Project 1 - Spot the Holes

CSCE 713 Software Security

Peiwen Wang (128001139)

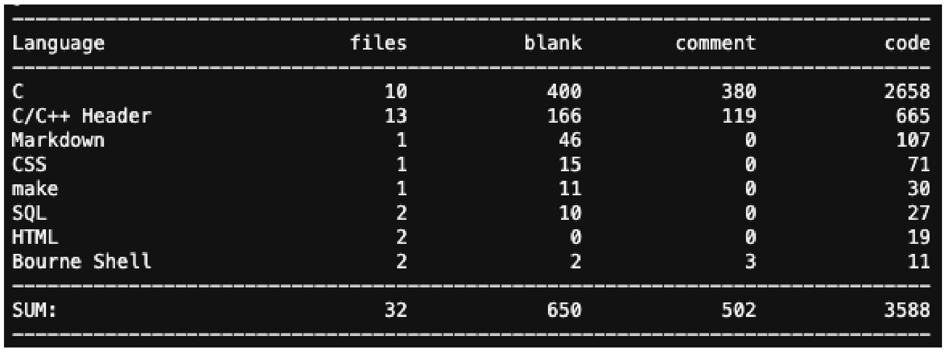
peiwen.wang@tamu.edu

Jingdi Zhang (825007281)

cindy1024i@tamu.edu

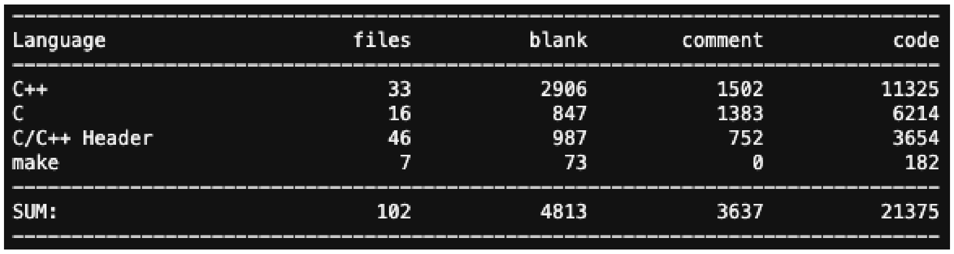
Two software targets - (over 10,000 lines of code in total)

Sparrow: https://github.com/codercheng/sparrow



```
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
C                               10            400            380           2658
C/C++ Header                    13            166            119            665
Markdown                         1             46              0            107
CSS                              1             15              0             71
make                             1             11              0             30
SQL                              2             10              0             27
HTML                             2              0              0             19
Bourne Shell                     2              2              3             11
-------------------------------------------------------------------------------
SUM:                            32            650            502           3588
-------------------------------------------------------------------------------
```

Whisper: https://github.com/syllable-org/whisper.git



```
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
C++                             33           2906           1502          11325
C                               16            847           1383           6214
C/C++ Header                    46            987            752           3654
make                             7             73              0            182
-------------------------------------------------------------------------------
SUM:                           102           4813           3637          21375
-------------------------------------------------------------------------------
```

Attacks Categories Covered in the report:

Spatial Memory Attacks

Control Flow Attacks

ConcurrencyAttacks

Program 1 Sparrow (C)

https://github.com/codercheng/sparrow

Vulnerability 1.1

=> Vulnerability description: Buffer overflow. The developer used the **sprintf** to store the directory information into the buffer with 512 B. But the developer did not check if the length of the directory entries exceeds 512 B or not. The attacker can make use of this to corrupt the next buffer or program code to run other evil programs.

=> Categories: **Spatial Memory Attacks**
=> Source code snippet

```
Line 106     char newpath[512];
...

Line 147     while ((temp_path = readdir(dir)) != NULL) {

             if (!strcmp(temp_path->d_name, ".") || !strcmp(temp_path->d_name,
"..") || !strcmp(temp_path->d_name, ".res"))
                     continue;

             if (path[strlen(path) - 1] == '/')
Line 153           sprintf(newpath, "%s%s", path, temp_path->d_name);
             else
Line 155           sprintf(newpath, "%s/%s", path, temp_path->d_name);

             lstat(newpath, &s);
```

As we can see from the source code in file.c file, the buffer, newpath, with length of 512 B is defined in Line 106. And then in Line 153 or 155, the temp_path->d_name is stored in this buffer. temp_path is the defined as: struct dirent *temp_path, and dirent has the member, d_name, character array which is of unspecified size, but the number of bytes preceding the terminating null byte will not exceed {NAME_MAX}. But the developer did not include the limits.h to declare the NAME_MAX. Thus, the buffer overflow can happen when the dir is too long, and thus the software can be attacked.

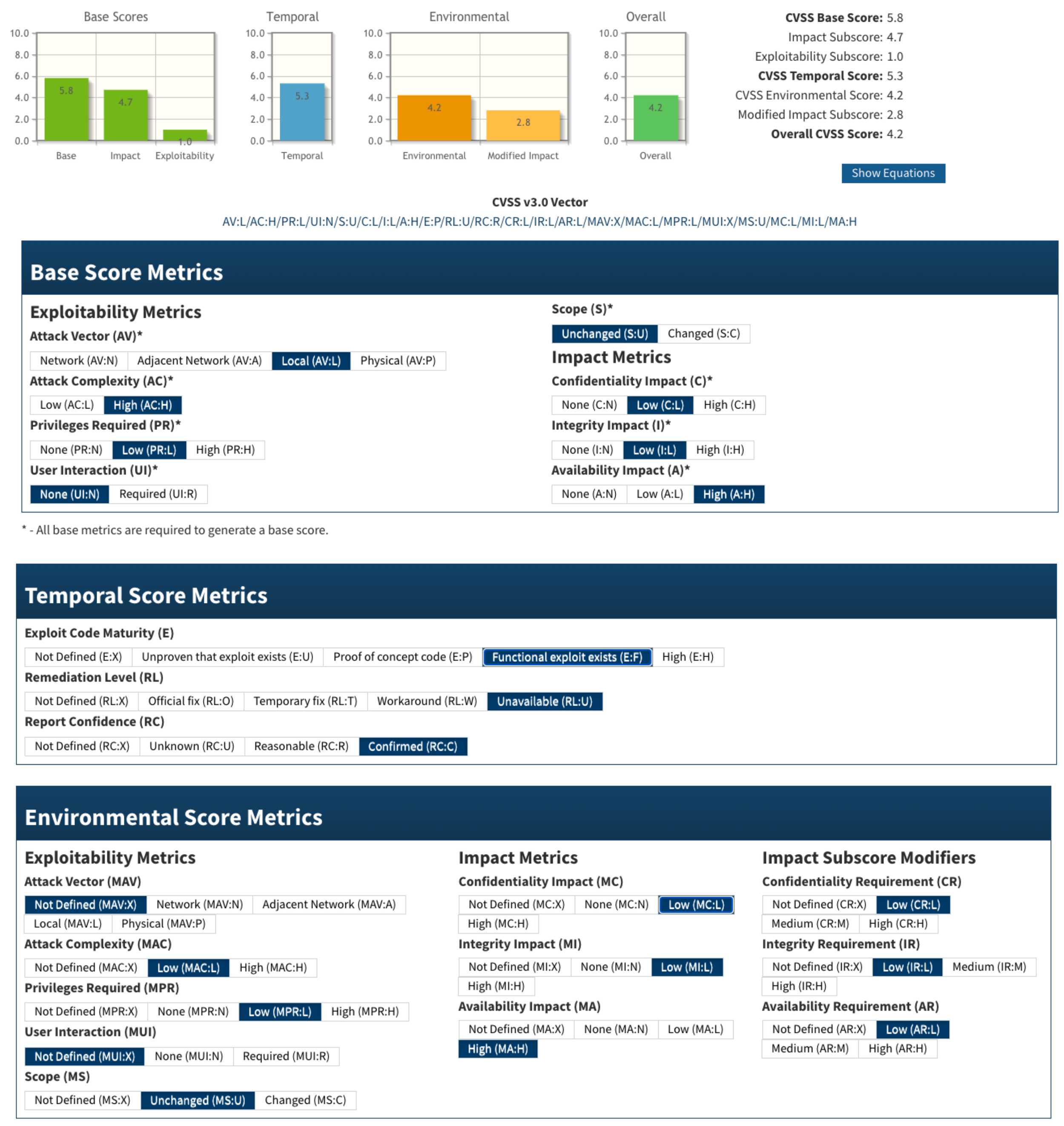
=> Discovery process: FlawFinder. Command: flawfinder sparrow

```
sparrow/file.c:106:  [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.
```

sparrow/file.c:153: [4] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  Vsnprintf.

This is not a False Positive, since the dir length is not checked before it is stored into the buffer with a fixed size.

=> Assessment:



CVSS Base Score: 5.8
Impact Subscore: 4.7
Exploitability Subscore: 1.0
CVSS Temporal Score: 5.3
CVSS Environmental Score: 4.2
Modified Impact Subscore: 2.8
Overall CVSS Score: 4.2

CVSS v3.0 Vector
AV:L/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:H/E:P/RL:U/RC:R/CR:L/IR:L/AR:L/MAV:X/MAC:L/MPR:L/MUI:X/MS:U/MC:L/MI:L/MA:H

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.

Temporal Score Metrics

Exploit Code Maturity (E)
Not Defined (E:X) | Unproven that exploit exists (E:U) | Proof of concept code (E:P) | [✓ Functional exploit exists (E:F)] | High (E:H)

Remediation Level (RL)
Not Defined (RL:X) | Official fix (RL:O) | Temporary fix (RL:T) | Workaround (RL:W) | [✓ Unavailable (RL:U)]

Report Confidence (RC)
Not Defined (RC:X) | Unknown (RC:U) | Reasonable (RC:R) | [✓ Confirmed (RC:C)]

Environmental Score Metrics

Exploitability Metrics

Attack Vector (MAV)
[✓ Not Defined (MAV:X)] | Network (MAV:N) | Adjacent Network (MAV:A) | Local (MAV:L) | Physical (MAV:P)

Attack Complexity (MAC)
Not Defined (MAC:X) | [✓ Low (MAC:L)] | High (MAC:H)

Privileges Required (MPR)
Not Defined (MPR:X) | None (MPR:N) | [✓ Low (MPR:L)] | High (MPR:H)

User Interaction (MUI)
[✓ Not Defined (MUI:X)] | None (MUI:N) | Required (MUI:R)

Scope (MS)
Not Defined (MS:X) | [✓ Unchanged (MS:U)] | Changed (MS:C)

Impact Metrics

Confidentiality Impact (MC)
Not Defined (MC:X) | None (MC:N) | [✓ Low (MC:L)] | High (MC:H)

Integrity Impact (MI)
Not Defined (MI:X) | None (MI:N) | [✓ Low (MI:L)] | High (MI:H)

Availability Impact (MA)
Not Defined (MA:X) | None (MA:N) | Low (MA:L) | [✓ High (MA:H)]

Impact Subscore Modifiers

Confidentiality Requirement (CR)
Not Defined (CR:X) | [✓ Low (CR:L)] | Medium (CR:M) | High (CR:H)

Integrity Requirement (IR)
Not Defined (IR:X) | [✓ Low (IR:L)] | Medium (IR:M) | High (IR:H)

Availability Requirement (AR)
Not Defined (AR:X) | [✓ Low (AR:L)] | Medium (AR:M) | High (AR:H)

CWE-120

1. Severity: Some of the vulnerabilities caused by buffer overflow can be severe, for example, in this case, there is no special privilege and the attack complexity is not that high, the directory can be manipulated by the attackers and make it super long and direct to other programs they want.

2. Impact: Buffer overflow generally causes the software to crash, and in some cases, it can also put the software into an infinite loop to prevent it from functioning well.

Vulnerability 1.2

=> Vulnerability description: Buffer overflow. The developer again used the **sprintf** to store the path information into the buffer. Developer indeed checked via “pos + 256 > max_len_limit” in the for loop, but for the last iteration, the developer did not check if the size of the path info will exceed 256 B or not.

=> Categories: **Spatial Memory Attacks**
=> Source code snippet

```
Line 144   item_t * items = (item_t *)malloc(max_item_num * sizeof(item_t));
...

Line 182   int i;
       for (i = 0; i<item_cnt; i++) {
              if (pos + 256 > max_len_limit)
                     break;
              if (items[i].dir) {
Line 187             ret = sprintf(buf + pos, "<div class=\"dir\"><a
href=\"%s%s/\"><img src=\"/.res/dir.png\">&nbsp;%s</a></div>\n", \
                            prefix, items[i].path, items[i].path);
                     pos += ret;
              }
              else {
Line 192             ret = sprintf(buf + pos, "<div class=\"file\"><a
href=\"%s%s\"><img src=\"/.res/file.ico\">&nbsp;%s</a></div>\n", \
                            prefix, items[i].path, items[i].path);
                     pos += ret;
              }
       }

// item_t is defined as below in the file.H
typedef struct {
              char path[256];
              time_t m_time;//modify time
              int dir;
```

```
            int size;
    } item_t;
```

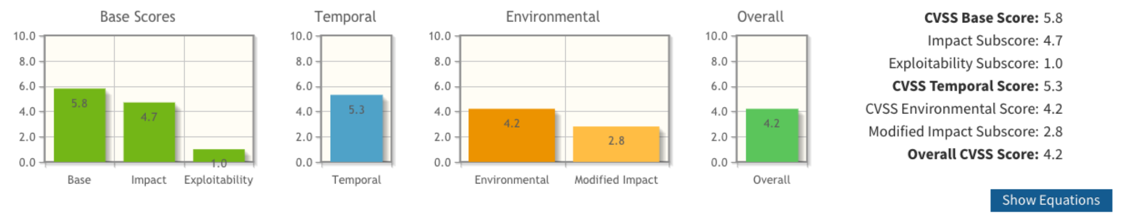
As the source code shows, the member, path, of the item_t has size of 256 B. In Line 187 or 192, the developer wants to store the path information to the buf. In the last iteration of the while loop, even though pos + 256 <= max_len_limit of the buffer, but two paths are written to the buffer. Since max length of the path of item[i] is 256, we can not guarantee that the content stored into the buffer is shorter than 256 B, thus an attacker can make use of this to achieve the buffer overflow, and exploit code.

=> Discovery process: FlawFinder. Command: flawfinder sparrow

```
sparrow/file.c:187:  [4] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf.
sparrow/file.c:192:  [4] (buffer) sprintf:
  Does not check for buffer overflows (CWE-120). Use sprintf_s, snprintf, or
  vsnprintf.
```

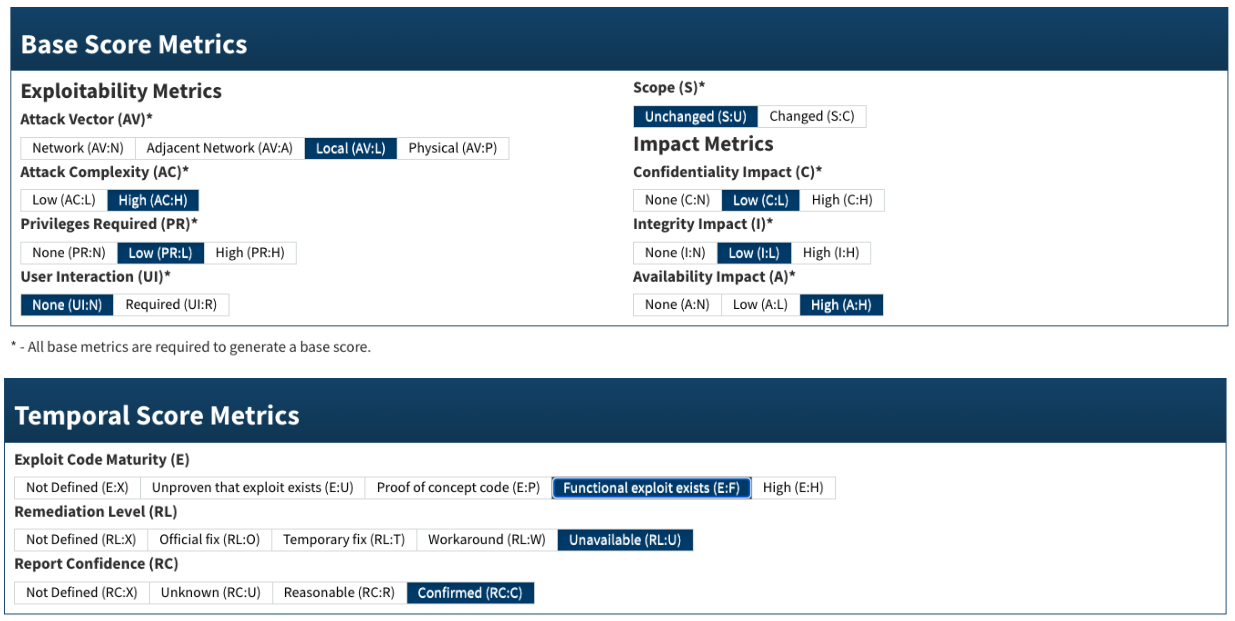
This is not a False Positive, since the checking of buffer length is not strong enough, to make sure the last iteration in the while loop will not overflow.

=> Assessment:



CVSS Base Score: 5.8
Impact Subscore: 4.7
Exploitability Subscore: 1.0
CVSS Temporal Score: 5.3
CVSS Environmental Score: 4.2
Modified Impact Subscore: 2.8
Overall CVSS Score: 4.2

CVSS v3.0 Vector
AV:L/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:H/E:P/RL:U/RC:R/CR:L/IR:L/AR:L/MAV:X/MAC:L/MPR:L/MUI:X/MS:U/MC:L/MI:L/MA:H



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)]

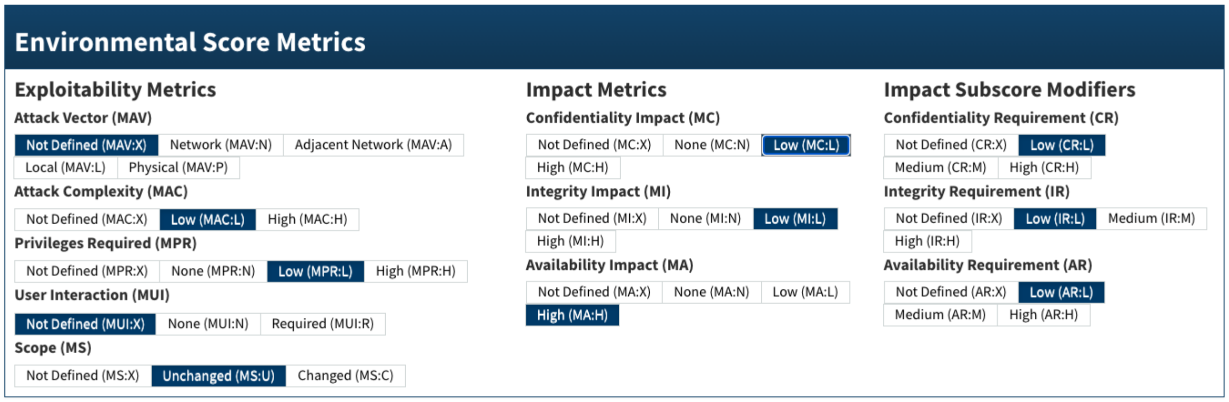
* - All base metrics are required to generate a base score.

Temporal Score Metrics

Exploit Code Maturity (E)
Not Defined (E:X) | Unproven that exploit exists (E:U) | Proof of concept code (E:P) | [✓ Functional exploit exists (E:F)] | High (E:H)

Remediation Level (RL)
Not Defined (RL:X) | Official fix (RL:O) | Temporary fix (RL:T) | Workaround (RL:W) | [✓ Unavailable (RL:U)]

Report Confidence (RC)
Not Defined (RC:X) | Unknown (RC:U) | Reasonable (RC:R) | [✓ Confirmed (RC:C)]



Environmental Score Metrics

Exploitability Metrics

Attack Vector (MAV)

[✓ Not Defined (MAV:X)] Network (MAV:N) Adjacent Network (MAV:A) Local (MAV:L) Physical (MAV:P)

Attack Complexity (MAC)

Not Defined (MAC:X) [✓ Low (MAC:L)] High (MAC:H)

Privileges Required (MPR)

Not Defined (MPR:X) None (MPR:N) [✓ Low (MPR:L)] High (MPR:H)

User Interaction (MUI)

[✓ Not Defined (MUI:X)] None (MUI:N) Required (MUI:R)

Scope (MS)

Not Defined (MS:X) [✓ Unchanged (MS:U)] Changed (MS:C)

Impact Metrics

Confidentiality Impact (MC)

Not Defined (MC:X) None (MC:N) [✓ Low (MC:L)] High (MC:H)

Integrity Impact (MI)

Not Defined (MI:X) None (MI:N) [✓ Low (MI:L)] High (MI:H)

Availability Impact (MA)

Not Defined (MA:X) None (MA:N) Low (MA:L) [✓ High (MA:H)]

Impact Subscore Modifiers

Confidentiality Requirement (CR)

Not Defined (CR:X) [✓ Low (CR:L)] Medium (CR:M) High (CR:H)

Integrity Requirement (IR)

Not Defined (IR:X) [✓ Low (IR:L)] Medium (IR:M) High (IR:H)

Availability Requirement (AR)

Not Defined (AR:X) [✓ Low (AR:L)] Medium (AR:M) High (AR:H)

Vulnerability 1.3

=> Vulnerability description: Buffer overflow. The developer used **strcpy** to copy the filename to the member of the struct fd_record_t, but didn't make sure the size of the filename is short enough to be stored to the destination to prevent overflow.
=> Categories: **Spatial Memory Attacks**
=> Source code snippet

```
Line 355    char filename[1024 + 1 + strlen(work_dir)];//full path
...

Line 454     if (fd_records[sock].http_code != 304) {

                 //fd_records[sock].ffd = fd;
                 fd_records[sock].read_pos = 0;

                 if (fd_records[sock].http_code != DIR_CODE) {
                       fd_records[sock].total_len = (int)filestat.st_size;
                       setnonblocking(fd);
                 }
Line 463        strcpy(fd_records[sock].path, filename);
               ...
            }
// fd_record_t is defined as below in the ev_loop.h
      typedef struct {
            int active;

            EV_TYPE events;
            cb_func_t cb_read;
            cb_func_t cb_write;

            int ffd;
            unsigned int write_pos;
            unsigned int read_pos;
            unsigned int total_len;
            char buf[MAXBUFSIZE];
            int http_code;
            char path[256];
            int keep_alive;

            void* timer_ptr;
        } fd_record_t;
```

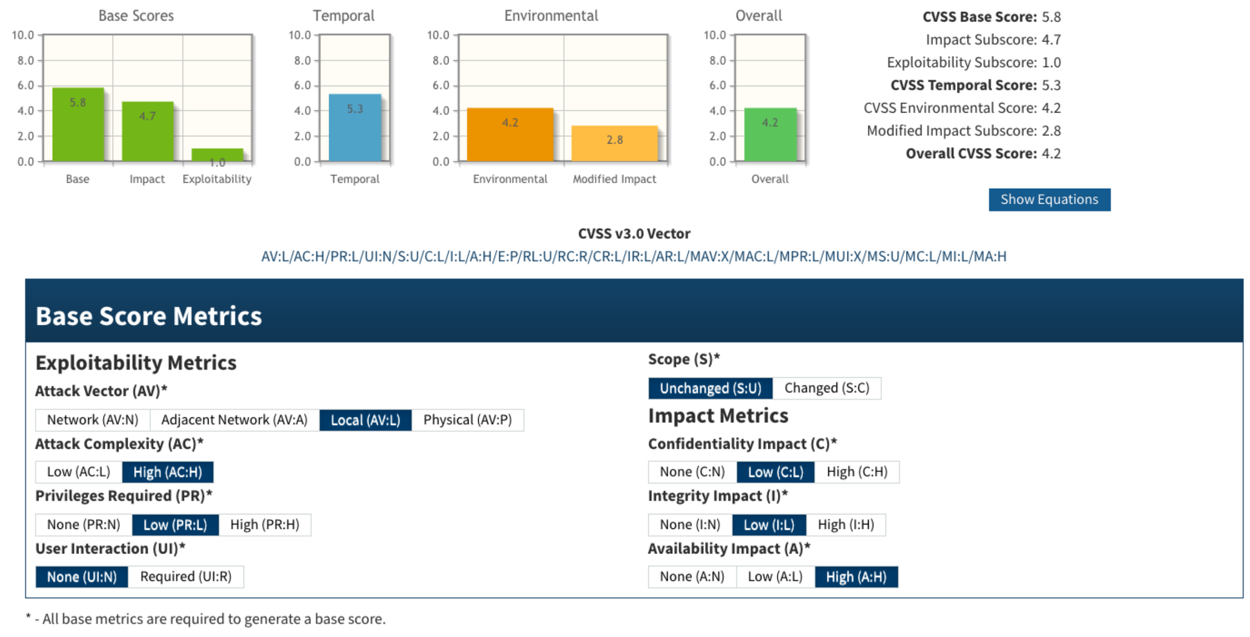
According to the definition of char * strcpy (char * destination, const char * source), the size of the destination file should be long enough, but here, the destination file is the member, path, from fd_record_t, which only has 256 B length. However, the source file, filename (in Line 355) can be longer than 1025 B. So the function strcpy in Line 463 can cause buffer overflow.

=> Discovery process: FlawFinder. Command: flawfinder sparrow

sparrow/sparrow.c:463: [4] (buffer) strcpy:

Does not check for buffer overflows when copying to destination (CWE-120). Consider using strcpy_s, strncpy, or strlcpy (warning, strncpy is easily misused).

=> Assessment:



CVSS Base Score: 5.8
Impact Subscore: 4.7
Exploitability Subscore: 1.0
CVSS Temporal Score: 5.3
CVSS Environmental Score: 4.2
Modified Impact Subscore: 2.8
Overall CVSS Score: 4.2

CVSS v3.0 Vector
AV:L/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:H/E:P/RL:U/RC:R/CR:L/IR:L/AR:L/MAV:X/MAC:L/MPR:L/MUI:X/MS:U/MC:L/MI:L/MA:H

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)

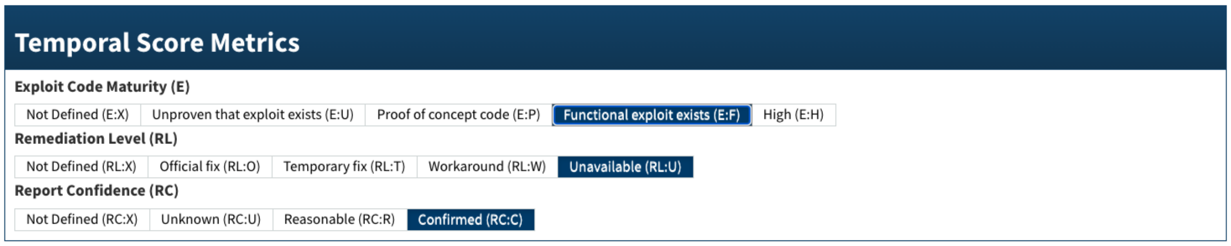
Impact Metrics

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.



Temporal Score Metrics

Exploit Code Maturity (E)

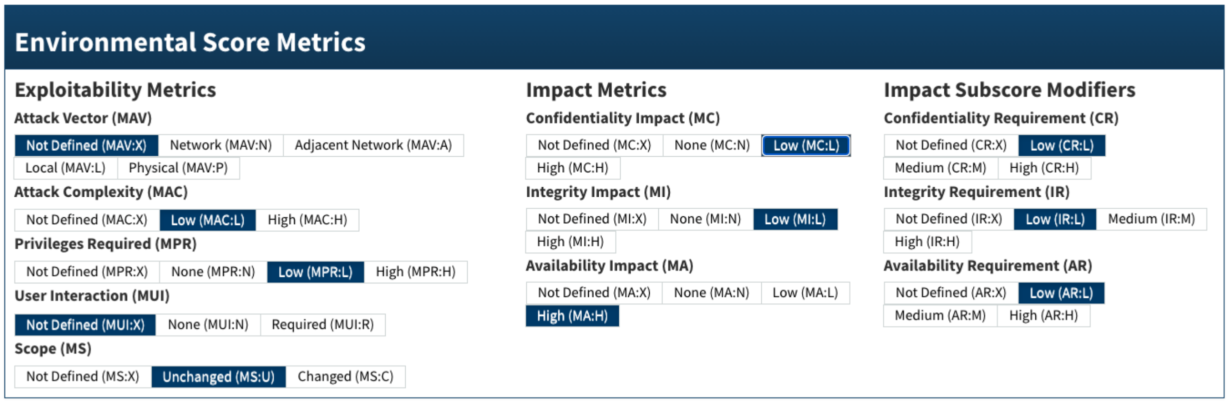
Not Defined (E:X) | Unproven that exploit exists (E:U) | Proof of concept code (E:P) | [✓ Functional exploit exists (E:F)] | High (E:H)

Remediation Level (RL)

Not Defined (RL:X) | Official fix (RL:O) | Temporary fix (RL:T) | Workaround (RL:W) | [✓ Unavailable (RL:U)]

Report Confidence (RC)

Not Defined (RC:X) | Unknown (RC:U) | Reasonable (RC:R) | [✓ Confirmed (RC:C)]



Environmental Score Metrics

Exploitability Metrics

Attack Vector (MAV)

[✓ Not Defined (MAV:X)] | Network (MAV:N) | Adjacent Network (MAV:A) | Local (MAV:L) | Physical (MAV:P)

Attack Complexity (MAC)

Not Defined (MAC:X) | [✓ Low (MAC:L)] | High (MAC:H)

Privileges Required (MPR)

Not Defined (MPR:X) | None (MPR:N) | [✓ Low (MPR:L)] | High (MPR:H)

User Interaction (MUI)

[✓ Not Defined (MUI:X)] | None (MUI:N) | Required (MUI:R)

Scope (MS)

Not Defined (MS:X) | [✓ Unchanged (MS:U)] | Changed (MS:C)

Impact Metrics

Confidentiality Impact (MC)

Not Defined (MC:X) | None (MC:N) | [✓ Low (MC:L)] | High (MC:H)

Integrity Impact (MI)

Not Defined (MI:X) | None (MI:N) | [✓ Low (MI:L)] | High (MI:H)

Availability Impact (MA)

Not Defined (MA:X) | None (MA:N) | Low (MA:L) | [✓ High (MA:H)]

Impact Subscore Modifiers

Confidentiality Requirement (CR)

Not Defined (CR:X) | [✓ Low (CR:L)] | Medium (CR:M) | High (CR:H)

Integrity Requirement (IR)

Not Defined (IR:X) | [✓ Low (IR:L)] | Medium (IR:M) | High (IR:H)

Availability Requirement (AR)

Not Defined (AR:X) | [✓ Low (AR:L)] | Medium (AR:M) | High (AR:H)

Vulnerability 1.4

=> Vulnerability description: The vulnerability is time-to-check-to-time-to-use. After using access() system call to check if there is permission to “log”, but before actually opening the file, if an attacker to use a symbolic link to let the “log” refer to a different file, e.g. a “secret” file includes more sensitive data instead of just the log information. And then the attacker is able to set the 755 permissions to this “secret” file, thus the group and other users could have the read and execution permissions to the sensitive data in the “secret” file.

=> Categories: **Concurrency Attacks**
=> Source code snippet

```
Line 246     //sprintf(buf,"log/");
        if (access("log", F_OK) != 0) {
                if (mkdir("log", 0755) < 0) {
                        fprintf(stderr, "create log folder error\n");
                }
        }
```

Just as explained above, the time-to-check is achieved by the access() system call. This software indeed can be in multi-thread mode, so if an attacker uses symbolic link to trick the software to set 755 permission to another “secret” file instead of “log”. The confidentiality of the software has been violated.

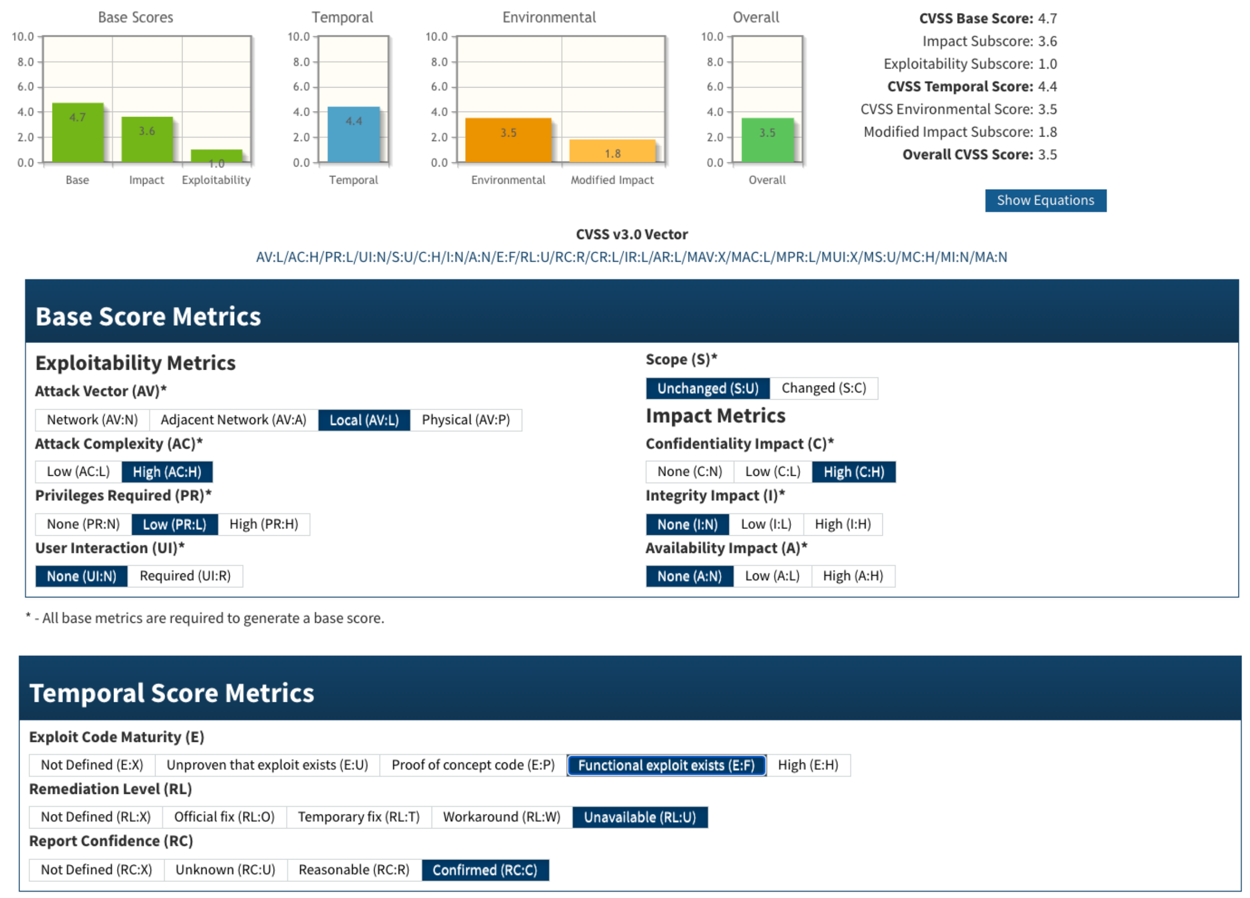
---------- 0000 no permissions
---x--x--x 0111 execute
--w--w--w- 0222 write
--wx-wx-wx 0333 write & execute
-r--r--r-- 0444 read
-r-xr-xr-x 0555 read & execute
-rw-rw-rw- 0666 read & write -rwxrwxrwx
0777 read. write & execute

Thus, for 755, **-rwxr-xr-x**, owner has full permission, and group and other users have read and execution permissions.

=> Discovery process: Flawfinder - static analysis tool. Command: <u>flawfinder sparrow</u>

```
sparrow/async_log.c:246:  [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.
```

=> Assessment: CWE-367: Time-of-check Time-of-use (TOCTOU)



CVSS Base Score: 4.7
Impact Subscore: 3.6
Exploitability Subscore: 1.0
CVSS Temporal Score: 4.4
CVSS Environmental Score: 3.5
Modified Impact Subscore: 1.8
Overall CVSS Score: 3.5

CVSS v3.0 Vector
AV:L/AC:H/PR:L/UI:N/S:U/C:H/I:N/A:N/E:F/RL:U/RC:R/CR:L/IR:L/AR:L/MAV:X/MAC:L/MPR:L/MUI:X/MS:U/MC:H/MI:N/MA:N

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)

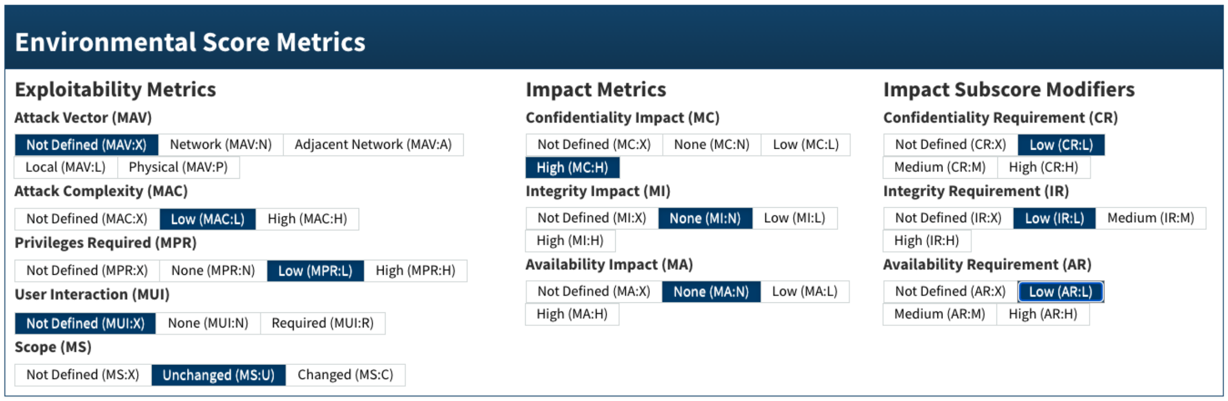
* - All base metrics are required to generate a base score.

Temporal Score Metrics

Exploit Code Maturity (E)
Not Defined (E:X) Unproven that exploit exists (E:U) Proof of concept code (E:P) [✓ Functional exploit exists (E:F)] High (E:H)

Remediation Level (RL)
Not Defined (RL:X) Official fix (RL:O) Temporary fix (RL:T) Workaround (RL:W) [✓ Unavailable (RL:U)]

Report Confidence (RC)
Not Defined (RC:X) Unknown (RC:U) Reasonable (RC:R) [✓ Confirmed (RC:C)]


Environmental Score Metrics

Exploitability Metrics

Attack Vector (MAV)
[✓ Not Defined (MAV:X)] Network (MAV:N) Adjacent Network (MAV:A) Local (MAV:L) Physical (MAV:P)

Attack Complexity (MAC)
Not Defined (MAC:X) [✓ Low (MAC:L)] High (MAC:H)

Privileges Required (MPR)
Not Defined (MPR:X) None (MPR:N) [✓ Low (MPR:L)] High (MPR:H)

User Interaction (MUI)
[✓ Not Defined (MUI:X)] None (MUI:N) Required (MUI:R)

Scope (MS)
Not Defined (MS:X) [✓ Unchanged (MS:U)] Changed (MS:C)

Impact Metrics

Confidentiality Impact (MC)
Not Defined (MC:X) None (MC:N) Low (MC:L) [✓ High (MC:H)]

Integrity Impact (MI)
Not Defined (MI:X) [✓ None (MI:N)] Low (MI:L) High (MI:H)

Availability Impact (MA)
Not Defined (MA:X) [✓ None (MA:N)] Low (MA:L) High (MA:H)

Impact Subscore Modifiers

Confidentiality Requirement (CR)
Not Defined (CR:X) [✓ Low (CR:L)] Medium (CR:M) High (CR:H)

Integrity Requirement (IR)
Not Defined (IR:X) [✓ Low (IR:L)] Medium (IR:M) High (IR:H)

Availability Requirement (AR)
Not Defined (AR:X) [✓ Low (AR:L)] Medium (AR:M) High (AR:H)

Vulnerability 1.5

=> Vulnerability description: time-to-check-to-time-to-use. Before opening the file, an attacker can use a symbolic link to track the software to open another “secret” file which will lead to the information leak.
=> Categories: **Concurrency Attacks**
=> Source code snippet:

```
                 if (fd_records[sock].http_code != 304) {
Line 431                 fd = open(filename, O_RDONLY);

                           if (fd == -1) {
                                  if (conf.log_enable) {
                                         log_error("can not open file:%s\n",
filename);
                                  }
                                  else {
                                         fprintf(stderr, "can not open file:%s,
because:%s\n", filename, strerror(errno));
                                  }
                                  safe_close(loop, sock);
                                  return NULL;
                           }

                           fd_records[sock].ffd = fd;
                    }
```

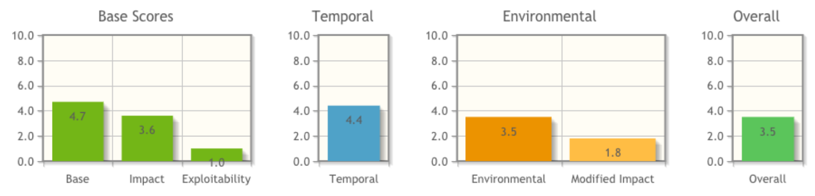
As we can see from the code, there is no further check when opening the file. So the filename can be maliciously referred to other files. So the “openat” command should be used to make sure that the file is in the directory the developer wants to open.

=> Discovery process: Flawfinder - static analysis tool. Command: flawfinder sparrow

sparrow/sparrow.c:431: [2] (misc) open:

Check when opening files - can an attacker redirect it (via symlinks), force the opening of special file type (e.g., device files), move things around to create a race condition, control its ancestors, or change its contents? (CWE-362).

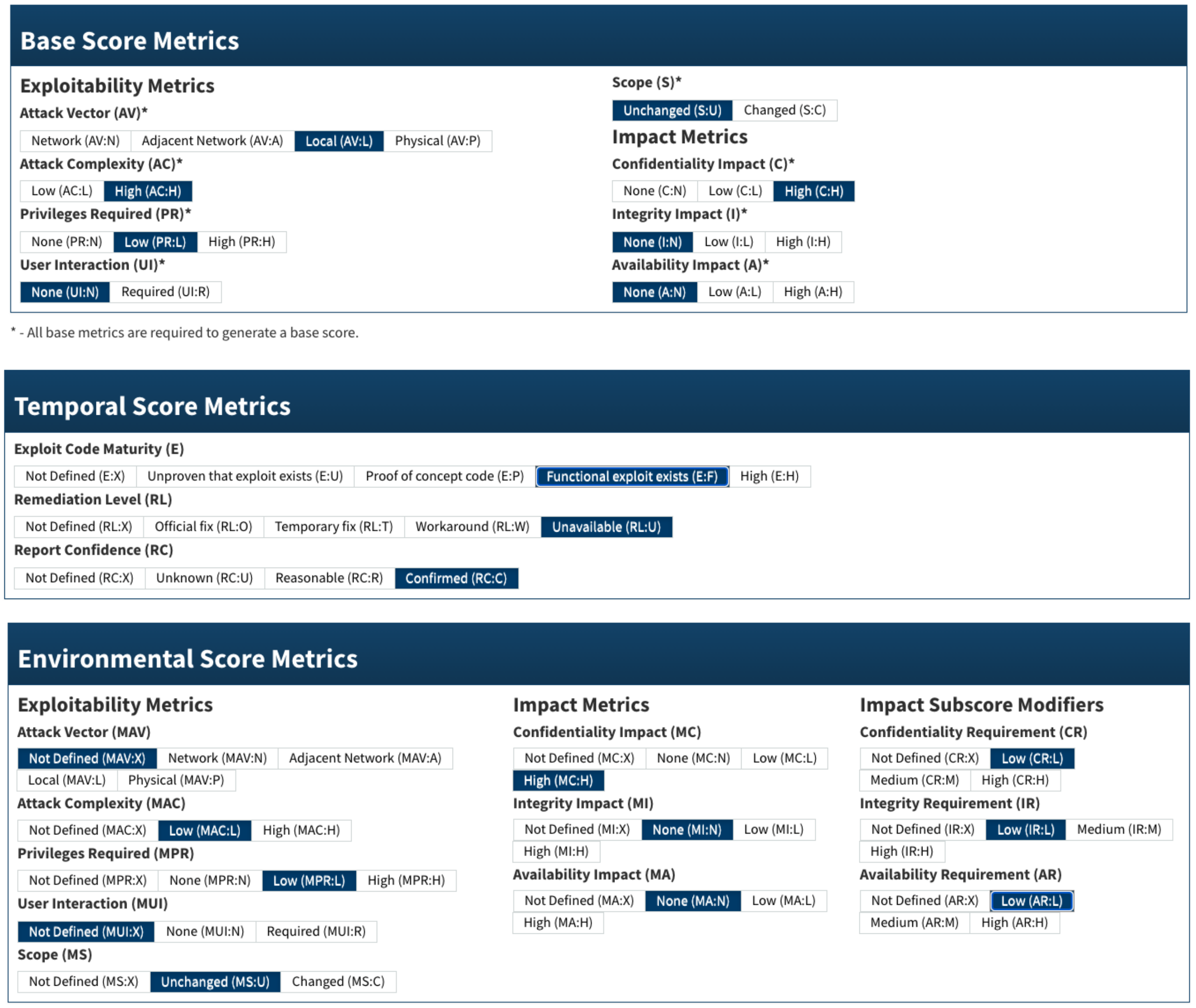
=> Assessment: CWE-367: Time-of-check Time-of-use (TOCTOU)



CVSS Base Score: 4.7
Impact Subscore: 3.6
Exploitability Subscore: 1.0
CVSS Temporal Score: 4.4
CVSS Environmental Score: 3.5
Modified Impact Subscore: 1.8
Overall CVSS Score: 3.5

Show Equations

CVSS v3.0 Vector
AV:L/AC:H/PR:L/UI:N/S:U/C:H/I:N/A:N/E:F/RL:U/RC:R/CR:L/IR:L/AR:L/MAV:X/MAC:L/MPR:L/MUI:X/MS:U/MC:H/MI:N/MA:N



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.

Temporal Score Metrics

Exploit Code Maturity (E)
Not Defined (E:X) | Unproven that exploit exists (E:U) | Proof of concept code (E:P) | [✓ Functional exploit exists (E:F)] | High (E:H)

Remediation Level (RL)
Not Defined (RL:X) | Official fix (RL:O) | Temporary fix (RL:T) | Workaround (RL:W) | [✓ Unavailable (RL:U)]

Report Confidence (RC)
Not Defined (RC:X) | Unknown (RC:U) | Reasonable (RC:R) | [✓ Confirmed (RC:C)]

Environmental Score Metrics

Exploitability Metrics

Attack Vector (MAV)
[✓ Not Defined (MAV:X)] | Network (MAV:N) | Adjacent Network (MAV:A) | Local (MAV:L) | Physical (MAV:P)

Attack Complexity (MAC)
Not Defined (MAC:X) | [✓ Low (MAC:L)] | High (MAC:H)

Privileges Required (MPR)
Not Defined (MPR:X) | None (MPR:N) | [✓ Low (MPR:L)] | High (MPR:H)

User Interaction (MUI)
[✓ Not Defined (MUI:X)] | None (MUI:N) | Required (MUI:R)

Scope (MS)
Not Defined (MS:X) | [✓ Unchanged (MS:U)] | Changed (MS:C)

Impact Metrics

Confidentiality Impact (MC)
Not Defined (MC:X) | None (MC:N) | Low (MC:L) | [✓ High (MC:H)]

Integrity Impact (MI)
Not Defined (MI:X) | [✓ None (MI:N)] | Low (MI:L) | High (MI:H)

Availability Impact (MA)
Not Defined (MA:X) | [✓ None (MA:N)] | Low (MA:L) | High (MA:H)

Impact Subscore Modifiers

Confidentiality Requirement (CR)
Not Defined (CR:X) | [✓ Low (CR:L)] | Medium (CR:M) | High (CR:H)

Integrity Requirement (IR)
Not Defined (IR:X) | [✓ Low (IR:L)] | Medium (IR:M) | High (IR:H)

Availability Requirement (AR)
Not Defined (AR:X) | [✓ Low (AR:L)] | Medium (AR:M) | High (AR:H)

Vulnerability 1.6

=> Vulnerability description: Use of Uninitialized Variable. The vulnerability is due to the member, epfd, of the loop being not initialized before it is used in the epoll_ctl. And the epfd is epoll file descriptor. If the epfd is not initialized, and possibly contains junk data, and thus an attacker can take control of these contents, to manipulate the events in the programm. Thus, this vulnerability may modify the control flow and enable code execution as the attacker wants.
=> Categories: **Control Flow Attacks**
=> Source code snippet:

```
Line 162    int ev_stop(ev_loop_t *loop, int fd, EV_TYPE events) {
        /*fd in use, and evnets on fd*/
        if (/*fd_records[fd].active &&*/ (fd_records[fd].events & events)) {
                if ((fd_records[fd].events & EV_READ) && (events & EV_READ)) {
                        fd_records[fd].events &= (~EV_READ);
                }
                if ((fd_records[fd].events & EV_WRITE) && (events & EV_WRITE)) {
                        fd_records[fd].events &= (~EV_WRITE);
                }

                struct epoll_event ev;
                ev.events = fd_records[fd].events;
                if (loop->etmodel) {
                        ev.events |= EPOLLET;
                }
                ev.data.fd = fd;

Line 179        if (-1 == epoll_ctl(loop->epfd, EPOLL_CTL_MOD, fd, &ev)) {
                        fprintf(stderr, "epoll_ctl mod in ev_stop\n");
                        ev_clear(fd);
                        return -1;
                }

                // if(!(fd_records[fd].events & EV_READ || fd_records[fd].events
&EV_WRITE)) {
                //      //printf("here...\n");
                //      //return ev_unregister(loop, fd);
                // }
                return 0;
        }
        return 0;
}

// ev_loop_t is defined as below in the ev_loop.h
typedef struct ev_loop_t{
```

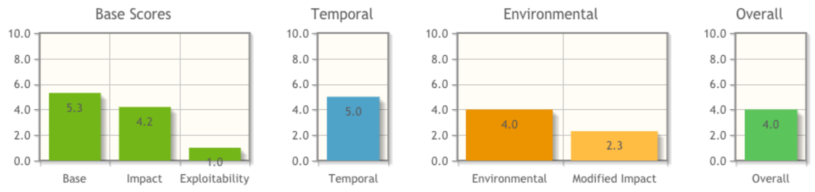
```
            int epfd;
            int maxevent;
            int etmodel;
            //fd_record_t *fd_records;
            struct epoll_event *events;

            //timer
            //struct ev_timer_t **heap;
            void **heap;
            int heap_size;
            int heap_capacity;
            int timer_fd;
    }ev_loop_t;
```

As we can see from the code, the epfd is defined in the struct ev_loop_t, but no initialization of epfd happens in the whole source code. Thus, before the epoll_ctl is invoked, the epfd is not initialized. And the epfd is epoll file descriptor, and epoll_ctl is used to add file descriptors it wants to be monitored under the specific epoll list, thus if the epfd is not clarified, possibly the file descriptors are not monitored successfully, and an attacker can make use of this to manipulate or crash the software.

=> Discovery process: Valgrind - dynamic analysis tool. Command: valgrind --track-origins=yes ./sparrow

```
==6711== Syscall param epoll_ctl(event) points to uninitialised byte(s)
==6711==    at 0x57250DA: epoll_ctl (syscall-template.S:78)
==6711==    by 0x10DFE9: ev_stop (ev_loop.c:179)
==6711==    by 0x10BE45: read_http (sparrow.c:528)
==6711==    by 0x10E1BA: ev_run_loop (ev_loop.c:216)
==6711==    by 0x10CAF5: worker_threads_entrance (thread_manage.c:27)
==6711==    by 0x53EB6DA: start_thread (pthread_create.c:463)
==6711==    by 0x572471E: clone (clone.S:95)
==6711==  Address 0xa631aa4 is on thread 2's stack
==6711==  in frame #1, created by ev_stop (ev_loop.c:162)
==6711==  Uninitialised value was created by a stack allocation
==6711==    at 0x10DE96: ev_stop (ev_loop.c:162)
```

=> Assessment: CWE-457: Use of Uninitialized Variable



CVSS Base Score: 5.3
Impact Subscore: 4.2
Exploitability Subscore: 1.0
CVSS Temporal Score: 5.0
CVSS Environmental Score: 4.0
Modified Impact Subscore: 2.3
Overall CVSS Score: 4.0

Show Equations

CVSS v3.0 Vector
AV:L/AC:H/PR:L/UI:N/S:U/C:N/I:H/A:L/E:P/RL:U/RC:C/CR:L/IR:L/AR:L/MAV:X/MAC:L/MPR:L/MUI:X/MS:U/MC:N/MI:H/MA: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.

Temporal Score Metrics

Exploit Code Maturity (E)
Not Defined (E:X) | Unproven that exploit exists (E:U) | [✓ Proof of concept code (E:P)] | Functional exploit exists (E:F) | High (E:H)

Remediation Level (RL)
Not Defined (RL:X) | Official fix (RL:O) | Temporary fix (RL:T) | Workaround (RL:W) | [✓ Unavailable (RL:U)]

Report Confidence (RC)
Not Defined (RC:X) | Unknown (RC:U) | Reasonable (RC:R) | [✓ Confirmed (RC:C)]

Environmental Score Metrics

Exploitability Metrics

Attack Vector (MAV)
[✓ Not Defined (MAV:X)] | Network (MAV:N) | Adjacent Network (MAV:A) | Local (MAV:L) | Physical (MAV:P)

Attack Complexity (MAC)
Not Defined (MAC:X) | [✓ Low (MAC:L)] | High (MAC:H)

Privileges Required (MPR)
Not Defined (MPR:X) | None (MPR:N) | [✓ Low (MPR:L)] | High (MPR:H)

User Interaction (MUI)
[✓ Not Defined (MUI:X)] | None (MUI:N) | Required (MUI:R)

Scope (MS)
Not Defined (MS:X) | [✓ Unchanged (MS:U)] | Changed (MS:C)

Impact Metrics

Confidentiality Impact (MC)
Not Defined (MC:X) | [✓ None (MC:N)] | Low (MC:L) | High (MC:H)

Integrity Impact (MI)
Not Defined (MI:X) | None (MI:N) | Low (MI:L) | [✓ High (MI:H)]

Availability Impact (MA)
Not Defined (MA:X) | None (MA:N) | [✓ Low (MA:L)] | High (MA:H)

Impact Subscore Modifiers

Confidentiality Requirement (CR)
Not Defined (CR:X) | [✓ Low (CR:L)] | Medium (CR:M) | High (CR:H)

Integrity Requirement (IR)
Not Defined (IR:X) | [✓ Low (IR:L)] | Medium (IR:M) | High (IR:H)

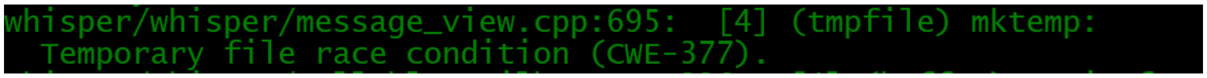
Availability Requirement (AR)
Not Defined (AR:X) | [✓ Low (AR:L)] | Medium (AR:M) | High (AR:H)

Program 2 Whisper (C++)

Source: https://github.com/syllable-org/whisper

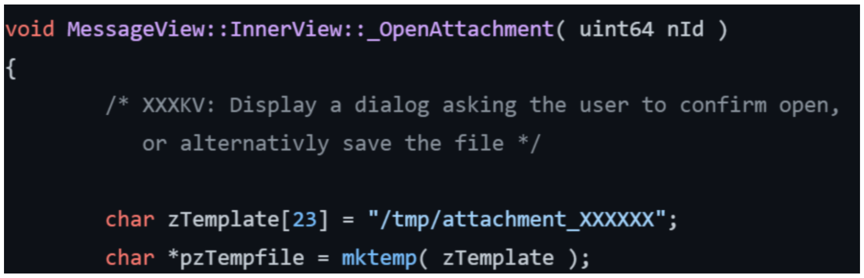
Vulnerability 2.1

- Description: The following code uses mktemp() to create a temporary file. The mktemp function generates a unique file name by modifying template 'tmp/attachment_XXX'. This is a potential vulnerability since it relies on an insecure method to create a temporary file. Although the mkrmp() function guarantees a unique file name at the time of creation, there is not a way to prevent an attacker or another process from choosing the same filename. There is a possible race condition problem because of choosing the same filenames. Further, if the process of generating filenames is not random enough, attackers may have a chance to guess the possible filenames.

- Category: Concurrency attack
- Discovery process:
    - Tool: Flawfinder - static analysis tool.
    - Command: flawfinder whisper
    - Error snippet:
    - 

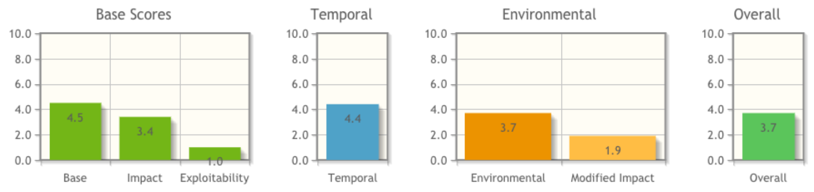
```
whisper/whisper/message_view.cpp:695:  [4] (tmpfile) mktemp:
  Temporary file race condition (CWE-377).
```

- Source code snippet:



```
void MessageView::InnerView::_OpenAttachment( uint64 nId )
{
        /* XXXKV: Display a dialog asking the user to confirm open,
           or alternativly save the file */

        char zTemplate[23] = "/tmp/attachment_XXXXXX";
        char *pzTempfile = mktemp( zTemplate );
```

- Assessment:



CVSS Base Score: 4.5
Impact Subscore: 3.4
Exploitability Subscore: 1.0
CVSS Temporal Score: 4.4
CVSS Environmental Score: 3.7
Modified Impact Subscore: 1.9
Overall CVSS Score: 3.7

Show Equations

CVSS v3.0 Vector
AV:L/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:L/E:F/RL:U/RC:C/CR:L/IR:L/AR:L/MAV:X/MAC:L/MPR:L/MUI:X/MS:U/MC:L/MI:L/MA: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.

Temporal Score Metrics

Exploit Code Maturity (E)
Not Defined (E:X) | Unproven that exploit exists (E:U) | Proof of concept code (E:P) | [✓ Functional exploit exists (E:F)] | High (E:H)

Remediation Level (RL)
Not Defined (RL:X) | Official fix (RL:O) | Temporary fix (RL:T) | Workaround (RL:W) | [✓ Unavailable (RL:U)]

Report Confidence (RC)
Not Defined (RC:X) | Unknown (RC:U) | Reasonable (RC:R) | [✓ Confirmed (RC:C)]

Environmental Score Metrics

Exploitability Metrics

Attack Vector (MAV)
[✓ Not Defined (MAV:X)] | Network (MAV:N) | Adjacent Network (MAV:A) | Local (MAV:L) | Physical (MAV:P)

Attack Complexity (MAC)
Not Defined (MAC:X) | [✓ Low (MAC:L)] | High (MAC:H)

Privileges Required (MPR)
Not Defined (MPR:X) | None (MPR:N) | [✓ Low (MPR:L)] | High (MPR:H)

User Interaction (MUI)
[✓ Not Defined (MUI:X)] | None (MUI:N) | Required (MUI:R)

Scope (MS)
Not Defined (MS:X) | [✓ Unchanged (MS:U)] | Changed (MS:C)

Impact Metrics

Confidentiality Impact (MC)
Not Defined (MC:X) | None (MC:N) | [✓ Low (MC:L)] | High (MC:H)

Integrity Impact (MI)
Not Defined (MI:X) | None (MI:N) | [✓ Low (MI:L)] | High (MI:H)

Availability Impact (MA)
Not Defined (MA:X) | None (MA:N) | [✓ Low (MA:L)] | High (MA:H)

Impact Subscore Modifiers

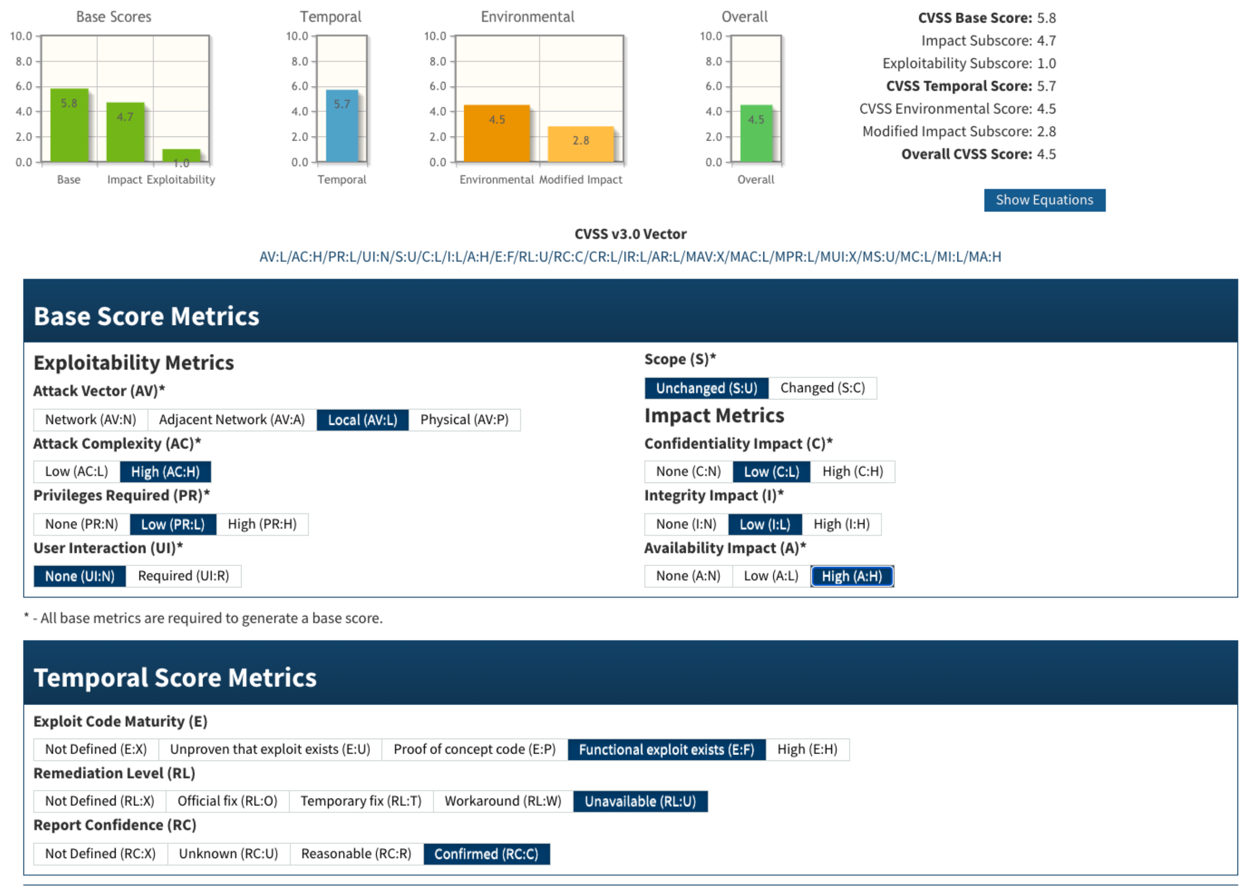
Confidentiality Requirement (CR)
Not Defined (CR:X) | [✓ Low (CR:L)] | Medium (CR:M) | High (CR:H)

Integrity Requirement (IR)
Not Defined (IR:X) | [✓ Low (IR:L)] | Medium (IR:M) | High (IR:H)

Availability Requirement (AR)
Not Defined (AR:X) | [✓ Low (AR:L)] | Medium (AR:M) | High (AR:H)

Vulnerability 2.2

- Description: To avoid overflows, the size of the array pointed by destination shall be long enough to contain the same C string as source. However, the following code fails to check this requirement. The destination part is a char array with size 512, but for the source part, we are not sure about the size of it. In the header file, the size of the buffer is 4096 which is larger than 512.
- Category: **Spatial Memory Attacks**
- Discovery process
  - Tool: Flawfinder - static analysis tool.
  - Command: flawfinder flawfinder whisper
- Assessment:
  - Impact: Attackers might use buffer overflow to inject the malicious code into the program. Other impacts already talked about in the first vulnerability section.
  - Severity: Already discussed above



CVSS Base Score: 5.8
Impact Subscore: 4.7
Exploitability Subscore: 1.0
CVSS Temporal Score: 5.7
CVSS Environmental Score: 4.5
Modified Impact Subscore: 2.8
Overall CVSS Score: 4.5

CVSS v3.0 Vector
AV:L/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:H/E:F/RL:U/RC:C/CR:L/IR:L/AR:L/MAV:X/MAC:L/MPR:L/MUI:X/MS:U/MC:L/MI:L/MA:H

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)]

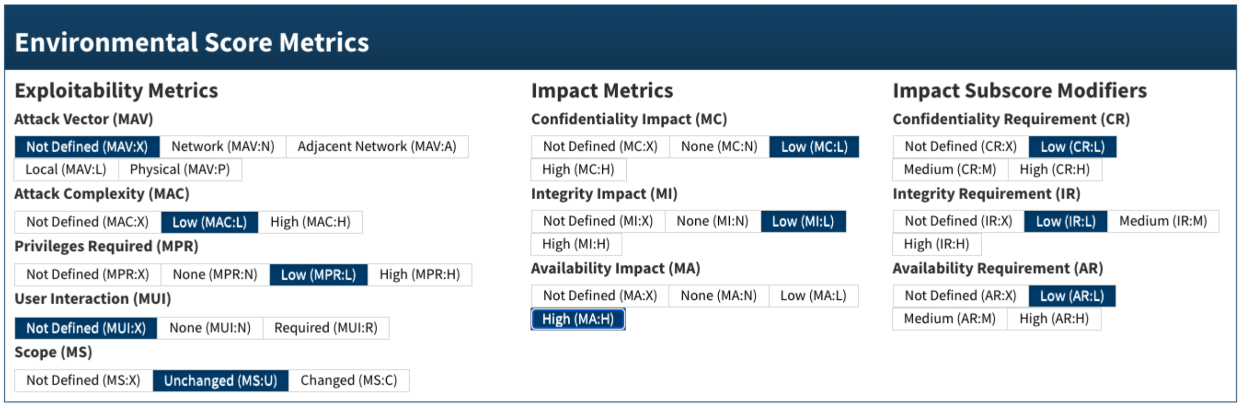
* - All base metrics are required to generate a base score.

Temporal Score Metrics

Exploit Code Maturity (E)
Not Defined (E:X) | Unproven that exploit exists (E:U) | Proof of concept code (E:P) | [✓ Functional exploit exists (E:F)] | High (E:H)

Remediation Level (RL)
Not Defined (RL:X) | Official fix (RL:O) | Temporary fix (RL:T) | Workaround (RL:W) | [✓ Unavailable (RL:U)]

Report Confidence (RC)
Not Defined (RC:X) | Unknown (RC:U) | Reasonable (RC:R) | [✓ Confirmed (RC:C)]



Environmental Score Metrics

Exploitability Metrics

Attack Vector (MAV)
[✓ Not Defined (MAV:X)] Network (MAV:N) Adjacent Network (MAV:A) Local (MAV:L) Physical (MAV:P)

Attack Complexity (MAC)
Not Defined (MAC:X) [✓ Low (MAC:L)] High (MAC:H)

Privileges Required (MPR)
Not Defined (MPR:X) None (MPR:N) [✓ Low (MPR:L)] High (MPR:H)

User Interaction (MUI)
[✓ Not Defined (MUI:X)] None (MUI:N) Required (MUI:R)

Scope (MS)
Not Defined (MS:X) [✓ Unchanged (MS:U)] Changed (MS:C)

Impact Metrics

Confidentiality Impact (MC)
Not Defined (MC:X) None (MC:N) [✓ Low (MC:L)] High (MC:H)

Integrity Impact (MI)
Not Defined (MI:X) None (MI:N) [✓ Low (MI:L)] High (MI:H)

Availability Impact (MA)
Not Defined (MA:X) None (MA:N) Low (MA:L) [✓ High (MA:H)]

Impact Subscore Modifiers

Confidentiality Requirement (CR)
Not Defined (CR:X) [✓ Low (CR:L)] Medium (CR:M) High (CR:H)

Integrity Requirement (IR)
Not Defined (IR:X) [✓ Low (IR:L)] Medium (IR:M) High (IR:H)

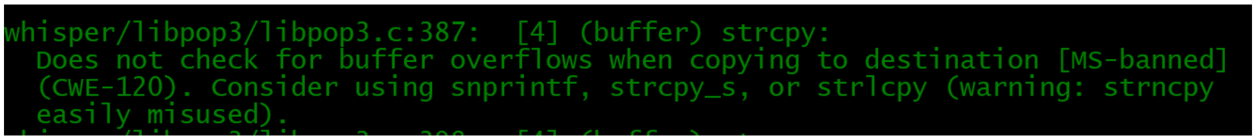
Availability Requirement (AR)
Not Defined (AR:X) [✓ Low (AR:L)] Medium (AR:M) High (AR:H)

- Code snippet

```
struct pop3_session {
        .....
        char *buffer;                   /* Input buffer */
        int status;                     /* Current connection/error status */
        .....
};
```

```
char response[512], response_count[256], response_size[256];
        int count, size;
        int ret, whitespace = 0;
        if( NULL == session || NULL == message_count || NULL == mailbox_size )
                return EINVAL;
        /* Wait for the server to respond with +OK */
        ret = __pop3_wait_for_ok( session );
        if( ret < 0 )
        {
                if( session->status & POP3_TIME )
                        return ETIME;
                else
                        return ECOMM;
        }
Line 387 strcpy( response, session->buffer + 4 );
```

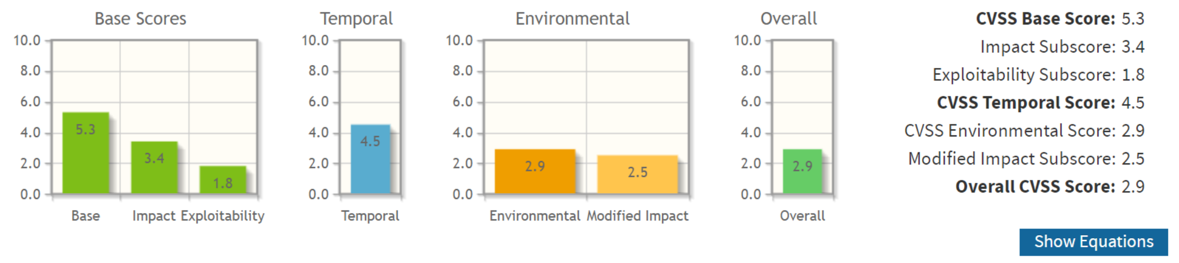
- Error snippet



```
whisper/libpop3/libpop3.c:387:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination [MS-banned]
  (CWE-120). Consider using snprintf, strcpy_s, or strlcpy (warning: strncpy
  easily misused).
```

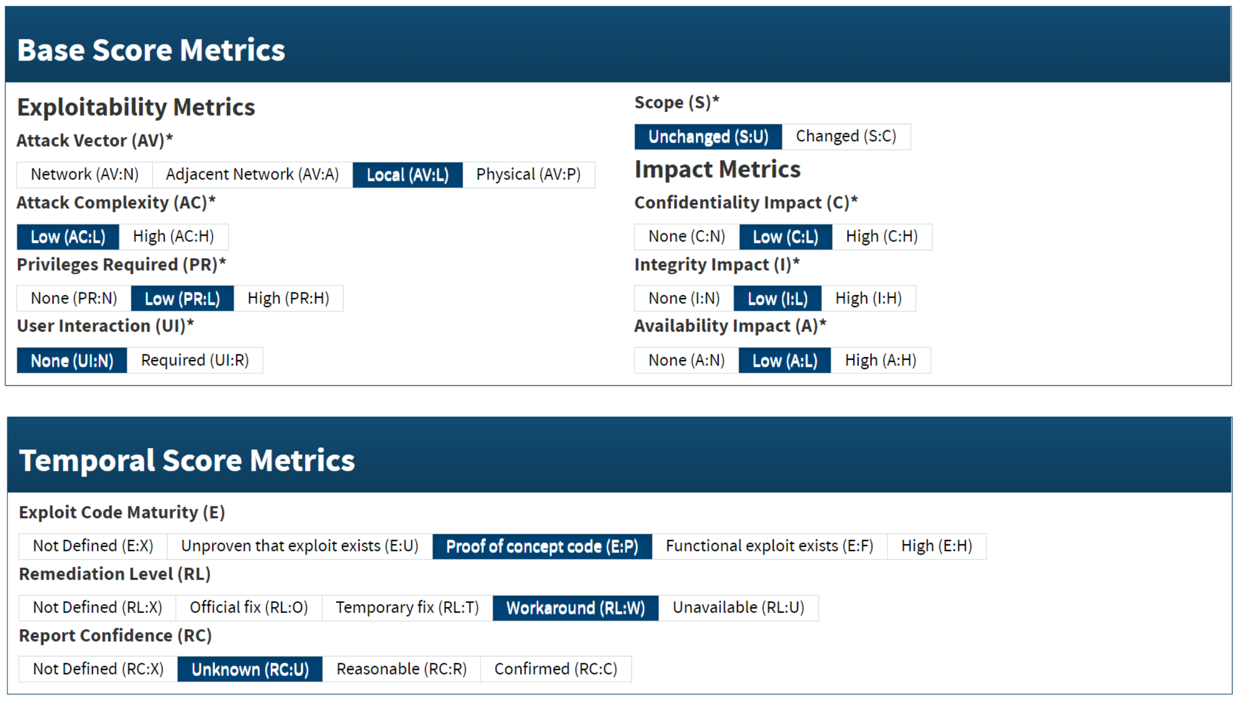
Vulnerability 2.3

- Description: Use getenv() command to get a certain environment variable. This command reports the value of the environment variable. Some environment variables visible for all the users of devices. Based on the official documents, the getenv() function is not thread-safe because it returns a value pointing to static data. Other processes or users may access and modify the same environment variables. There is a potential possibility of race conditions. After using the getenv() function, the program fails to check the environment variable before using it. The author then directly use variable *pzHome later.
- Category: Concurrency attack
- Discovery process:
    - Tool: Flawfinder - static analysis tool.
    - Command: flawfinder flawfinder whisper
- Assessment
    - CWE-807: Reliance on Untrusted Inputs in a Security Decision
    - Severity: Attacks may modify the environment variables, if we didn't check before use, the program may behave differently or even crash.
    - Impact: It will affect availability if attackers can crash the system.



CVSS Base Score: 5.3
Impact Subscore: 3.4
Exploitability Subscore: 1.8
CVSS Temporal Score: 4.5
CVSS Environmental Score: 2.9
Modified Impact Subscore: 2.5
Overall CVSS Score: 2.9

CVSS v3.1 Vector



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)

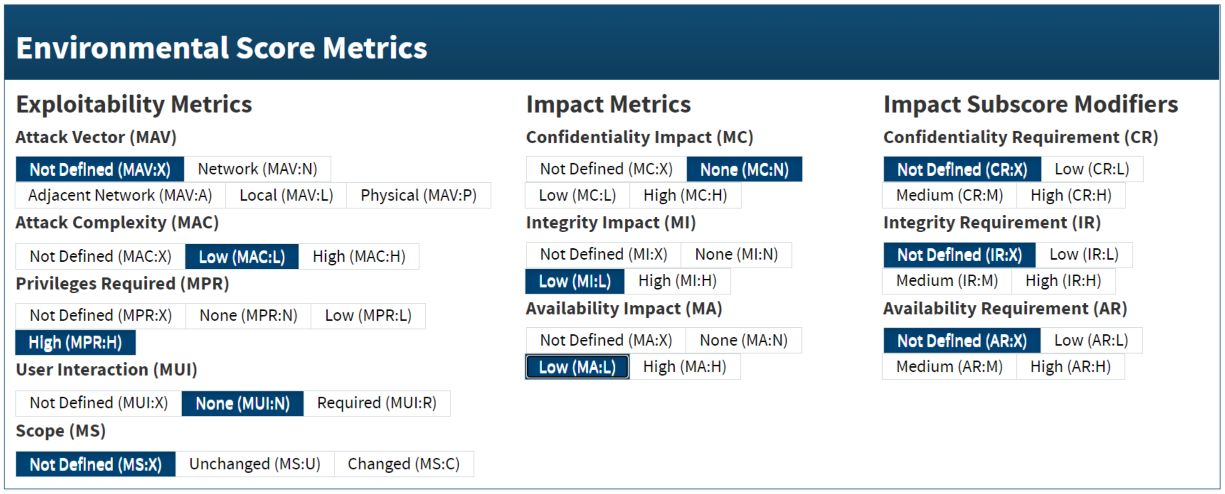
Availability Impact (A)*
None (A:N) | [✓ Low (A:L)] | High (A:H)

Temporal Score Metrics

Exploit Code Maturity (E)
Not Defined (E:X) | Unproven that exploit exists (E:U) | [✓ Proof of concept code (E:P)] | Functional exploit exists (E:F) | High (E:H)

Remediation Level (RL)
Not Defined (RL:X) | Official fix (RL:O) | Temporary fix (RL:T) | [✓ Workaround (RL:W)] | Unavailable (RL:U)

Report Confidence (RC)
Not Defined (RC:X) | [✓ Unknown (RC:U)] | Reasonable (RC:R) | Confirmed (RC:C)



Environmental Score Metrics

Exploitability Metrics

Attack Vector (MAV)

[✓ Not Defined (MAV:X)] Network (MAV:N) Adjacent Network (MAV:A) Local (MAV:L) Physical (MAV:P)

Attack Complexity (MAC)

Not Defined (MAC:X) [✓ Low (MAC:L)] High (MAC:H)

Privileges Required (MPR)

Not Defined (MPR:X) None (MPR:N) Low (MPR:L) [✓ High (MPR:H)]

User Interaction (MUI)

Not Defined (MUI:X) [✓ None (MUI:N)] Required (MUI:R)

Scope (MS)

[✓ Not Defined (MS:X)] Unchanged (MS:U) Changed (MS:C)

Impact Metrics

Confidentiality Impact (MC)

Not Defined (MC:X) [✓ None (MC:N)] Low (MC:L) High (MC:H)

Integrity Impact (MI)

Not Defined (MI:X) None (MI:N) [✓ Low (MI:L)] High (MI:H)

Availability Impact (MA)

Not Defined (MA:X) None (MA:N) [✓ Low (MA:L)] High (MA:H)

Impact Subscore Modifiers

Confidentiality Requirement (CR)

[✓ Not Defined (CR:X)] Low (CR:L) Medium (CR:M) High (CR:H)

Integrity Requirement (IR)

[✓ Not Defined (IR:X)] Low (IR:L) Medium (IR:M) High (IR:H)

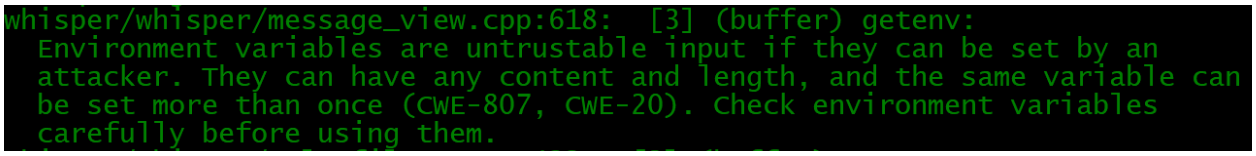
Availability Requirement (AR)

[✓ Not Defined (AR:X)] Low (AR:L) Medium (AR:M) High (AR:H)

- Code snippet

```
        Multipart cPart;
Line618 char *pzHome = getenv( "HOME" );

       if( m_pcMessage->GetPartById( cPart, nId ) != EOK )
             return;

       if( cPart.cFilename == "" )
             return;

Line626 os::String cFilename = os::String( pzHome ) + "/" + cPart.cFilename;
```

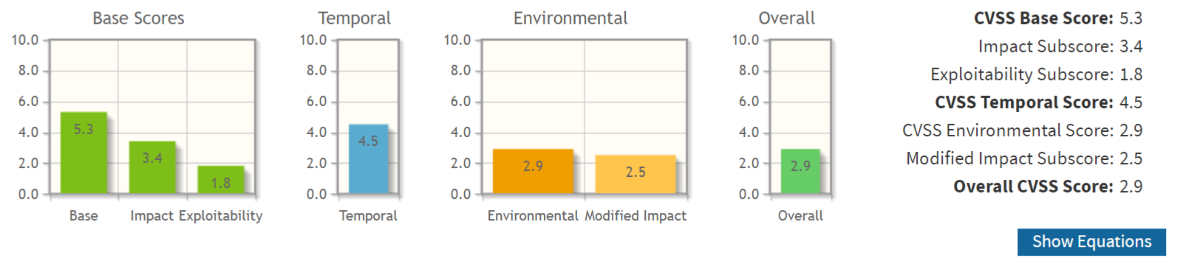
- Error snippet



```
whisper/whisper/message_view.cpp:618:  [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.
```

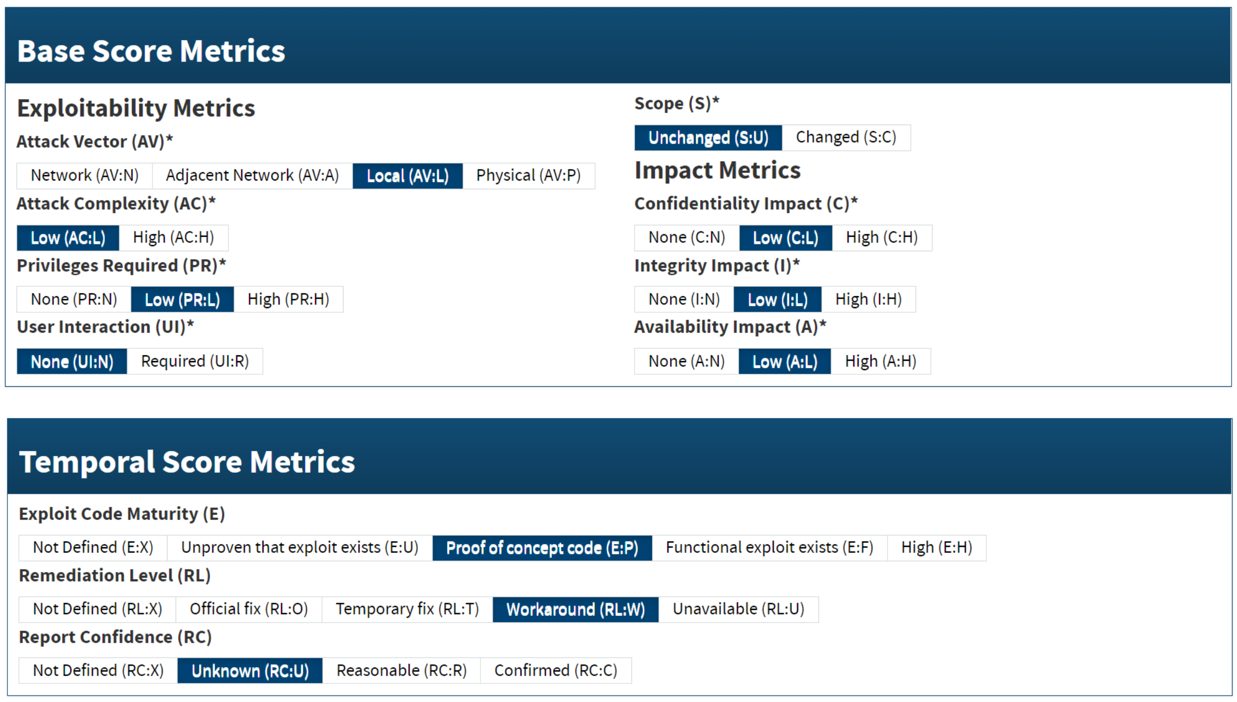
Vulnerability 2.4

- Description: Based on the official documents, the getenv() function is not thread-safe because it returns a value pointing to static data. After using getenv to get the value of Home, the program fails to perform any check before use the value in cMailboxPath().
- Category: Concurrency attack
- Discovery process:
  - Tool: Flawfinder - static analysis tool.
  - Command: flawfinder flawfinder whisper
- Assessment: Already explained in the 2.3.



CVSS Base Score: 5.3
Impact Subscore: 3.4
Exploitability Subscore: 1.8
CVSS Temporal Score: 4.5
CVSS Environmental Score: 2.9
Modified Impact Subscore: 2.5
Overall CVSS Score: 2.9

CVSS v3.1 Vector



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)

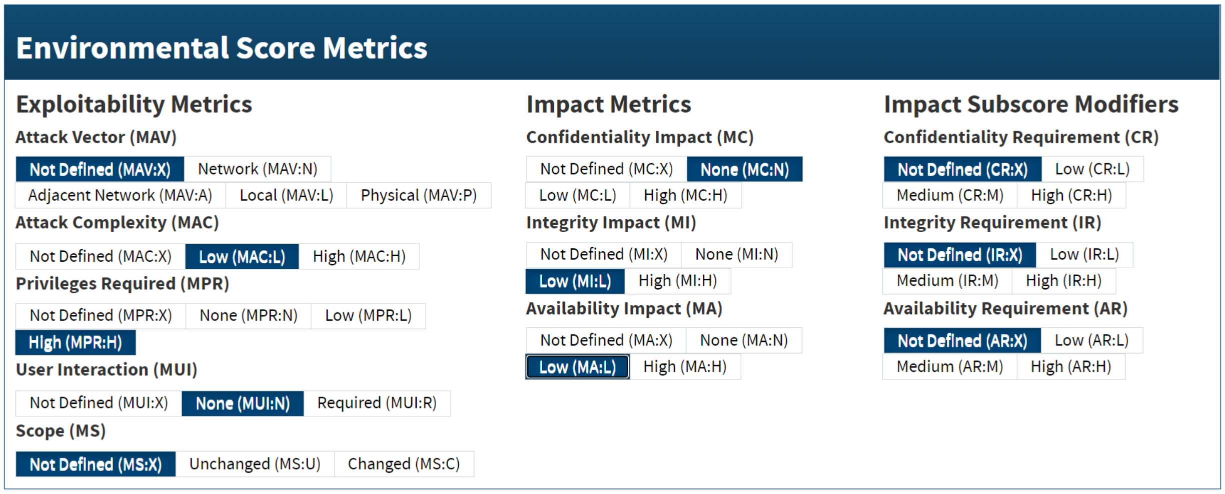
Availability Impact (A)*
None (A:N) | [✓ Low (A:L)] | High (A:H)

Temporal Score Metrics

Exploit Code Maturity (E)
Not Defined (E:X) | Unproven that exploit exists (E:U) | [✓ Proof of concept code (E:P)] | Functional exploit exists (E:F) | High (E:H)

Remediation Level (RL)
Not Defined (RL:X) | Official fix (RL:O) | Temporary fix (RL:T) | [✓ Workaround (RL:W)] | Unavailable (RL:U)

Report Confidence (RC)
Not Defined (RC:X) | [✓ Unknown (RC:U)] | Reasonable (RC:R) | Confirmed (RC:C)



Environmental Score Metrics

Exploitability Metrics

Attack Vector (MAV)

[✓ Not Defined (MAV:X)] Network (MAV:N) Adjacent Network (MAV:A) Local (MAV:L) Physical (MAV:P)

Attack Complexity (MAC)

Not Defined (MAC:X) [✓ Low (MAC:L)] High (MAC:H)

Privileges Required (MPR)

Not Defined (MPR:X) None (MPR:N) Low (MPR:L) [✓ High (MPR:H)]

User Interaction (MUI)

Not Defined (MUI:X) [✓ None (MUI:N)] Required (MUI:R)

Scope (MS)

[✓ Not Defined (MS:X)] Unchanged (MS:U) Changed (MS:C)

Impact Metrics

Confidentiality Impact (MC)

Not Defined (MC:X) [✓ None (MC:N)] Low (MC:L) High (MC:H)

Integrity Impact (MI)

Not Defined (MI:X) None (MI:N) [✓ Low (MI:L)] High (MI:H)

Availability Impact (MA)

Not Defined (MA:X) None (MA:N) [✓ Low (MA:L)] High (MA:H)

Impact Subscore Modifiers

Confidentiality Requirement (CR)

[✓ Not Defined (CR:X)] Low (CR:L) Medium (CR:M) High (CR:H)

Integrity Requirement (IR)

[✓ Not Defined (IR:X)] Low (IR:L) Medium (IR:M) High (IR:H)

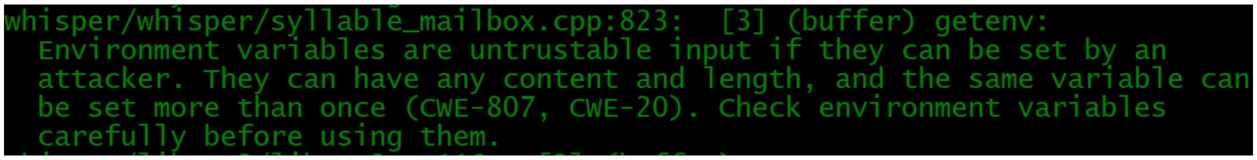
Availability Requirement (AR)

[✓ Not Defined (AR:X)] Low (AR:L) Medium (AR:M) High (AR:H)

- Code snippet

```
        String cPath;
        struct stat sStat;

Line823 Path cMailboxPath( getenv( "HOME" ) );
        cMailboxPath.Append( "/Mail" );
        cPath = cMailboxPath.GetPath();
```

- Error snippet



```
whisper/whisper/syllable_mailbox.cpp:823:  [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.
```