.

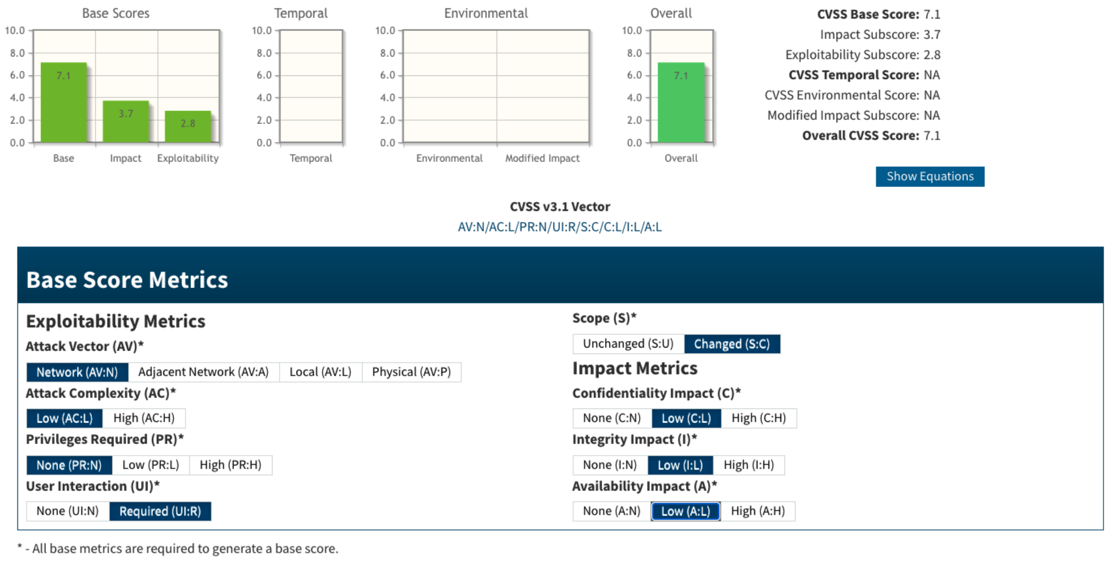
- **Vulnerability Description:** The vulnerability is a buffer overflow. Without verifying the sizes of the input and output buffers, the program copies the input buffer to the output buffer. This leads to a buffer overflow and this overflow can corrupt adjacent storage. Such a vulnerability can cause the program to crash and it creates entry points for attacks.
- **Categories:** Spatial Memory Attacks: Buffer Overflow, Out-of-Bound Access
- **Source Code Snippet:**



```
 8  namespace skyline::kernel::type {
 9      /**
10       * @brief KTransferMemory is used to transfer memory from one application to another on HOS, we e
11       */
12      class KTransferMemory : public KSharedMemory {
13        public:
14          /**
15           * @note 'ptr' needs to be in guest-reserved address space
16           */
17          KTransferMemory(const DeviceState &state, u8 *ptr, size_t size, memory::Permission permission
18              std::memcpy(host.ptr, ptr, size);
19              Map(ptr, size, permission);
20          }
21      };
22  }
```

In this program, the command memcpy in line 18 copies data from the source (ptr) to the destination (host.ptr). The code is not checking for buffer overflow in case the size of the source exceeds that of the destination.

- **Discovery Process:** We ran the FlawFinder tool on the git rep of this software in Ubuntu.
- **Assessment of the severity and impact:** The overall CVSS score is 7.1, which is somewhat high. This vulnerability impacts confidentiality, integrity and availability. Confidentiality impact is low since the vulnerability might provide access to some restricted information but the attacker does not have control over the information obtained. Integrity impact is low because the attacker might be able to modify some data, but the attacker does not necessarily have control over the amount of modification. Availability impact is low since buffer overflow might cause the game to crash and have reduced performance, but the attacker does not have the ability to completely deny service to legitimate users.



CVSS Base Score: 7.1
Impact Subscore: 3.7
Exploitability Subscore: 2.8
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 7.1

CVSS v3.1 Vector
AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:L

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*
[✓ Network (AV:N)] Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*
[✓ Low (AC:L)] High (AC:H)

Privileges Required (PR)*
[✓ None (PR:N)] Low (PR:L) High (PR:H)

User Interaction (UI)*
None (UI:N) [✓ Required (UI:R)]

Scope (S)*
Unchanged (S:U) [✓ Changed (S:C)]

Impact Metrics

Confidentiality Impact (C)*
None (C:N) [✓ Low (C:L)] High (C:H)

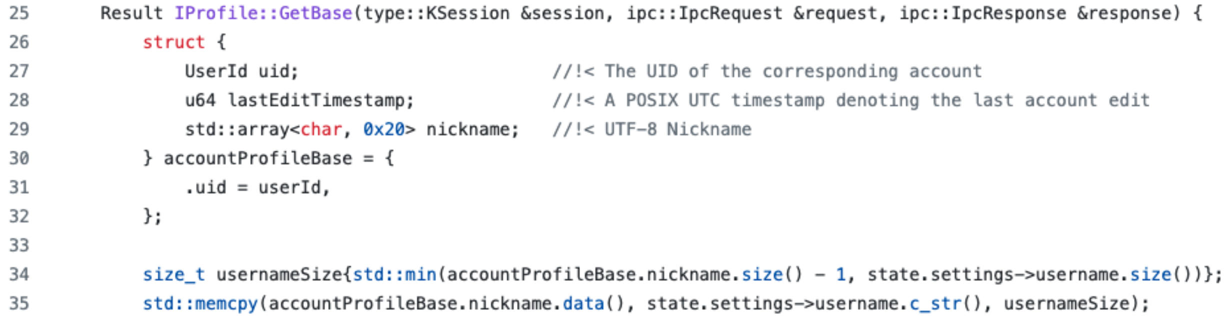
Integrity Impact (I)*
None (I:N) [✓ Low (I:L)] High (I:H)

Availability Impact (A)*
None (A:N) [✓ Low (A:L)] High (A:H)

* - All base metrics are required to generate a base score.

10. /app/src/main/cpp/skyline/services/account/IProfile.cpp:35: [2] (buffer) memcpy:
Does not check for buffer overflows when copying to destination (CWE-120).
Make sure destination can always hold the source data.

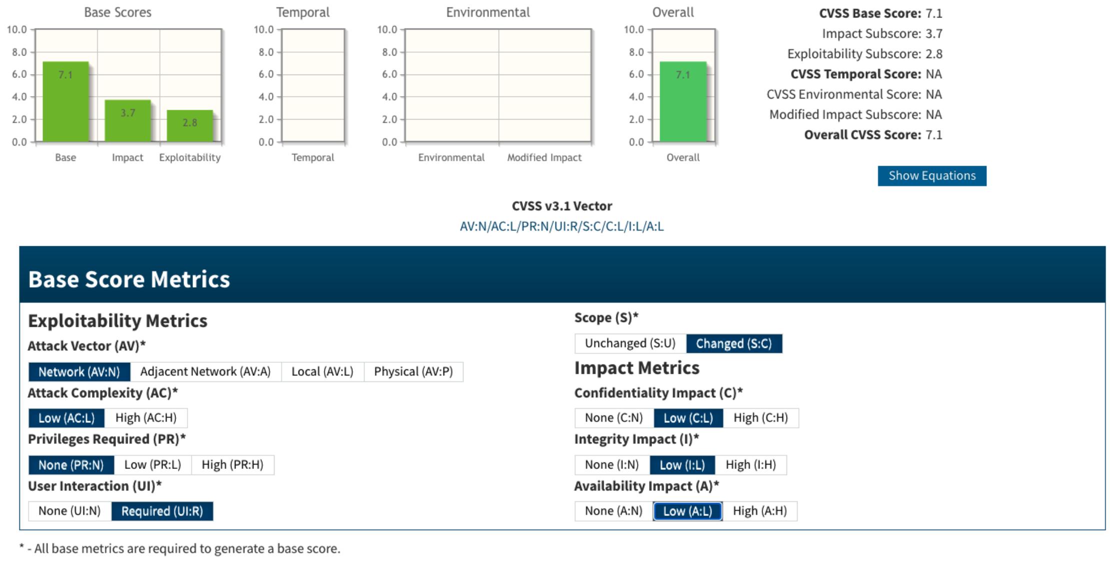
- **Vulnerability Description:** The vulnerability is a buffer overflow. Without verifying the sizes of the input and output buffers, the program copies the input buffer to the output buffer. This leads to a buffer overflow and this overflow can corrupt adjacent storage. Such a vulnerability can cause the program to crash and it creates entry points for attacks.
- **Categories:** Spatial Memory Attacks: Buffer Overflow, Out-of-Bound Access
- **Source Code Snippet:**



```
25  Result IProfile::GetBase(type::KSession &session, ipc::IpcRequest &request, ipc::IpcResponse &response) {
26      struct {
27          UserId uid;                        //!< The UID of the corresponding account
28          u64 lastEditTimestamp;             //!< A POSIX UTC timestamp denoting the last account edit
29          std::array<char, 0x20> nickname;   //!< UTF-8 Nickname
30      } accountProfileBase = {
31          .uid = userId,
32      };
33
34      size_t usernameSize{std::min(accountProfileBase.nickname.size() - 1, state.settings->username.size())};
35      std::memcpy(accountProfileBase.nickname.data(), state.settings->username.c_str(), usernameSize);
```

In this program, the command memcpy in line 35 copies data from the source (state.settings->username.c_str) to the destination (accountProfileBase.nickname). The code is not checking for buffer overflow in case the size of the source exceeds that of the destination.

- **Discovery Process:** We ran the FlawFinder tool on the git rep of this software in Ubuntu.
- **Assessment of the severity and impact:** The overall CVSS score is 7.1, which is somewhat high. This vulnerability impacts confidentiality, integrity and availability. Confidentiality impact is low since the vulnerability might provide access to some restricted information but the attacker does not have control over the information obtained. Integrity impact is low because the attacker might be able to modify some data, but the attacker does not necessarily have control over the amount of modification. Availability impact is low since buffer overflow might cause the game to crash and have reduced performance, but the attacker does not have the ability to completely deny service to legitimate users.



CVSS Base Score: 7.1
Impact Subscore: 3.7
Exploitability Subscore: 2.8
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 7.1

CVSS v3.1 Vector
AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:L

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*

[✓ Network (AV:N)] Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*

[✓ Low (AC:L)] High (AC:H)

Privileges Required (PR)*

[✓ None (PR:N)] Low (PR:L) High (PR:H)

User Interaction (UI)*

None (UI:N) [✓ Required (UI:R)]

Scope (S)*

Unchanged (S:U) [✓ Changed (S:C)]

Impact Metrics

Confidentiality Impact (C)*

None (C:N) [✓ Low (C:L)] High (C:H)

Integrity Impact (I)*

None (I:N) [✓ Low (I:L)] High (I:H)

Availability Impact (A)*

None (A:N) [✓ Low (A:L)] High (A:H)

* - All base metrics are required to generate a base score.