

VULNERABILITIES ASSESSMENT REPORT

Tyler Dao

SECU73000 – INTRODUCTION TO SOFTWARE SECURITY – SECTION 1

Contents

Code Review.....	2
Issues found by g++ compiler.....	3
Issues found manually.....	4
'Proof-of-concept' exploit.....	6
Steps to run exploit script.....	7
Code fix	8

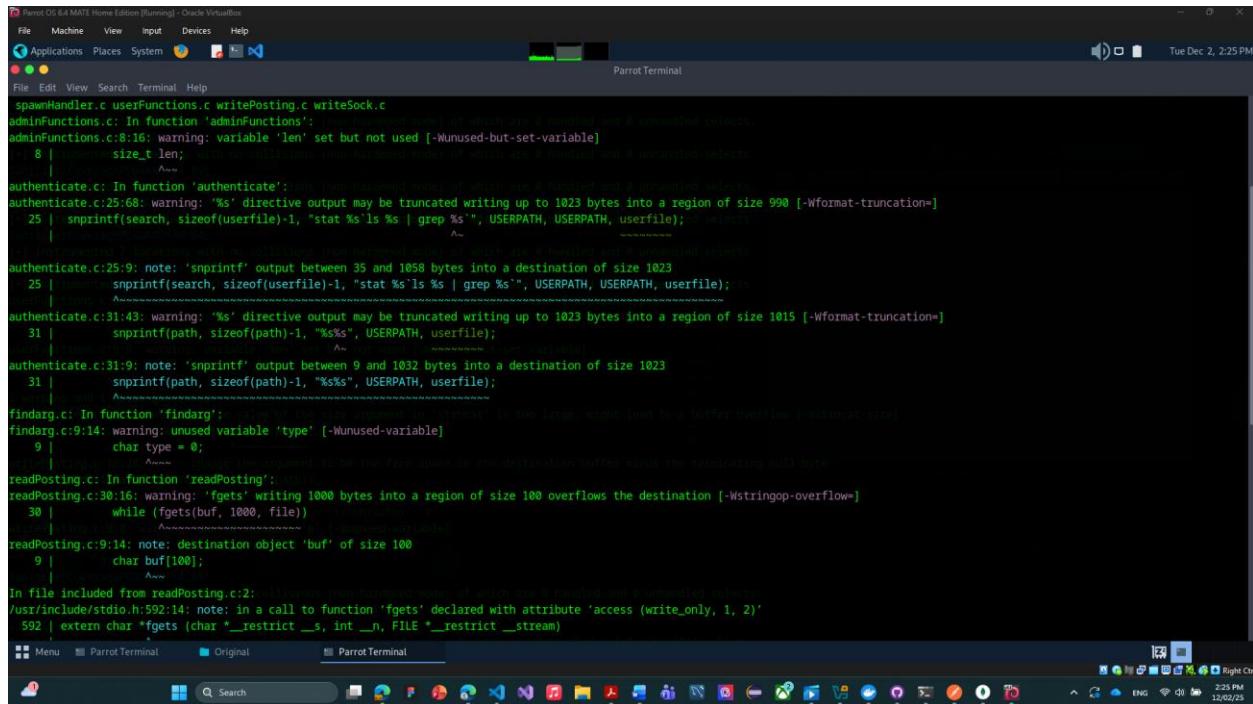
Code Review

To run vulnerabilities assessment, I used g++ compiler to find buffer overflow issues, unused variable issues; as well as manually check the code to find security issues, these are issues found:

Issue no.	File Name	Line	What	Risk
1	adminFunctions.c	8	“len” variable is unused	Unused variable
2	authenticate.c	16	Wrong size for memset, in the code, it takes sizeof(1024) but it's actually take the size of an integer which is 8 bytes	Buffer overflow
3	authenticate.c	19-20	Hard coded password	Attacker may guess the backdoor password
4	authenticate.c	25	snprintf() may write between 35 bytes to 1058 bytes into destination with size of 511 bytes	Buffer overflow
5	authenticate.c	26	system() runs search command without sanitizing	Injection attack → Attacker can inject a command in the username and run it on bash.
6	authenticate.c	31	snprintf() may write up to 1032 bytes into destination size of 1024 bytes	Buffer overflow
7	authenticate.c	48	Password is only checked by first 3 characters	Improper logic
8	findarg.c	9	“type” variable is unused	Unused variable
9	readPosting.c	14-15	Using strcpy and strcat may cause buffer overflow if action[1] is long	Buffer overflow

10	readPosting.c	30	Fgets() writes maximum 1000 bytes into a 100-byte buffer	Buffer overflow
11	userFunctions.c	9	"len" variable is unused	Unused variable
12	userFunctions.c	38	The function returns nothing, it should return an integer	Unchecked return
13	writePosting.c	9	"p" variable is unused	Unused variable
14	writePosting.c	16	strncat() might cause buffer overflow because it appends &action[1] into path, but path also has POSTINGPATH	Buffer overflow
15	writePosting.c	20	fopen() takes argument directly from action[1]	Dangerous method → Attacker can gain access to any file on the server.

Issues found by g++ compiler



```

Parrot OS 6.4 MATE Home Edition (Running - Oracle VirtualBox)
File Machine View Input Devices Help
Applications Places System Terminal Tue Dec 2, 2:25 PM
ParrotTerminal
File Edit View Search Terminal Help
spawnHandler.c userFunctions.c writePosting.c writeSock.c
adminFunctions.c: In function 'adminFunctions':
adminFunctions.c:8:16: warning: variable 'len' set but not used [-Wunused-but-set-variable]
  8 |     size_t len; /* len is the length of the file which are allocated and are unaligned selects.
  |     ^
authenticate.c: In function 'authenticate':
authenticate.c:25:68: warning: "%s" directive output may be truncated writing up to 1023 bytes into a region of size 990 [-Wformat-truncation=]
  25 |     sprintf(search, sizeof(userfile)-1, "stat %s ls %s | grep %s", USERPATH, USERPATH, userfile); /* len is the length of the file which are allocated and are unaligned selects.
  |     ^
authenticate.c:25:9: note: 'sprintf' output between 35 and 1058 bytes into a destination of size 1023
  25 |     sprintf(search, sizeof(userfile)-1, "stat %s ls %s | grep %s", USERPATH, USERPATH, userfile);
  |     ^
authenticate.c:31:43: warning: "%s" directive output may be truncated writing up to 1023 bytes into a region of size 1015 [-Wformat-truncation=]
  31 |     sprintf(path, sizeof(path)-1, "%s%s", USERPATH, userfile);
  |     ^
authenticate.c:31:9: note: 'sprintf' output between 9 and 1032 bytes into a destination of size 1023
  31 |     sprintf(path, sizeof(path)-1, "%s%s", USERPATH, userfile);
  |     ^
findarg.c: In function 'findarg':
findarg.c:9:14: warning: unused variable 'type' [-Wunused-variable]
  9 |     char type = 0;
  |     ^
  |     ^~~~~~ unused variable 'type' to be free space in the destination buffer since the reinitializing null byte.
readPosting.c: In function 'readPosting':
readPosting.c:30:16: warning: fgets' writing 1000 bytes into a region of size 100 overflows the destination [-Wstringop-overflow=]
  30 |     while (fgets(buf, 1000, file))
  |     ^
  |     ^~~~~~ fgets() writes maximum 1000 bytes into a 100-byte buffer
readPosting.c:9:14: note: destination object 'buf' of size 100
  9 |     char buf[100];
  |     ^
In file included from readPosting.c:2:
/usr/include/studio.h:592:14: note: in a call to function 'fgets' declared with attribute 'access (write_only, 1, 2)'
  592 | extern char *fgets (char *_restrict __s, int __n, FILE *_restrict __stream)
  |     ^
  |     ^~~~~~ fgets() writes maximum 1000 bytes into a 100-byte buffer

```

Parrot OS 6.4 MATE Home Edition [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Applications Places System Terminal Help

Parrot Terminal

```
In file included from readPosting.c:2:
/usr/include/stdio.h:592:14: note: in a call to function 'fgets' declared with attribute 'access (write_only, 1, 2)'
 592 | extern char *fgets (char *_restrict __s, int __n, FILE *_restrict __stream)
               ^~~~~~
readPosting.c:30:16: warning: 'fgets' writing 1000 bytes into a region of size 100 overflows the destination [-Wstringop-overflow=]
 30 |         while (fgets(buf, 1000, file))
               ^~~~~~                                         in the source code of which are & höchstens and & minimum selects
readPosting.c:9:14: note: destination object 'buf' of size 100
 9 |         char buf[100];
               ^~~~~~                                         in the source code of which are & höchstens and & minimum selects
/usr/include/stdio.h:592:14: note: in a call to function 'fgets' declared with attribute 'access (write_only, 1, 2)'
 592 | extern char *fgets (char *_restrict __s, int __n, FILE *_restrict __stream)
               ^~~~~~                                         in the source code of which are & höchstens and & minimum selects
userFunctions.c: In function 'userFunctions':
userFunctions.c:38:25: warning: 'return' with no value, in function returning non-void [-Wreturn-type]
 38 |     return;
               ^~~~~~
userFunctions.c:6:5: note: declared here
 6 | int userFunctions(FILE *logfile, int sock, char *user) {
               ^~~~~~                                         in the source code of which are & höchstens and & minimum selects
userFunctions.c:9:16: warning: variable 'len' set but not used [-Wunused-but-set-variable]
 9 |     size_t len;
               ^~~~~~                                         in the source code of which are & höchstens and & minimum selects
writePosting.c: In function 'writePosting':
writePosting.c:9:15: warning: unused variable 'p' [-Wunused-variable]
 9 |     char *p;
               ^~~~~~                                         in the source code of which are & höchstens and & minimum selects
writePosting.c:16:9: warning: 'strncat' specified bound 1024 equals destination size [-Wstringop-overflow=]
 16 |     strncat(path, &action[1], sizeof(path));
               ^~~~~~                                         in the source code of which are & höchstens and & minimum selects
-[User@parrot]=[/Desktop/Software Security Project/Original/Project]
```

Issues found manually

Nano OS 0.4 MATE Home Edition [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Applications Places System

File Edit Selection View Go Run Terminal Help

Project

C authenticate.c

```
8 {
18 /* FIXME: hard coded admin backdoor for password recovery */
19 if (memcmp(pass, "_letMeIn!", 9) == 0)
20     return 1;
21
22 /* look up user by checking user files: done via system() to /bin/ls|grep user */
23 logData(logfile, "performing lookup for user via system()\n");
24 snprintf(userfile, sizeof(userfile)-1, "%s.txt", user);
25 snprintf(search, sizeof(userfile)-1, "stat %s ls %s | grep %s", USERPATH, USERPATH, userfile);
26 ret = system(search);
27
28 if (ret != 0)
29     return 2;
30
31 snprintf(path, sizeof(path)-1, "%s%s", USERPATH, userfile);
32
33 /* open file and check if contents == password */
34 file = fopen(path, "r");
35
36 if (!file)
37 {
38     logData(logfile, "fopen for userfile failed\n");
39     return 2;
40 }
41
42 logData(logfile, "getting userfile info\n");
43 fgets(data, sizeof(data)-1, file);
44
45 fclose(file);
46
47 /* Password Check! */
48 if (memcmp(data, pass, 3))
49     return 3;
50
51 return 1;
52 }
```

Ln 27, Col 1(22 selected) Tab Size:4 UTF-8 LF () C Q

ParrotTerminal Original ParrotTerminal authenticate.c - Proj...

Search

File Machine View Input Devices Help

Applications Places System

File Edit Selection View Go Run Terminal Help

Project

C authenticate.c

```
8 {
18 /* FIXME: hard coded admin backdoor for password recovery */
19 if (memcmp(pass, "_letMeIn!", 9) == 0)
20     return 1;
21
22 /* look up user by checking user files: done via system() to /bin/ls|grep user */
23 logData(logfile, "performing lookup for user via system()\n");
24 snprintf(userfile, sizeof(userfile)-1, "%s.txt", user);
25 snprintf(search, sizeof(userfile)-1, "stat %s ls %s | grep %s", USERPATH, USERPATH, userfile);
26 ret = system(search);
27
28 if (ret != 0)
29     return 2;
30
31 snprintf(path, sizeof(path)-1, "%s%s", USERPATH, userfile);
32
33 /* open file and check if contents == password */
34 file = fopen(path, "r");
35
36 if (!file)
37 {
38     logData(logfile, "fopen for userfile failed\n");
39     return 2;
40 }
41
42 logData(logfile, "getting userfile info\n");
43 fgets(data, sizeof(data)-1, file);
44
45 fclose(file);
46
47 /* Password Check! */
48 if (memcmp(data, pass, 3))
49     return 3;
50
51 return 1;
52 }
```

Ln 49, Col 18 (38 selected) Tab Size:4 UTF-8 LF () C Q

ParrotTerminal Original ParrotTerminal authenticate.c - Proj...

Search

Kali OS 6.4 MATE Home Edition [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Applications Places System Terminal Help

File Edit Selection View Go Run Terminal Help

Project

EXPLORER

readPosting.c

```

1 #include <string.h>
2 #include <stdio.h>
3
4 #include "globals.h"
5
6 void readPosting(int sock, FILE *logfile, char *action)
7 {
8     FILE *file;
9     char buf[100];
10    char path[100];
11
12    logData(logfile, &action[1]);
13
14    strcpy(path, POSTINGPATH);
15    strcat(path, &action[1]);
16
17    logData(logfile, "user request to read posting: %s", path);
18
19    file = fopen(path, "r");
20
21    if (!file)
22    {
23        writeSock(sock, FILENOTAVAIL, sizeof(FILENOTAVAIL));
24        return;
25    }
26
27    /* fgets for the size of the buffer (100), from the file
28    writing the article to the user each time */
29
30    while (fgets(buf, 1000, file))
31        writeSock(sock, buf, strlen(buf));
32
33    fclose(file);
34
35    return;
36 }

```

OUTLINE

Timeline

0.0.0

Menu ParrotTerminal [Fixed] [ParrotTerminal] Final project authenticate.c - Project... readPosting.c - Project...

Search

Tue Dec 2, 4:01 PM

Ln 14, Col 1 (54 selected) Tab Size: 4 UTF-8 LF () C Q

4:02 PM ENG 12/02/25

Kali OS 6.4 MATE Home Edition [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Applications Places System Terminal Help

File Edit Selection View Go Run Terminal Help

Project

EXPLORER

writePosting.c

```

1 #include <string.h>
2 #include <stdio.h>
3
4 #include "globals.h"
5
6 void writePosting(int sock, FILE *logfile, char *action)
7 {
8     FILE *file;
9     char *p;
10    size_t x, y;
11    int complete = 0;
12    char buf[1024];
13    char path[1024];
14
15    strcpy(path, POSTINGPATH);
16    strcat(path, &action[1], sizeof(path));
17
18    logData(logfile, "user writing posting: %s", path);
19
20    file = fopen(&action[1], "w");
21
22    if (!file)
23    {
24        writeSock(sock, FILENOTAVAIL, sizeof(FILENOTAVAIL));
25        return;
26    }
27
28    writeSock(sock, BEGINFILE, sizeof(BEGINFILE));
29
30    while (1)
31    {
32        memset(buf, 0, sizeof(buf));
33        x = readSock(sock, buf, sizeof(buf)-1);
34        for (y = 0; y < x; ++y)
35        {
36            if (buf[y] == '\n')
37            {
38                if (buf[y+1] == '\r' & buf[y+2] == '\n')
39                {
40                    buf[y] = 0x0;
41                    complete = 1;
42                }
43            }
44        }
45    }
46
47    logData(logfile, "%s", buf);
48
49    if (complete)
50    {
51        writeSock(sock, ENDFILE, sizeof(ENDFILE));
52    }
53
54 }

```

OUTLINE

Timeline

0.0.0

Menu ParrotTerminal Original ParrotTerminal writePosting.c - Project...

Search

Tue Dec 2, 2:31 PM

Ln 20, Col 5 (30 selected) Tab Size: 4 UTF-8 LF () C Q

2:33 PM ENG 12/02/25

‘Proof-of-concept’ exploit

To run this exploit, I have created a python script. The script will insert a .txt file in the server directory, where the user is not supposed to be accessed. In this case, the python script will insert a

hacked.txt file in tmp folder, which is a folder that most user will have permissions to write in it, this vulnerability can be taken by attacker to deliver malware in real attack. The script will send an authentication request to server; however, it never actually tries to authenticate with the server, it will insert a malicious username “test; /bin/sh -c ‘echo hacked > /tmp/hacked.txt’; #” that will trigger bash command to create hacked.txt in the tmp folder. The python script is included in the zip file submitted with this report.

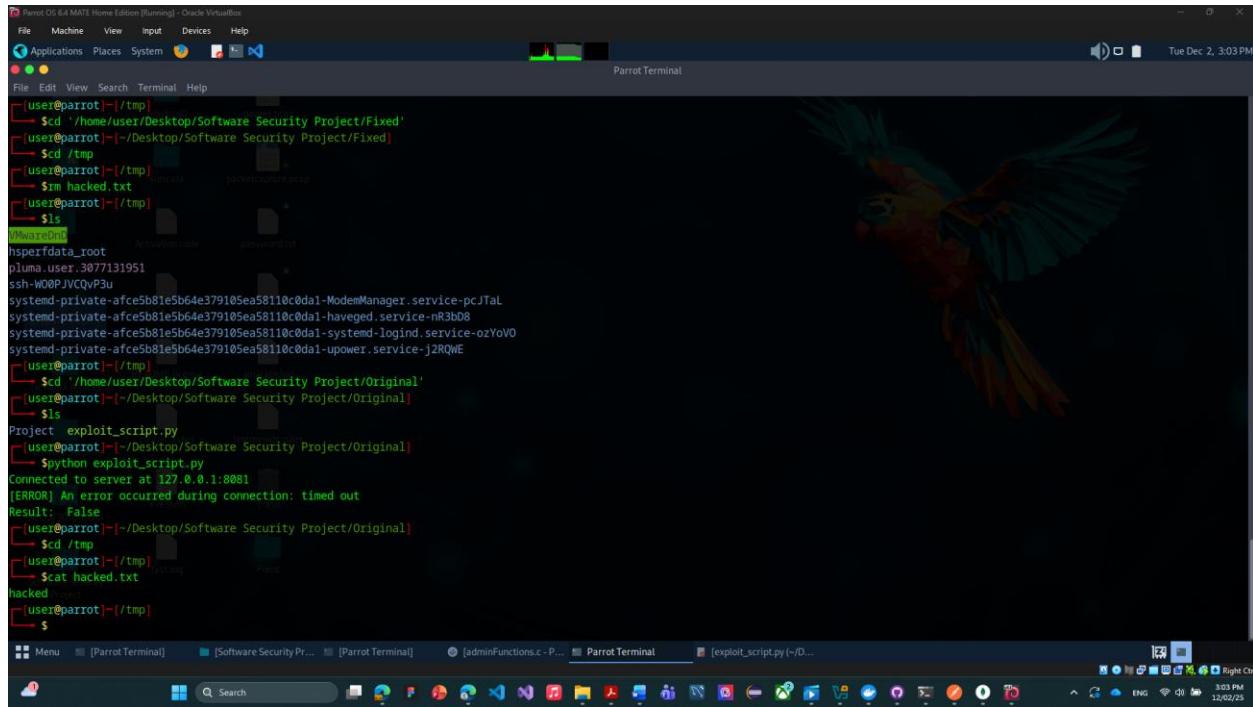
Steps to run exploit scrip

Prerequisite:

- **Server is running locally on port 8081**
- **Python**

Step to run the python script:

1. cd into the project folder that includes the source code files
2. Run **make** to compile the code
3. Run **./project** to start the server, the server is running in background
4. cd into the folder that includes the script
5. Run **python exploit_script.py**, then wait for the script to terminate the connection with server
6. cd into **/tmp** folder, using **ls** to see all the files in tmp folder and you will see a file called hacked.txt created by the exploit script, or run **cat hacked.txt** to see the content of the file.



The screenshot shows a terminal window titled "Parrot Terminal" on a Parrot OS desktop environment. The terminal output is as follows:

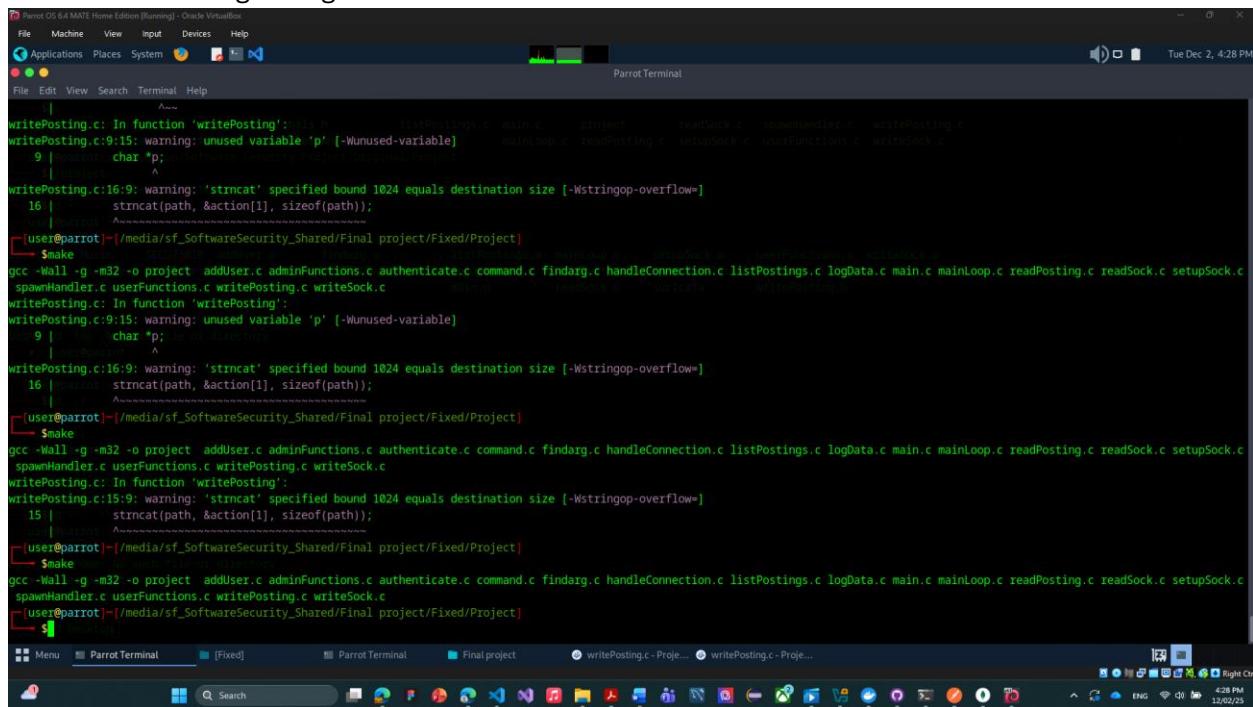
```
[user@parrot]~/Desktop/Software Security Project/Fixed$ cd /tmp
[User@parrot]~/Desktop/Software Security Project/Fixed$ ./project
[User@parrot]~/Desktop/Software Security Project/Fixed$ cd /tmp
[User@parrot]~/Desktop/Software Security Project/Fixed$ ls
[User@parrot]~/Desktop/Software Security Project/Fixed$ python exploit_script.py
[User@parrot]~/Desktop/Software Security Project/Fixed$ Connected to server at 127.0.0.1:8081
[User@parrot]~/Desktop/Software Security Project/Fixed$ [ERROR] An error occurred during connection: timed out
[User@parrot]~/Desktop/Software Security Project/Fixed$ Result: False
[User@parrot]~/Desktop/Software Security Project/Fixed$ cd /tmp
[User@parrot]~/Desktop/Software Security Project/Fixed$ cat hacked.txt
hacked
[User@parrot]~/Desktop/Software Security Project/Fixed$
```

Code fix

Issue no.	File name	Line	Original code	Fixed code
1	adminFunctions.c	8, 12	8 size_t len; 12 len = readSock(sock, action, sizeof(action));	8 Remove unused variable 12 readSock(sock, action, sizeof(action));
2	authenticate.c	16	memset(path, 0, sizeof(1024));	memset(path, 0, sizeof(path));
3	authenticate.c	19- 20	if (memcmp(pass, "_letM3In!", 9) == 0) return 1;	Remove it
4, 5, 6	authenticate.c	24- 31	24 snprintf(userfile, sizeof(userfile)-1, "%s.txt", user); 25 snprintf(search, sizeof(userfile)-1, "stat %s`ls %s grep %s `", USERPATH, USERPATH, userfile); 26 ret = system(search); 28 if (ret != 0) 29 return 2; 31 snprintf(path, sizeof(path)-1, "%s%s", USERPATH, userfile);	10 char path[1033] //Included sizeof(USERPATH) 24 sprint(path, sizeof(path)-1, "%s%s", USERPATH, userfile); 25 if(access(path, 0) != 0){ //Check if path existed, use access() instead of system() so attacker can not perform command injection attack 26 return 2;} //If path does not exist
7	authenticate.c	48	if (memcmp(data, pass, 3))	If(strcmp(data, pass)!=0) //Compare the whole string
8	findarg.c	9	char type = 0;	Remove unused variable
9	readPosting.c	14- 15	14 strcpy(path, POSTINGPATH); 15 strcat(path, &action[1], sizeof(path) - sizeof(POSTINGPATH)-1); //make sure only 99 characters are copied into path + \0	14 strcpy(path, POSTINGPATH); 15 strncat(path, &action[1], sizeof(path) - sizeof(POSTINGPATH)-1); //make sure only 99 characters are copied into path + \0
10	readPosting.c	30	while (fgets(buf, 1000, file))	while (fgets(buf, 100, file))
11	userFunctions.c	9, 21	9 size_t len; 21 len = readSock(sock, action, sizeof(action));	9 Remove unused variable 21 readSock(sock, action, sizeof(action));

12	userFunctions.c	38	return;	return -1; //Return an integer
13	writePosting.c	9	char* p;	Remove unused variable
14	writePosting.c	16	strncat(path, &action[1], sizeof(path));	strncat(path, &action[1], sizeof(path) - sizeof(POSTINGPATH) - 1);
15	writePosting.c	20	file = fopen(&action[1], "w");	file = fopen(path, "w"); //Use the correct path

There is no warning from g++ after the code is fixed



```

Parrot OS 6.4 MATE Home Edition [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System Firefox Terminal
Parrot Terminal
Tue Dec 2, 4:28 PM
File Edit View Search Terminal Help
|   ^~~
writePosting.c: In function 'writePosting':
writePosting.c:9:15: warning: unused variable 'p' [-Wunused-variable]
  9 |     char *p;
     |
writePosting.c:16:9: warning: 'strncat' specified bound 1024 equals destination size [-Wstringop-overflow=]
 16 |     strncat(path, &action[1], sizeof(path));
     |
[user@parrot]~/media/sf_SoftwareSecurity_Shared/Final project/Fixed/Project]
  -> Make
gcc -Wall -g -m32 -o project addUser.c adminFunctions.c authenticate.c command.c findarg.c handleConnection.c listPostings.c logData.c main.c mainLoop.c readPosting.c readSock.c setupSock.c spawnHandler.c userFunctions.c writePosting.c writeSock.c
writePosting.c: In function 'writePosting':
writePosting.c:9:15: warning: unused variable 'p' [-Wunused-variable]
  9 |     char *p;
     |
writePosting.c:16:9: warning: 'strncat' specified bound 1024 equals destination size [-Wstringop-overflow=]
 16 |     strncat(path, &action[1], sizeof(path));
     |
[user@parrot]~/media/sf_SoftwareSecurity_Shared/Final project/Fixed/Project]
  -> Make
gcc -Wall -g -m32 -o project addUser.c adminFunctions.c authenticate.c command.c findarg.c handleConnection.c listPostings.c logData.c main.c mainLoop.c readPosting.c readSock.c setupSock.c spawnHandler.c userFunctions.c writePosting.c writeSock.c
writePosting.c: In function 'writePosting':
writePosting.c:15:9: warning: 'strncat' specified bound 1024 equals destination size [-Wstringop-overflow=]
 15 |     strncat(path, &action[1], sizeof(path));
     |
[user@parrot]~/media/sf_SoftwareSecurity_Shared/Final project/Fixed/Project]
  -> Make
gcc -Wall -g -m32 -o project addUser.c adminFunctions.c authenticate.c command.c findarg.c handleConnection.c listPostings.c logData.c main.c mainLoop.c readPosting.c readSock.c setupSock.c spawnHandler.c userFunctions.c writePosting.c writeSock.c
[user@parrot]~/media/sf_SoftwareSecurity_Shared/Final project/Fixed/Project]

```

There is no file created by the script when trying with fixed code

