Lecture 1: Buffer Overflows
Once Upon a Time…

MS08-067: Vulnerability in Server service could allow remote code execution
Memory: A Quick Review

- Stack
- Heap
- Globals
- Code
- 0x00000000
The Stack

Procedure A

...call B...

Procedure B

...call C...

Procedure C

...

Stack

---

B Param 3
B Param 2
B Param 1
Ret Addr Ptr A
Stack Ptr A
B Local Var 1
B Local Var 2
C Param 2
C Param 1
RetAddr Ptr B
Stack Ptr B
Buffer Overflows
Stack Smashing

Procedure A

... call B ...

Procedure B

... Buffer[20]; ...

<table>
<thead>
<tr>
<th></th>
<th>B Param 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B Param 2</td>
</tr>
<tr>
<td></td>
<td>B Param 1</td>
</tr>
<tr>
<td></td>
<td>New Ret Addr Ptr</td>
</tr>
<tr>
<td></td>
<td>New Stack Ptr</td>
</tr>
<tr>
<td></td>
<td>Overflow Buffer</td>
</tr>
<tr>
<td></td>
<td>B Local Var 2</td>
</tr>
</tbody>
</table>
Canaries

Procedure A

... call B ...

| B Param 3 |
| B Param 2 |
| B Param 1 |
| Ret Addr Ptr |
| Stack Ptr |
| Overflow Buffer |
| B Local Var 2 |

Procedure B

... ...
Buffer[20]; ...

Memory: A Quick Review

Stack

Heap

Globals

Code

0x00000000
The Heap

INTERNAL_SIZE_T prev_size;    /* size of prev chunk (if free) */
INTERNAL_SIZE_T size;         /* size of chunk */

struct chunk * fd;            /* double links -- used only if free */
struct chunk * bw;
Heap Smashing

INTERNAL_SIZE_T prev_size;  /* size of prev chunk (if free) */
INTERNAL_SIZE_T size;       /* size of chunk */

struct chunk * fd;          /* double links -- used only if free */
struct chunk * bw;

<table>
<thead>
<tr>
<th>Chunk C (in use)</th>
</tr>
</thead>
<tbody>
<tr>
<td>size C (01001-000)</td>
</tr>
<tr>
<td>prev_size C (00110)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Exploit code</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>fake bk B</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>fake fw B</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>size B (00110-000)</td>
</tr>
<tr>
<td>prev_size B (NULL)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Chunk A (free)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>bk A</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>fw A</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>size A (01101-001)</td>
</tr>
<tr>
<td>prev_size A</td>
</tr>
</tbody>
</table>
Memory Tagging

\[ W \Theta X \]
Return-into-libc

Procedure A
...
call B
...
exec
...
...

Procedure B
...
...
Buffer[20];
...

| \\sh\\0" |
| "\\bin |
| String Ptr |
| Fake Ret Addr Ptr |
| Fake Stack Ptr |
| New Ret Addr Ptr |
| New Stack Ptr |
| OverFlow Buffer |
| B Local Var 2 |
Address Space Layout Randomization
Language Support

```c
int a[20];
for(int i=0; i<max; i++){
    a[i]=0;
}
```
Why Buffer Overflows?

- Still a major problem
- Low level details can impact security
- Capabilities motivate threat models
- Example arms race
Course Logistics

• Class: F 10:10-11:25, Hollister 314
• Website: www.cs.cornell.edu/courses/5431/2017sp

• Instructor: Eleanor Birrell
  • eleanor@cs.cornell.edu
  • Gates Hall 441
  • Office Hours: M 4-6pm
Practicum = Course Project

• Build and secure a software system (with non-trivial security functionality)

• You choose and design your system!
  - Authentication. The system must authenticate its clients (humans, machines, and/or programs). Might include user registration, passwords, two-factor, generation of secrets, distribution of keys, etc.
  - Authorization. The system must enforce some authorization policy to control some subset of its operation.
  - Audit. The system must provide infrastructure for audit or other means of establishing accountability for actions.
  - Confidentiality and Integrity. The system must involve information that resides in long-term storage or that is transmitted over a network. The system’s mission must require that information to be kept secret and/or be protected from corruption.
tldr;

If it has:

1) More than one computer
2) More than one user
3) Some data somebody cares about

then it is probably ok
Project Schedule

- Six Project Milestones
  - Milestone 0: Charter Due: February 8
  - Milestone 1: Security Goals Due: February 17
  - Milestone 2: Prototype Due: March 8
  - Milestone 3: Authentication Due: March 22
  - Milestone 4: Authorization Due: April 28
  - Milestone 5: Final Due: May 10

- Demos: Milestone 2, Milestone 3, Milestone 4

- Final Project Presentations
Conficker, again