CS 465

Buffer Overflow

Slides by Kent Seamons and Tim van der Horst
Last Updated: Nov 11, 2011

Buffer Overflow

- The most common security vulnerability
- Root cause
  - Unsafe programming languages
- What areas of process memory are vulnerable to a buffer overflow?
  - Stack
  - Heap
  - Code/Data

Stack Smashing Attack

- A specific type of buffer overflow attack
- How does it work?
  - During a function call, the return address is pushed on the stack
  - An attacker overflows a buffer (local variable)
  - The return address on the stack is overwritten to point to an existing function or to injected code
  - During the function return the instruction pointer is set to the new value stored on the stack, not the original stored value

Vulnerable Code Examples

This code snippet caused the Morris Worm (1988)

```
char buf[20];
gets(buf);
```

Vulnerable Code Examples

```
void foo(char *input) {
  //make a local working copy
  char buf[1024];
  strcpy(buf, input);
}
```

Limitations on the Attack

- Usually only get one chance
  - Usually makes the program crash after the buffer is overflowed
  - Remote attacker doesn't know the exact address location of the injected attack code
    - NOP Sled helps create a window of opportunity

Questions on Stack Smashing

- How does the stack normally operate during a function call/return?
- Describe how an attacker can inject code on the stack
- What is a NOP sled and how/why is it used in a stack smashing attack?
- What are the requirements for the format of the injected code?

Defenses

- Write correct code
  - Avoid vulnerable functions
  - Audit code – use analysis tools
  - Fuzz testing
- Non-executable buffers
  - Kernel patches make the stack non-executable
- Array bounds checking
  - Compile time or run-time checks
  - Use a type-safe language
- Code pointer integrity checking
  - Detect when a pointer is corrupted
  - StackGuard and PointerGuard
- Address space randomization (ASLR)

Stack Guard

- How does it work?
- What is a canary?
  - Terminator canary
  - Random canary
  - XOR canary

Questions

- What are the approaches to defend against a buffer overflow attack?
  - What are the pros/cons of each?
- Is a buffer overflow only useful for a remote attack?
- (True or False) Making the stack non-executable makes a stack smashing attack impossible?
- (True or False) If your web server is written in Java, it is not vulnerable to a stack smashing attack?
- What is the *principle of least privilege* and how does it relate to buffer overflow attacks?

Integer Manipulation Vulnerabilities

- Three main integer manipulations that can lead to security vulnerabilities
  - Overflow and underflow
  - Signed vs. unsigned errors
  - Truncation
- Reviewing Code for Integer Manipulation Vulnerabilities
  - http://msdn2.microsoft.com/en-us/library/ms972818.aspx