Stack Overflow

COMP620
“There are two kinds of people in America today: those who have experienced a foreign cyber attack and know it, and those who have experienced a foreign cyber attack and don’t know it.”

Frank Wolf
Adobe Acrobat and Reader Vulnerabilities

Original release date: June 10, 2009
Last revised: --
Source: US-CERT

Systems Affected

* Adobe Reader versions 9.1.1 and earlier
* Adobe Acrobat versions 9.1.1 and earlier

Overview

Adobe has released Security Bulletin APSB09-07, which describes several buffer overflow vulnerabilities that could allow a remote attacker to execute arbitrary code.
Buffer Overflows Are the Major Threat

• Buffer overflows are the greatest security vulnerability

• When a security alert contains the phrase “The most severe of these vulnerabilities allows a remote attacker to execute arbitrary code.”, the underlying problem is probably a buffer overflow

• The Morris worm (the first Internet worm) spread in part by exploiting a stack buffer overflow in the Unix finger server

• Many students do not know what they are
Critical Vulnerabilities

Vulnerabilities by Year

What will this C++ program do?

```cpp
void examplefunc() {
    int stuff = 0;
    char info[4];
    int i;
    for (i = 0; i < 7; i++) {
        info[i] = stuff++;  
    }
}
```

A. Compiler Error
B. Run time buffer overflow error
C. Corrupt data
D. Data execute exception
Basic Buffer Overflow

boolean rootPriv = false;
char name[8];
cin >> name;

• When the program reads the name “Smith”

| S | m | i | t | h |   | false |

char name[8] rootPriv
Basic Buffer Overflow

boolean rootPriv = false;
char name[8];
cin >> name;

• When the program reads the name “Armstrong”
Stack Overflow

• A stack overflow exploit occurs when a user enters data that exceeds the memory reserved for the input

• The input can change adjacent data or the return address on the stack

```c
char myStuff[4];
```

Program Stack
The Other Stack Overflow

• The phrase “Stack Overflow” also applies to programs whose stack grows to exceed the available memory

• These stack overflows are often generated by runaway recursion or allocated large data structures on the stack

• This form of stack overflow is not usually the result of a malicious attack

*This is a problem for another day*
Program Memory Organization

- Heap
- Stack
- Global data
- Program instructions

Intel method
Byte Ordering

• Some systems store the least significant byte first (Little Endian). Others store the most significant byte first (Big Endian)

• The decimal number 258 (0100000010₂) would be stored in as a 32 bit binary number

Big Endian

\[
\begin{bmatrix}
00000000 & 00000000 & 00000001 & 00000010 \\
\text{byte 0} & \text{byte 1} & \text{byte 2} & \text{byte 3}
\end{bmatrix}
\]

Little Endian (Intel processor)

\[
\begin{bmatrix}
00000010 & 00000001 & 00000000 & 00000000 \\
\text{byte 0} & \text{byte 1} & \text{byte 2} & \text{byte 3}
\end{bmatrix}
\]
Stack Review

• Consider the function

```c
void thefunc( float &dog, int cat ){
    char cow[4];
}
```

• that is called by the main program

```c
int bass = 5;
float carp = 7.0;
float *trout = &carp;
thefunc( trout, bass );
```
Stack for Call

- push bass

\texttt{thefunc( trout, bass );}
Stack for Call

• push bass
• push trout

thefunc( trout, bass );
Stack for Call

• push bass
• push trout
• push return address

thefunc( trout, bass );
Stack for Call

- push bass
- push trout
- push return address
- push frame pointer

<table>
<thead>
<tr>
<th>5 (value of bass)</th>
</tr>
</thead>
<tbody>
<tr>
<td>address of carp</td>
</tr>
<tr>
<td>return address</td>
</tr>
<tr>
<td>Addr of last frame</td>
</tr>
</tbody>
</table>

```
thefunc( trout, bass );
```
Stack for Call

- push bass
- push trout
- push return address
- push frame pointer
- Allocate space for local variable, cow[4]

```c
thefunc( trout, bass );
```
Overflowing Local Variables

• On an Intel processor (*and many others*) the stack is extended to lower addresses
• If you address beyond a local variable, you will overwrite the return address.

<table>
<thead>
<tr>
<th>5 (value of bass)</th>
</tr>
</thead>
<tbody>
<tr>
<td>address of carp</td>
</tr>
<tr>
<td>return address</td>
</tr>
<tr>
<td>Addr of last frame</td>
</tr>
<tr>
<td>cow[4]</td>
</tr>
</tbody>
</table>
Hacking the Stack

• If a program does not properly check array bounds, it may be possible to give the program specially crafted input that overwrites the return address with a binary value

• \texttt{cow[8] to cow[11]} are the return address
Hacking the Stack

• If a program does not properly check array bounds, it may be possible to give the program specially crafted input that overwrites the return address with a binary value

• The return address can be changed to the address of a function in the program

• Function parameters can also be put on the stack

<table>
<thead>
<tr>
<th>5 (value of bass)</th>
</tr>
</thead>
<tbody>
<tr>
<td>address of carp</td>
</tr>
<tr>
<td>return addr='?'</td>
</tr>
<tr>
<td>Frame ptr = ‘wxyz’</td>
</tr>
<tr>
<td>cow[4]=‘abcd’</td>
</tr>
</tbody>
</table>
Loading Malicious Code

• A long input to a short buffer might also contain binary machine language
• The return address can be overwritten to cause the program to jump to the newly loaded machine language when it returns
What is the most evil thing a stack overflow exploit can do?

A. Change the return address to call some function
B. Load binary machine language
C. Crash the program
D. Alter data
Research Paper

• Your research paper is due by 11:00am on Wednesday
• Technical requirements
  – 3 – 5 pages in length
  – single spaced or space and a half
  – 12 point font with one inch margins on all sides
  – in Microsoft Word or PDF format
Research Paper

• The paper MUST be your opinion on the topic
• Your opinion must be backed up with facts
• At least 4 reliable references
• Other references are allowed
Plagiarism

• Whenever you use the ideas of another, you must cite the source
• Direct quotes must be “quoted”
  – Beware of citations in quotes
• Whenever you use another’s idea, indicate the source

John Doe [4] says this is ridiculous.
Preventing Stack Overflow Exploits

• Better software engineering
• Avoid dangerous functions
• Language choice
• Compiler tools (Stack Guard)
• Analysis tools
• Execution Prevention
• Randomize stack location
It’s All About You

You are the cause

• As a programmer, insecure code that you may write can leave users vulnerable to attack

You are the solution

• By writing secure code, you can protect users from attack
Stack Canaries

- A stack canary is a random number placed on the stack between the user data and the return address.
- Overflowing the local variable and changing the return address will also change the stack canary.
- Before returning, the program checks the canary value.

<table>
<thead>
<tr>
<th>Stack canary</th>
<th>5 (value of bass)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>address of carp</td>
</tr>
<tr>
<td></td>
<td>return address</td>
</tr>
<tr>
<td>Addr of last frame</td>
<td></td>
</tr>
<tr>
<td>cow[4]</td>
<td></td>
</tr>
</tbody>
</table>
Data Execution Prevention

• Most newer processors have a bit in the page table that inhibits instruction fetches from that page
• Operating systems can set data execution prevention for stacks
• This prevents the program from executing machine language loaded on the stack by an exploit
• This does not prevent programs from overwriting the return address
Random Stack Location

• Microsoft Windows locates a programs stack at a random address since Windows Vista
• Each time the program is executed, the stack is at a different address
• Hackers cannot learn stack addresses from a previous execution of the program
Real Overflow Attack

• A handout gives step-by-step instructions on how to perform a real stack overflow
• This attack puts machine language on the stack to print your name
• The exercise uses a virtual machine running Windows XP