Feb 9, 2005 -- Lecture 10

22C:169
Computer Security

Douglas W. Jones

Department of Computer Science

Worms, channels

Worms

John Bruner's *Shockwave Rider*, 1975
First Implemented, Xerox PARC, 1978

Self reproducing code

Spreads between network hosts
Spread via network links

Requirements

Read from link executes code

Deliberately or not

Deliberate worm

```
# Unix shell script in file f
setenv  host `randomhost`
rcp     f $(host):f
rsh     $(host) f
# insert payload here
rm      f
```

Within a secure setting, worms are useful!

The Xerox Worms

Shoch and Hupp, CACM, March 1982

Screen-saver augmentation:

accept application downloads
kills application on keypress or click

The existential worm:

Search for running screensavers
Download self

Many Useful Payloads

How can worms invade?

Error in network interface that allows
- *injection of code where data intended*
  - Buffer Overflow Attack

Debugging interfaces left in place
- *Beware: Sensible development tools can be dangerous in production*

Morris' Internet Worm

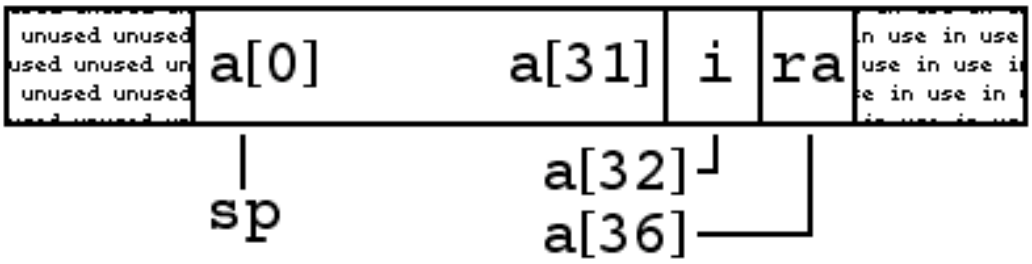
> There may be a virus loose on the internet.
> Andy Sudduth of Harvard,
> 34 minutes after midnight, Nov. 3, 1988

1: Try to infect hosts in same domain
   `/etc/hosts.equiv` *usually lists them*
2: Try stupid passwords
   dictionary attack plus user name tricks
3: Buffer overflow attack on **`fingerd`**

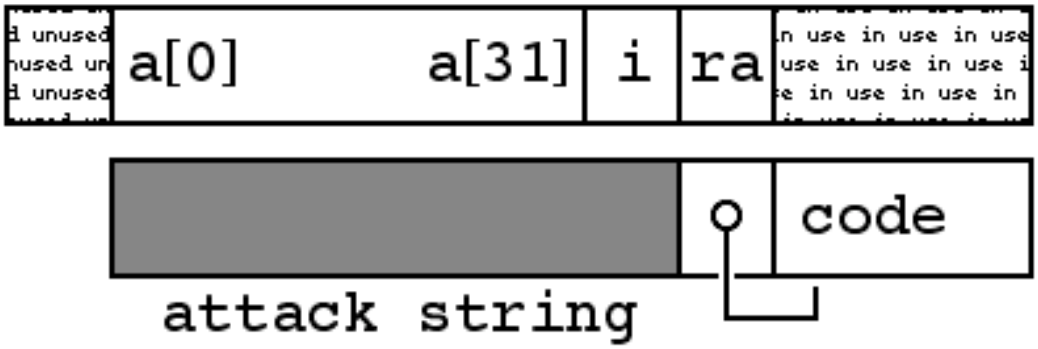
4: Attack **`sendmail`** with debug option

Buffer Overflow Vulnerability:

```
int f( int i )
{
   char a[32];
   gets( a );
   return lookup(a);
}
```



Buffer Overflow Attack



NOTICE:

Buffer overflow attack injects

executable object code where text is expected!

Only works if attacker knows (or guesses)

instruction set of target machine

The domain **`uiowa.edu`** survived because

Our gateway was an Encore Multimax

NOTICE:

`Sendmail` attack relied on

Debugging code
that allowed injecting shell commands

Again, attack only works if

Debugging option left active
Many users of same **`sendmail`**
Attacker knows scripting language

monocultures are vulnerable

Another example, the `christma` "exec"

December 9, 1987

Attacked IBM mainframe E-mail

Written in REXX scripting language

Sent as a Christmas Card on BITNET

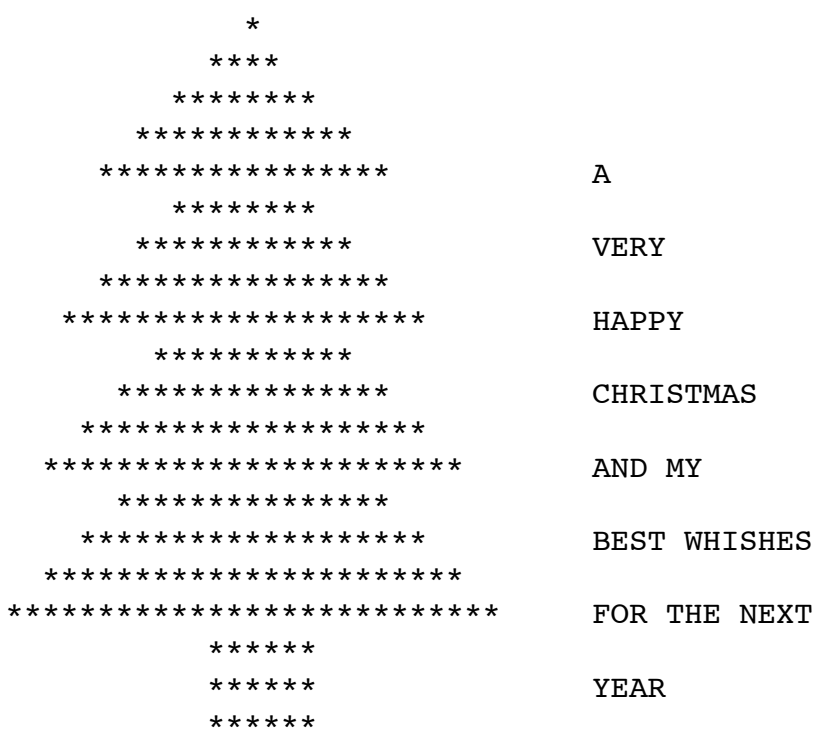
`"browsing this file is no fun ...`
`just type CHRISTMAS from cms"`

This is a Trojan Horse Attack

The Christma Exec

Sent by a German CS undergrad

Innocent of evil intent!



The Christma Exec

Only a mild problem on BITNET

Roughly half IBM mainframes

Remainder: mostly DEC equipment

Not a monoculture!

Escape from BITNET to IBM's Internal net

Disaster! all hosts identical!

Monoculture!

Protection Domain

Definition

The set of objects on which a program may operate

Problem

Control of interdomain communication

Example

Worms are a threat when code is passed between domains

Interdomain Channels

Overt channels

Those that are intended

messages

function calls

Covert channels

Those not intended in system design

covert communications

secret interfaces