

# **Buffer Overflows**



***The computer vulnerability  
of the decade***

# Nature of the problem

---

- exploit a lack of bounds checking on the size of input
- most common to attack buffers allocated on the stack (stack smashing)
- writing data past the end of an allocated array, the attacker can make arbitrary changes to program state

# Nature of the problem



- two mutually dependent goals:
  - inject executable attack code to e.g. produce root shell (attack targets are usually root-privileged daemons)
  - change return address pointer to attack code

# History



- Nothing new - dates back to 1960s
- Internet Worm (1988)
- Most common and widely exploited vulnerability
- Often the result of performance-oriented, careless programming in C/C++

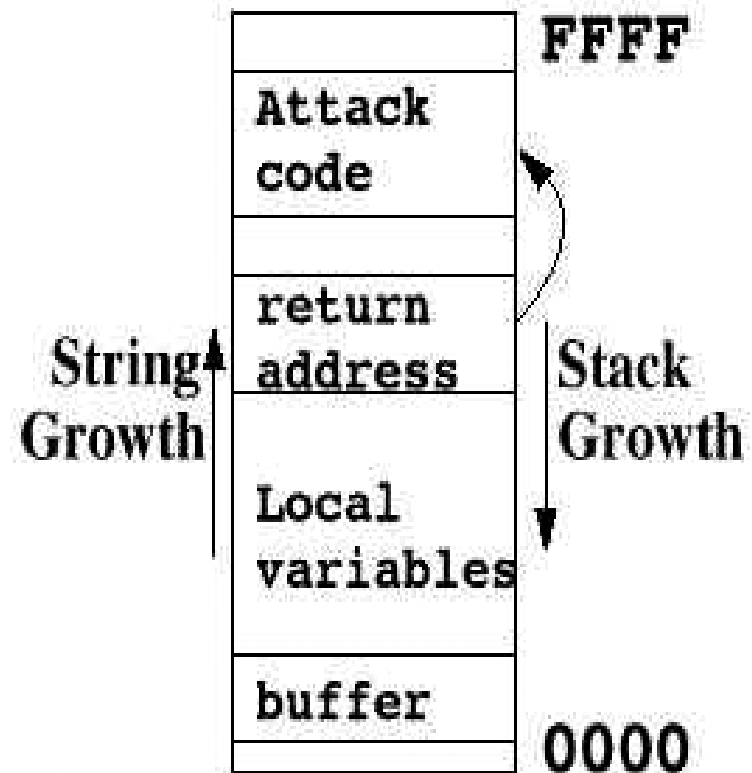
# History



- Attacks formerly based on reverse engineering, now fairly easy with cook books (Aleph One 1996)
- Traditionally more exploits seen for Unix-based systems as Windows API more complex and more recent

# Overview

- Carries on overwriting past the end of the buffer
- Overwrites the return address so it points to the attack code



# THE WORM

The first well-known case  
of buffer overflow

# First two means of attack

---

## *Password Guessing*

- Guess password from local account name
- Try to use rexec to use account password to open other host accounts
- If it fails, use rsh to try and bypass password identification

## *Sendmail*

- Sendmail has a bug when compiled with the DEBUG flag
- The worm sends a command instead of a mail address to infect the machine by compiling the code

# 3rd means of attack: *finger*

---

- The finger service is subject to buffer overflows.
- The finger program is run where the variables set in the request are used as arguments.
- Requests are 512 bytes long. But finger handles them with `gets()`, which does not check for overflows
- The worm sends a 536 bytes request. The extra bytes contain instructions to run the *sh* command interpreter and go back to the VAX code in the request buffer to ensure infection.

# Finger request code

Part of the buffer overflow that starts the sh session.

```
if (i >= 6)
    return 0;
for(i = 0; i < 536; i++)
    buf[i] = '\0';
for(i = 0; i < 400; i++)
    buf[i] = 1;
for(j = 0; j < 28; j++)
    buf[i+j] =
    "\335\217/sh\0\335\217/bin\320^Z\335\0\335\0\335Z\335\003\320^\0
    \274;\344\371\344\342\241\256\343\350\357\256\362\351"[j];
```

# Microsoft Exchange



- Microsoft Exchange 5.5
  - Supports LDAP version 3
- LDAP
  - Lightweight Directory Access Protocol (RFC 1777)
  - Operations on directory database are:
    - | READ
    - | SEARCH
    - | ADD
    - | REMOVE
  - Microsoft Exchange only supports "READ" operation
  - LDAP Directory service allow interoperability to 3rd party LDAP compliant clients

# The Exploit



- Discovered on 15 March 1999
- Buffer overflow exploited using
  - A malformed bind request during LDAP binding process
- Methodology
  - Creating and sending a particular type of invalid LDAP bind packet
    - *Crashes* the LDAP services
  - Creating and sending a large malformed LDAP bind packet
    - *Executes* arbitrary code
- Recommendations
  - Applying Microsoft's Exchange Service Pack
  - Use Firewall to prevent external attack

# Netscape Web Server



- Netscape Enterprise Server and Netscape FastTrack Server
- Components
  - Administration Server
    - | Lightweight HTTP Server
    - | Administration
    - | Runs as 'root' on Unix, 'System' on Windows NT
  - HTTP Server
    - | Internet Web Server
    - | Runs as 'nobody'

# The Exploit



## ■ Products Vulnerable

- Netscape Administration Server version 3.0 - 3.6
- Netscape Enterprise Server version 3.5.1 - 3.6sp2
- Netscape FastTrack Server version 3.01

## ■ Problem

- Occurs when HTTP Basic Authentication is used

## ■ Methodology

- Sending username and password that is longer than 508 characters
  - Crashes the daemon
  - Executes arbitrary code

## ■ Recommendations

- Upgrade to Netscape Enterprise Server 4.0sp2

# CERT Advisories from 1997



- HP-UX newgrp Buffer Overrun
- talkd Vulnerability
- MIME Conversion Buffer Overflow in Sendmail Versions 8.8.3 and 8.8.4
- Vulnerability in rlogin/term
- Vulnerability in IMAP and POP
- Vulnerability in Natural Language Service
- Vulnerability in libXt
- Vulnerability in xlock
- Vulnerability in suidperl (sperl)
- Vulnerability in the at(1) program
- lpr Buffer Overrun Vulnerability
- SGI Buffer Overflow Vulnerabilities
- Buffer Overflow Problem in rdist
- Buffer overrun Vulnerability in Count.cgi cgi-bin Program
- Buffer Overrun Vulnerability in statd(1M) Program

# Countermeasures



- Write good code!
- Run-time bounds checking, e.g. Java, Ada
- Non-executable stack
- StackGuard

# StackGuard



- Puts a 'canary' value on the stack next to the return address
- Check this before returning from the function
- Attacker would have to overwrite 'canary' in order to change return address
- Implemented as patch to C compiler

# Conclusion



- Very common problem
- DoS attacks easiest
- Generic attack tools and cook books make exploit attacks easier
- Can be avoided
- Tools are available to help