Lecture 6: Software Vulnerabilities: Input Validation

Sanchuan Chen

schen@auburn.edu

9/4/2023

Why Software is Vulnerable

- Complexity
- Developed w/ memory unsafe languages (C/C++)
- Programmer's mistakes
  - Lack of input validation (when opening connection to anyone in the Internet)
  - Improper data validation
- Architectural vulnerabilities

Software Vulnerability Categories

1. Input validation errors

- Code injection
- Cross-site scripting in web applications
- Directory traversal
- Format string
- E-mail injection
- HTTP header injection
- HTTP response splitting
- SQL injection

Software Vulnerability Categories

2. Memory safety errors

- Stack overflow
- Heap overflow
- Global data (.got, .data, .bss) overflow
- Integer overflow
- User-after-free
- Double free

3. Privilege-confusion bugs

- Clickjacking
- Cross-site request forgery
- FTP bounce attack

Software Vulnerability Categories

4. **Privilege escalation**

5. **Race conditions**
   - Symlink races
   - Time-of-check-to-time-of-use (TOCTOU)

6. **Side-channel vulnerabilities**
   - Timing side channel
   - Power side channel
   - Micro-architectural side channel (Meltdown, Spectre)

- One of the key factors in developing secure software is to **validate (e.g. check and verify) the input**. Without input validation as a primary software development approach, the implemented software could be susceptible to evil attackers
- **Input validation** is considered a "best practice" for coding, regardless of the software that is being used for the development of applications (Java, C++, C, and yes COBOL, etc.).
- With input validation, the software developer configures all input fields to only accept very specific types of data with a predefined length (no arbitrary length).

Without Input Validation

cat_grade.c

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    char command[100];
    sprintf(command, "cat grades/%s", argv[1]);
    system(command);
    return 0;
}
```

```
$ ls grades/
alice  bob
```

```
$ ./cat_grade alice
A
```

```
$ ./cat_grade bob
B
```

Without Input Validation

cat_grade.c

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    char command[100];
    sprintf(command, "cat grades/%s", argv[1]);
    system(command);
    return 0;
}
```

```
$ ./cat_grade `python3 -c 'print("A"*100)'`
cat: grades/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\
        AAAAAAAAAAAAAAAA: No such file or directory
Segmentation fault (core dumped)
```

```
$ ./cat_grade `perl -e 'print("A"x100)'`
cat: grades/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\
        AAAAAAAAAAAAAAAA: No such file or directory
Segmentation fault (core dumped)
```

```
$ ./cat_grade $(perl -e 'print("A"x100)')
cat: grades/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\
        AAAAAAAAAAAAAAAA: No such file or directory
Segmentation fault (core dumped)
```

```
$ echo $(perl -e 'print("A"x30)')
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
$ echo `perl -e 'print("A"x30)'`
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

With Input Validation

cat_grade_arg_check.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define CMD_LEN 128
int main(int argc, char **argv)
{
  char command[CMD_LEN + 12];
  if (argc !=2)
  {
    printf("cat_grade <name>\n");
    return 1;
  }

  if (strlen(argv[1])>CMD_LEN)
  {
    printf("The <name> you provide is too long\n");
    return 2;
  }

  snprintf(command, 100, "cat grades/%s", argv[1]);
  system(command);
  return 0;
}
```

```
$ ./cat_grade_arg_check alice
A


$ ./cat_grade_arg_check `python3 -c 'print("A"*100)'`
cat: grades/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\
    A: No such file or directory


$ ./cat_grade_arg_check `python3 -c 'print("A"*200)'`
The <name> you provide is too long


$ ./cat_grade_arg_check `perl -e 'print("A"x30)'`
cat: grades/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA: No such
file or directory
```

With Input Validation

cat_grade_arg_check.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define CMD_LEN 128
int main(int argc, char **argv)
{
  char command[CMD_LEN + 12];
  if (argc !=2)
  {
    printf("cat_grade <name>\n");
    return 1;
  }

  if (strlen(argv[1])>CMD_LEN)
  {
    printf("The <name> you provide is too long\n");
    return 2;
  }

  snprintf(command, 100, "cat grades/%s", argv[1]);
  system(command);
  return 0;
}
```

```
$ ./cat_grade_arg_check "alice;/bin/cat /etc/passwd"
A
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
...
```

With Improper Input Validation

cat_grade_exec.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#define CMD_LEN 128
int main(int argc, char **argv)
{
  char path[CMD_LEN + 12];
  if (argc !=2)
  {
    printf("cat_grade <name>\n");
    return 1;
  }

  if (strlen(argv[1])>CMD_LEN)
  {
    printf("The <name> you provide is too long\n");
    return 2;
  }

  snprintf(path, CMD_LEN+12, "./grades/%s", argv[1]);

  execl("/bin/cat","/bin/cat", path, NULL);

  return 0;
}
```

```
$ ./cat_grade_exec alice
A
```

```
$ ./cat_grade_exec "alice;/bin/cat /etc/passwd"
/bin/cat: './grades/alice;/bin/cat /etc/passwd':
No such file or directory
```

```
$ ./cat_grade_exec ../../../../../etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
...
```

With Proper Input Validation

cat_grade_exec_path_check.c

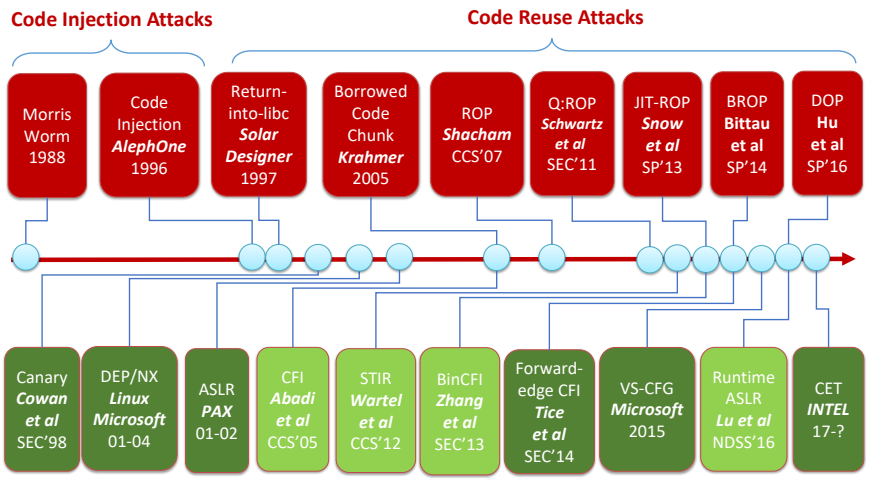
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#define CMD_LEN 128
int valid_path(char *path) {
  if (strchr(path,'.')) return 0;
  if (strchr(path,'/'))return 0;

  return 1;
}
int main(int argc, char **argv) {
  char path[CMD_LEN + 12];
  if (argc !=2) {
    printf("cat_grade <name>\n");
    return 1;
  }
  if (strlen(argv[1])>CMD_LEN) {
    printf("The <name> you provide is too long\n");
    return 2;
  }
  if(!valid_path(argv[1])) {
    printf("invalid path input %s\n",argv[1]);
    return 3;
  }
  snprintf(path, CMD_LEN+12, "./grades/%s", argv[1]);
  execl("/bin/cat","/bin/cat", path, NULL);
  return 0;
}
```

```
$ ./cat_grade_exec_path_check alice
A

$ ./cat_grade_exec_path_check ../..
invalid path input ../..
```

Thank You



1

schen@auburn.edu
schuan.github.io

[1] Instructor appreciates the help from Prof. Zhiqiang Lin.