Software Security

Daniel Bosk[1]

Department of Information and Communication Systems,
Mid Sweden University, SE-851 70 Sundsvall

software.tex 1999 2014-09-23 11:14:06Z danbos

Overview

1. Introduction
   - Security and Reliability
   - Malware
   - Change in Environment
2. Broken Abstractions
   - Numbers and Characters
   - Function Composition
3. Memory Management
   - Memory Structure
   - Overruns
   - Type Confusion
4. Data and Code
   - Scripting
   - SQL Injection

Overview

1. Introduction
   - Security and Reliability
   - Malware
   - Change in Environment
2. Broken Abstractions
   - Numbers and Characters
   - Function Composition
3. Memory Management
   - Memory Structure
   - Overruns
   - Type Confusion
4. Data and Code
   - Scripting
   - SQL Injection

Security and Reliability

- As long as our computer is offline, used only by ourselves, and we don't add any accessories (e.g. USB devices [Sch14]), then we don't have any problems.
- Problems start to occur when other users start using our software (in some way), then input to our programs isn't necessarily what we expect.

Security and Reliability

- Software reliability concerns software quality in the sense of accidental failures, i.e. the assumption that input is benign.
- Software security concerns software quality in the sense of intentional failures, i.e. the assumption that input is malign.
- We will focus on the latter.

Malware

Definition (Malware)

Comes from *malicious software* and means software with a malicious intent.

Definition (Computer Virus)

A form of malware which has self-replicating code. It *infects* other programs by inserting itself into their program code, and in turn when these programs are run the virus payload is run to replicate even further.

Definition (Worm)

A form of malware which replicates itself, not by infection, but by copying itself to different disks, via networks, or even emailing itself automatically to everyone in the user's contact list.

Malware

Definition (Trojan Horse)

A form of malware which acts as a legitimate program but has hidden features which are malicious, e.g. a utility program which steals your login credentials in the background.

Definition (Logic Bomb)

A form of malware which resides doing nothing until a logical condition is satisfied, then it executes its malicious code – e.g. erasing all files etc.

Change in Environment

- Change is one of the dangers to security.
- There are systems which are designed to be secure, and actually are secure, but then . . .
- upgrades are needed, or not needed but wanted.
- This might come in the form of updating a component or utilising the system in an environment it wasn't designed for.

Overview

1. Introduction
   - Security and Reliability
   - Malware
   - Change in Environment
2. Broken Abstractions
   - Numbers and Characters
   - Function Composition
3. Memory Management
   - Memory Structure
   - Overruns
   - Type Confusion
4. Data and Code
   - Scripting
   - SQL Injection

Numbers and Characters

- Imagine we want to keep the user in the directory `''/A/B/C''`.
- Our program implements this by taking the name of the input file as input from the user.
- Then to access the file it opens `''/A/B/C/''+filename`.
- What if the user inputs
  `filename = ''../../../etc/passwd''`?
- Then this would evaluate to opening
  `/A/B/C/../../../etc/passwd`.

Numbers and Characters

- Fine, we ban the string ’’../’’.
- Then what about ’’..%c0%af..’’?

Numbers and Characters

- All character representations in the computer comes in the form of different encodings, e.g. UTF-8 encoding.
- The decoders might be programmed differently, some takes into account the errors in different encoders to compensate – and this can be exploited.
- Where the encoding is done can also be exploited.

Numbers and Characters

- Integer overflows is another problem.
- Consider the following example.

```
1  char buf[128];
2
3  void
4  combine( char *s1, size_t len1, char *s2,
       size_t len2)
5  {
6    if ( len1 + len2 + 1 <= sizeof(buf) ) {
7      strncpy( buf, s1, len1 );
8      strncat( buf, s2, len2 );
9    }
10 }
```

Numbers and Characters

- Let `len2` be very long, say $2^{32} - 1$, i.e. `len2 = 0xffffffff`.
- Now we have $\texttt{len1} + \texttt{len2} + 1 \pmod{2^{32}} = \texttt{len1} + 2^{32} - 1 + 1 \pmod{2^{32}} = \texttt{len1} \pmod{2^{32}} < \texttt{sizeof(buf)}$.

Function Composition

- The login(1) and rlogin(1) composition bug was found in Linux and AIX systems which didn't check the syntax of the username.
- The syntax of login(1) is `login [-p] [-h host] [[-f] user]`.
- The syntax of rlogin(1) is `rlogin [-l user] machine`.
- rlogin(1) connects to the machine and runs `login user machine`.
- However, the user could be chosen to be "-froot".

Overview

1. Introduction
   - Security and Reliability
   - Malware
   - Change in Environment
2. Broken Abstractions
   - Numbers and Characters
   - Function Composition
3. Memory Management
   - Memory Structure
   - Overruns
   - Type Confusion
4. Data and Code
   - Scripting
   - SQL Injection

Memory Structure

- We have the code of the program.
- We have some program data.
- We have a stack growing downwards.
- We have a heap growing upwards.

Overruns

- Buffer overruns
- Stack overruns
- Heap overruns
- All variables in a program use storage from either the stack or heap.

Overruns

```
 1 int
 2 login( void )
 3 {
 4   char correct_password[] = "swordfish";
 5   char user_password[16] = {0};
 6
 7   printf( "user␣password:␣");
 8   fscanf( "\%s", user_password );
 9
10   if ( !strcmp( correct_password, user_password
       ) )
11     return 0;
12   return 1;
13 }
```

Type Confusion

- There are some problems in object-oriented languages too.
- Trick the system to point to a different memory location.
- Thus a write using one type actually modifies something believed to be of another type somewhere else.

Overview

1. Introduction
   - Security and Reliability
   - Malware
   - Change in Environment
2. Broken Abstractions
   - Numbers and Characters
   - Function Composition
3. Memory Management
   - Memory Structure
   - Overruns
   - Type Confusion
4. Data and Code
   - Scripting
   - SQL Injection

Scripting

```
1    cat thefile | mail addresses
```

- What happens with the address `foo@bar.org | rm -Rf /`?

SQL Injection

```
1    \$sql = "SELECT * FROM client WHERE name =
         '\$name'"
```

- Insert the name `Eve' OR 1=1--`.
- This will get a totally different meaning.

Referenser

[Sch14] David Schneider. “USB Flash Drives Are More Dangerous Than You Think”. In: *IEEE Spectrum* (Aug. 2014). URL: http://spectrum.ieee.org/tech-talk/computing/embedded-systems/usb-flash-drives-are-more-dangerous-than-you-think.