Computer Worms

A computer worm is a self-replicating computer program. It uses a network to send copies of itself to other nodes on the network. It may do this without any user intervention. Unlike a computer virus, a computer worm does not need to attach itself to an existing program. Worms tend to harm the network by consuming network bandwidth where viruses infect or corrupt files on the targeted computer.

Def: A computer worm is a program that copies itself from one computer to another computer.

e.g.

The Morris worm was one of the first computer worms distributed via the Internet. It is considered the first worm and was certainly the first to gain significant mainstream media attention. It was written by a graduate student at Cornell University, Robert Morris Jr, and launched on November 2, 1988 from MIT. The worm was released from MIT to disguise the fact that the worm originally came from Cornell. Morris ended up with a $10,050 fine, 3 yrs suspended jail sentence, and 400 hours of community service.

What the worm did:

1. determines where it could spread
2. spreads its infection
3. remains undiscovered & undiscoverable

What its effect was:

1. resource exhaustion – due to program flaw
2. 2nd order effect: disconnection of many systems from the network
3. 3rd order effect: isolation and inability to perform necessary work
   6,000 major UNIX (SUN and Berkeley) installations down
   estimated damage $100,000 - $97,000,000

How it worked:

1. password guessing via rexec/rsh
   system error - /etc/passwd
   user error – common passwords
2. fingerd buffer overflow
3. sendmail trap door

How it spread:

1. target machine, loader, get rest of worm
   one time password
2. undiscovered & undiscoverable
   encrypt, delete from disk, change name
3. CERT – Computer Emergency Response Team

Types of Computer Worms

Email Worms

Spread via email messages. Typically the worm will arrive as email, where the message body or attachment contains the worm code, but it may also link to code on an external website. Poor design aside, most email systems requires the user to explicitly open an attachment to activate the worm, but "social engineering" can often successfully be used to encourage this; as the author of the "Anna Kournikova" worm set out to prove. Once activated the worm will send itself out using either local email systems (e.g. MS Outlook services,

Windows MAPI functions), or directly using SMTP. The addresses it sends to are often harvested from the infected computers email system or files. Since Klez.E in 2002, worms using SMTP typically fake the sender's address, so recipients of email worms should assume that they are not sent by the person listed in the 'From' field of e-mail message (sender's address).

Instant Messaging Worms

The spreading used is via instant messaging applications by sending links to infected websites to everyone on the local contact list. The only difference between these and email worms is the way chosen to send the links.

IRC worms

Chat channels are the main target and the same infection/spreading method is used as above — sending infected files or links to infected websites. Infected file sending is less effective as the recipient needs to confirm receipt, save the file, and open it before the infection will take place.

File-sharing networks worms

File-sharing worms copies itself into a shared folder which is most likely located on the local machine. The worm will place a copy of itself in a shared folder under a harmless name. Now the worm is ready for download via the P2P network and spreading of the infected file will continue.

Internet worms

Internet worms are those that target low level TCP/IP ports directly, rather than going via higher level protocols such as email or IRC. A classic example is "Blaster" which exploited a vulnerability in Microsoft's RPC. An infected machine aggressively scans random computers on both its local network and the public Internet attempting an exploit against port 135 which, if successful, spreads the worm to that machine.

Rabbits & Bacteria

Some malicious logic multiplies so rapidly that resources become exhausted. This creates a denial of service attack.

Def: A *bacterium* or a *rabbit* is a program that absorbs all of some class of resource.

e.g.

Consider the shell script program below presented by Dennis Ritchie.

```
while true
do
   mkdir x
   chdir x
done
```

The above shell script will quickly exhaust either disk space or the inode tables.

Logic Bombs

Some malicious logic triggers on an external event such as a user logging in or the arrival of midnight. For example, a programmer may hide a piece of code that starts deleting files such as the salary database should she ever leave the company.

Def: A *logic bomb* is a program that performs an action that violates the security policy when some external event occurs.

e.g. In June 1992, a defense contractor General Dynamics employee, Michael Lauffenburger, was arrested for inserting a logic bomb that would delete vital rocket project data. It was alleged that his plan was to return as a highly paid consultant to fix the problem once it triggered. Another employee of the company stumbled upon the bomb before it was triggered. Lauffenburger was charged with computer tampering and attempted fraud and faced potential fines of $500,000 and jail time. He was ultimately fined $5,000.

e.g. In February 2000, Tony Xiaotong, indicted before a grand jury, was accused of planting a logic bomb during his employment as a programmer and securities trader at Deutsche Morgan Grenfell. The bomb had a trigger date of July 2000 and was discovered by other programmers in the company. Removing and cleaning up after the bomb allegedly took several months.

e.g. In June 2006 Roger Duronio, a disgruntled systems administrator for UBS PaineWebber was charged with using a "logic bomb" to damage the company's computer network and with securities fraud for his failed plan to drive down the company's stock with activation of the logic bomb. Duronio was later convicted and sentenced to 8 years and 1 month in prison, as well as a $3.1 million restitution to UBS.

e.g. In early 1980 a program posted on the USENET news network promised to make administering systems easier. Buried in the code was the following segment:

```
cd /
rm –rf *
```