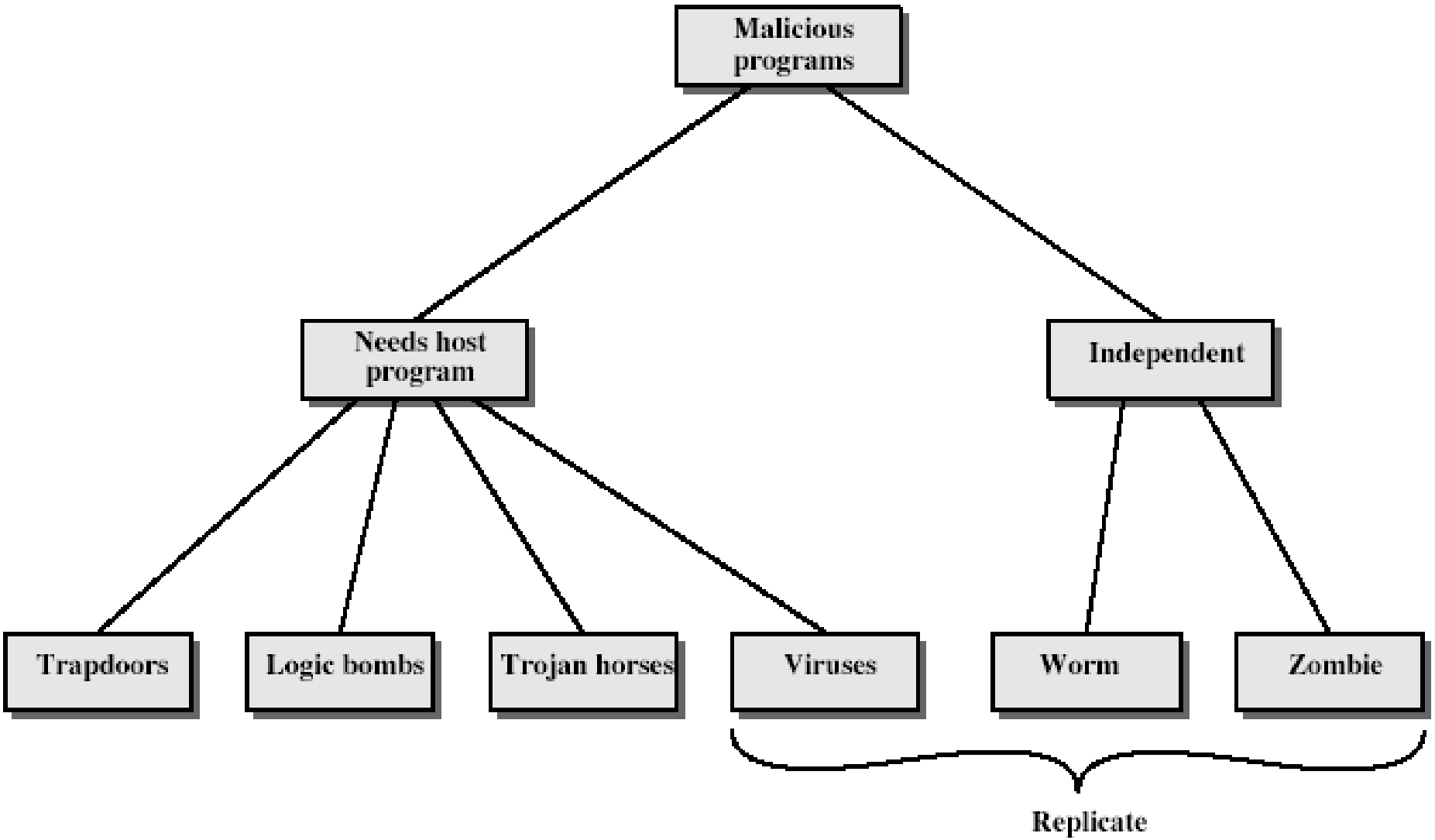
Outlines

- Mobile malcode Overview
- Viruses
- Worms

Mobile Malcode Overview

- Malicious programs which spread from machine to machine *without* the consent of the owners/operators/users
  - Windows Automatic Update is (effectively) consensual
- Many strains possible
  - Viruses
  - Worms
  - Compromised Auto-updates
    - No user action required, very dangerous

Malicious Software



Trapdoors (Back doors)

- Secret entry point into a program
- Allows those who know access bypassing usual security procedures
- Have been commonly used by developers
- A threat when left in production programs allowing exploited by attackers
- Very hard to block in O/S
- Requires good s/w development & update

Logic Bomb

- one of oldest types of malicious software
- code embedded in legitimate program
- activated when specified conditions met
  - eg presence/absence of some file
  - particular date/time
  - particular user
  - particular series of keystrokes
- when triggered typically damage system
  - modify/delete files/disks

Trojan Horse

- Programs that appear to have one function but actually perform another.
- Modern Trojan Horse: resemble a program that the user wishes to run - usually superficially attractive
  - eg game, s/w upgrade etc
- When run performs some additional tasks
  - allows attacker to indirectly gain access they do not have directly
- Often used to propagate a virus/worm or install a backdoor
- Or simply to destroy data

Zombie

- program which secretly takes over another networked computer
- then uses it to indirectly launch attacks
- often used to launch distributed denial of service (DDoS) attacks
- exploits known flaws in network systems

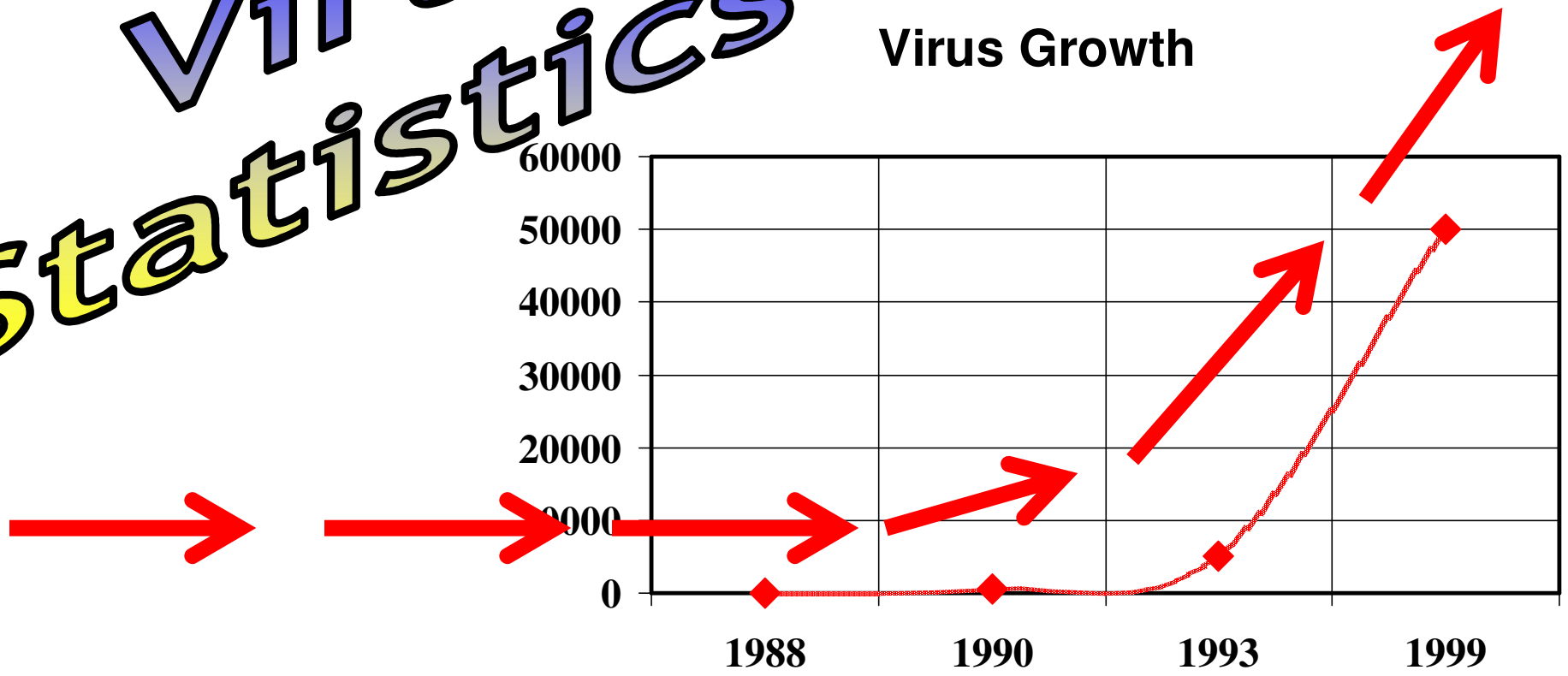
Outlines

- Mobile malcode Overview
- Viruses
- Worms

Viruses

- Definition from RFC 1135: *A virus* is a piece of code that inserts itself into a host, including operating systems, to propagate. It cannot run independently. It requires that its host program be run to activate it.
- On execution
  - Search for valid target files
    - Usually executable files
    - Often only infect uninfected files
  - Insert a copy into targeted files
    - When the target is executed, the virus starts running
- Only spread when contaminated files are moved from machine to machine
- Mature defenses available

Virus Statistics



- 1988: Less than 10 known viruses
- 1990: New virus found every day
- 1993: 10-30 new viruses per week
- 1999: 45,000 viruses and variants

Source: McAfee

Virus Operation

- virus phases:
  - dormant – waiting on trigger event
  - propagation – replicating to programs/disks
  - triggering – by event to execute payload
  - execution – of payload
- details usually machine/OS specific
  - exploiting features/weaknesses

Anatomy of a Virus

- Two primary components
  - Propagation mechanism
  - Payload
- Propagation
  - Method by which the virus spreads itself.
  - Old days: single PC, transferred to other hosts by ways of floppy diskettes.
  - Nowadays: Internet.

Structure of A Virus

```
Virus() {
  infectExecutable();
  if (triggered()) {
     doDamage();
  }
  jump to main of infected program;
}

void infectExecutable() {
  file = choose an uninfected executable file;
  prepend V to file;
}

void doDamage() { ... }
int triggered() { return (some test? 1 : 0); }
```

Virus Infectables

- Executable files: .com, .exe, .bat
- Macros
  - With macro languages the line between pure data files and executable files is blurring
  - An infected file might be attached to an E-mail
  - E-mail programs may use other programs (e.g., word) with macros to display incoming mail
- System sector viruses
  - Infect control sectors on a disk
    - DOS boot sectors
    - Partition (MBR) sectors
  - System sector viruses spread easily via floppy disk infections

Virus Infectables (cont'd)

- Companion viruses
  - Create a .com files for each .exe files
  - DOS runs COM files before EXE files
  - Relatively easy to find and eliminate
- Cluster viruses
  - Change the DOS directory info so that directory entries point to the virus code instead of the real program
  - Even though every program on the disk may be "infected", there is only one copy of the virus on the disk

Variable Viruses

- Polymorphic viruses
  - Change with each infection
    - Executables virus code changing (macros: var name, line spacing, etc.)
    - Control flow permutations (rearrange code with goto's)
  - Attempt to defeat scanners
- Virus writing tool kits have been created to "simplify" creation of new viruses
  - Current tool kits create viruses that can be detected easily with existing scanner technology
  - But just a matter of time …

Virus Detection/Evasion

- Look for changes in size
- Check time stamp on file
- Look for bad behavior
  - False alarm prone
- Look for patterns (byte streams) in virus code that are unique
- Look for changes in file checksum
- Compression of virus and target code
- Modify time stamp to original
- Do bad thing insidiously
- Change patterns – polymorphism
- Rearrange data in the file
- Disable anti-virus programs

More on Virus Detection

- Scanning
  - Depend on prior knowledge of a virus
  - Check programs before execution
  - Need to be regularly updated
- Integrity Checking
  - Read entire disk and record integrity data that acts as a signature for the files and system sectors
  - Use cryptographic computation technique instead of simple checksum

More on Virus Detection

- Interception
  - Monitoring for system-level routines that perform destructive acts
  - Good for detecting logic bomb and Trojan horse
  - Cannot depend entirely upon behavior monitors as they are easily bypassed.
- Combination of all three techniques can detect most viruses

Virus Recovery

- Extricate the virus from the infected file to leave the original behind
- Remove the redirection to the virus code
- Recover the file from backup
- Delete the files and move on with life

History of Viruses

First Wild Viruses Apple I/II/III: 1981

- Three viruses for the Apple machines emerged in 1981
  - Boot sector viruses
- Floppies of that time had the disk operating system (DOS) on them by default
  - Wrote it without malice

First PC Virus: Pakistani Brain Virus (1986)

- Written by Pakistani brothers to protect their copyright
  - Claim: infect only machines that had an unlicensed copy of their software
  - Boot sector, memory resident
  - Printed

"Welcome to the Dungeon (c) 1986 Basit * Amjad (pvt) Ltd. BRAIN COMPUTER SERVICES

730 NIZAB BLOCK ALLAMA IQBAL TOWN

LAHORE-PAKISTAN PHONE :430791,443248,280530.

Beware of this VIRUS.... Contact us for vaccination ..... !!"

Destructive Virus: Chernobyl (1998)

- Designed to inflict harm
  - Flash BIOS: would cause permanent hardware damage to vulnerable motherboards
  - Also overwrote first 2K sectors of each disk
    - Typically resulted in a loss of data and made it unbootable
- Previously believed that being benign was necessary for virus longevity
  - Chernobyl provided evidence to the contrary

Early Macro Virus: Melissa (1999)

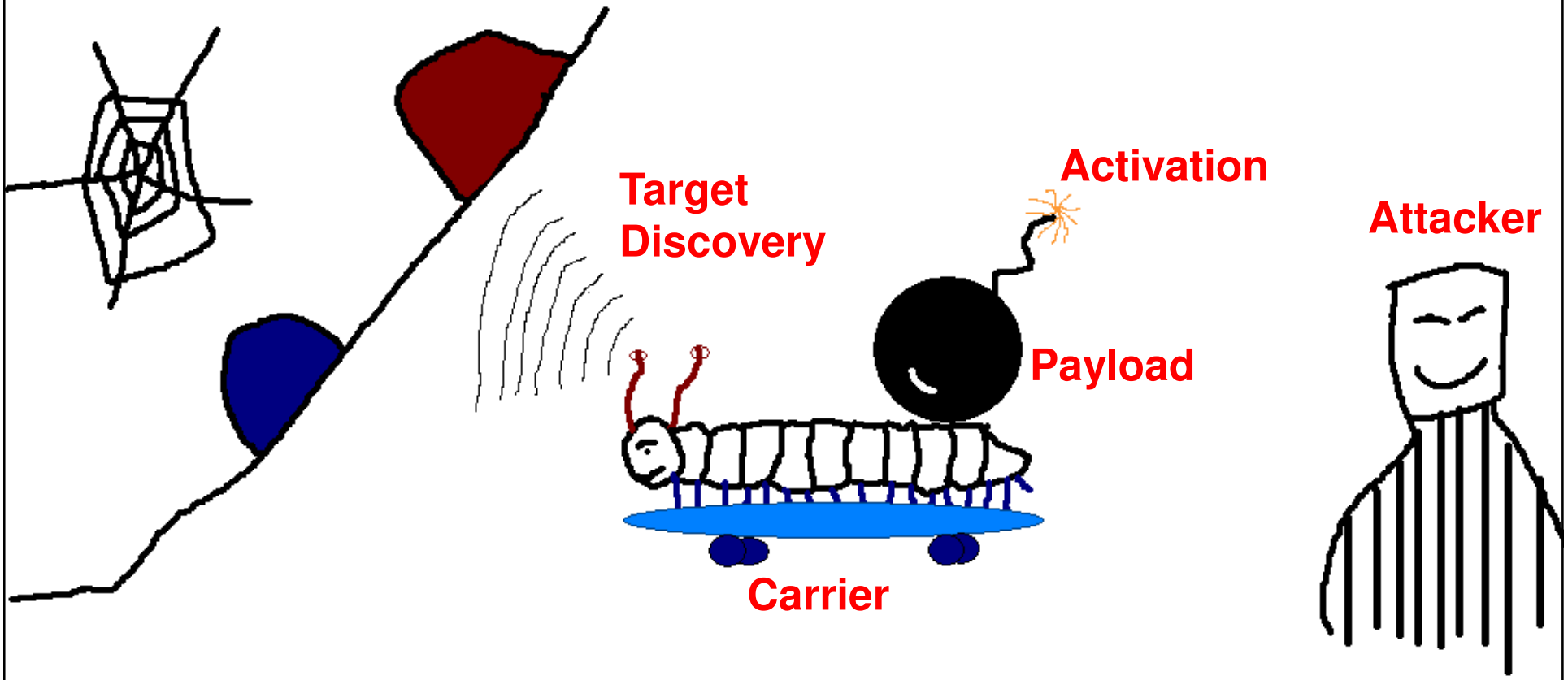
- Microsoft Word 97 Macro virus
- Target first 50 entries in Outlook's address book
- Adjusted subject "Important messages from _____"
- Points to attachment as a document requested
  - Contains a list of porn sites
- Macro security was greatly increased with Melissa

Outlines

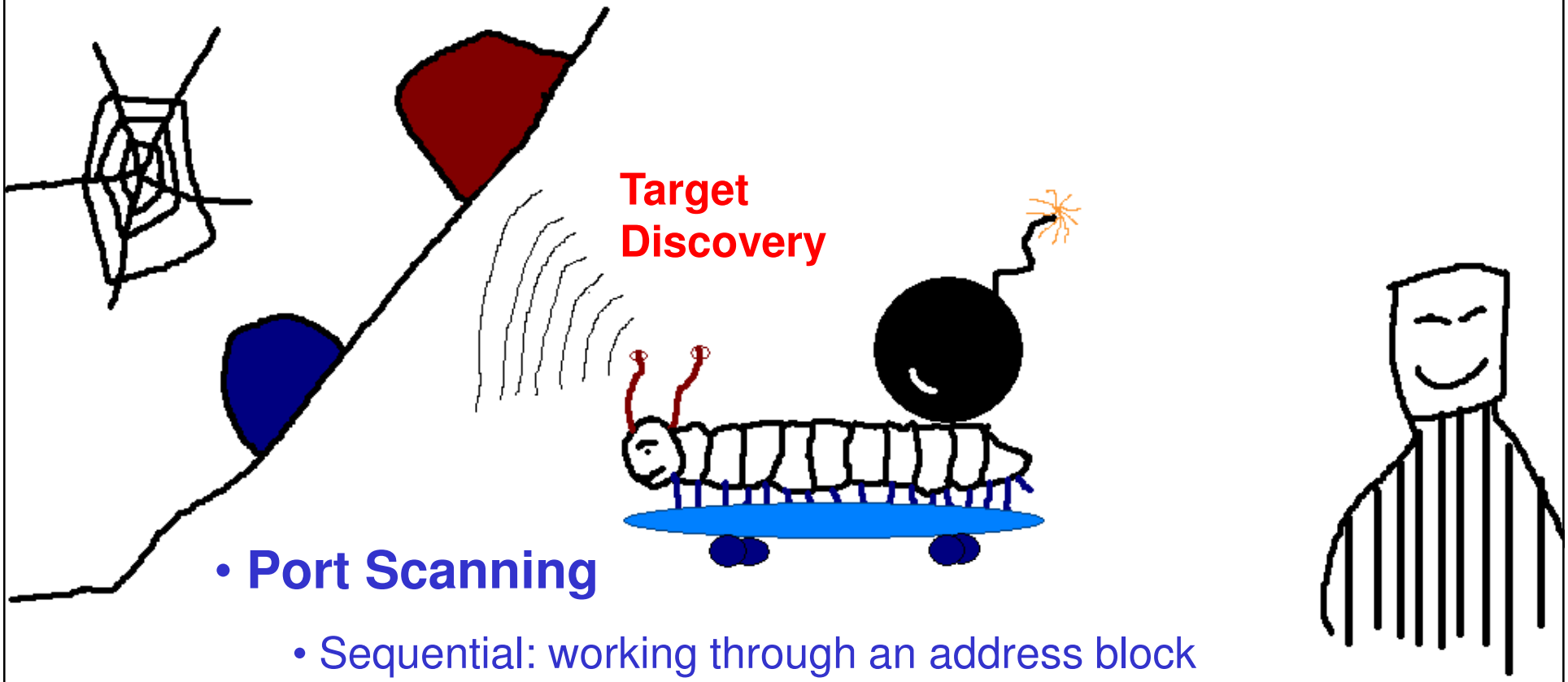
- Mobile malcode Overview
- Viruses
- Worms

Worms

- Autonomous, active code that can replicate to remote hosts without any triggering
  - Replicating but not infecting program
- Because they propagate autonomously, they can **spread much more quickly** than viruses!
- Speed and general lack of user interaction make them the most significant threats



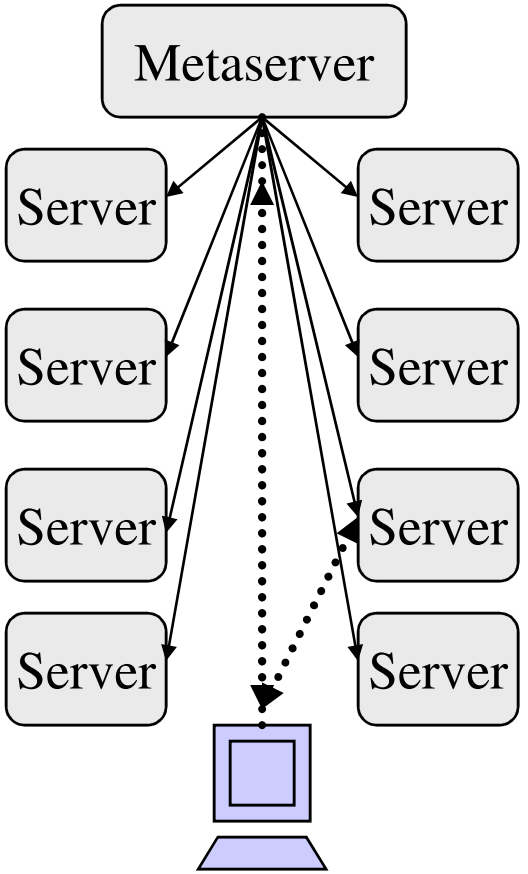
Worm Overview



- **Port Scanning**
  - Sequential: working through an address block
  - Random
- **Target Lists**
  - Externally generated through Meta servers
  - Internal target list
  - Passive worms

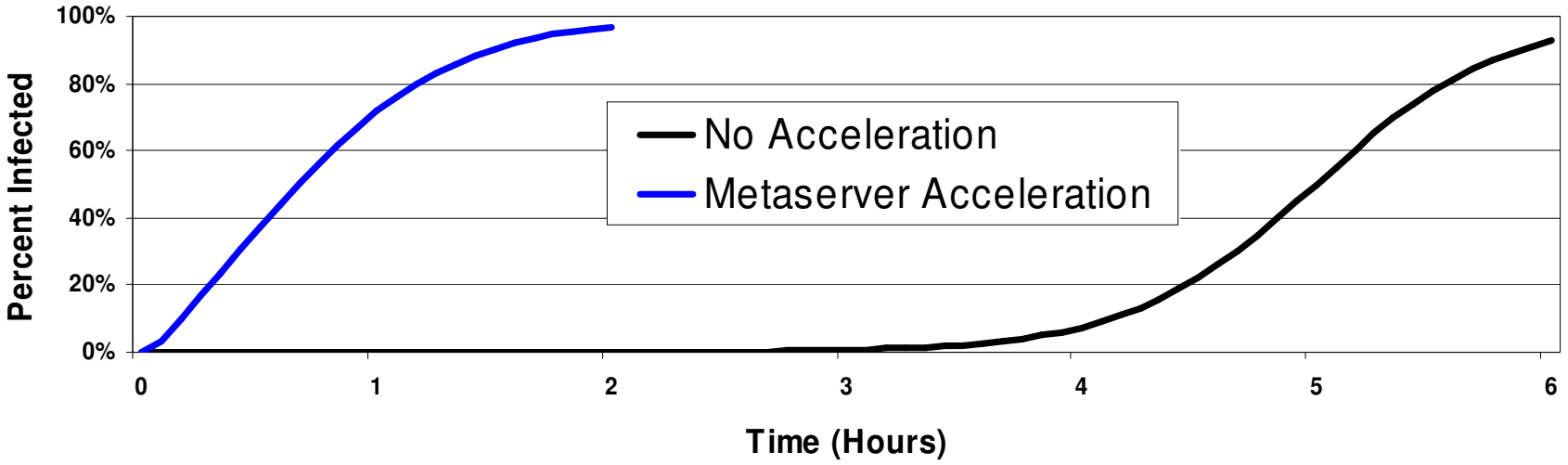
External Target Lists: Metaserver Worms

- Many systems use a "metaserver", a server for information about other servers
  - Games: Use as a matchmaker for local servers
  - Google: Query google to find web servers
  - Windows Active Directory: Maintains the "Network Neighborhood"
- Worm can leverage these services
  - Construct a query to find new targets
  - Each new victim also constructs queries
    - Creates a divide-and-conquer infection strategy
- Original strategy, not yet seen



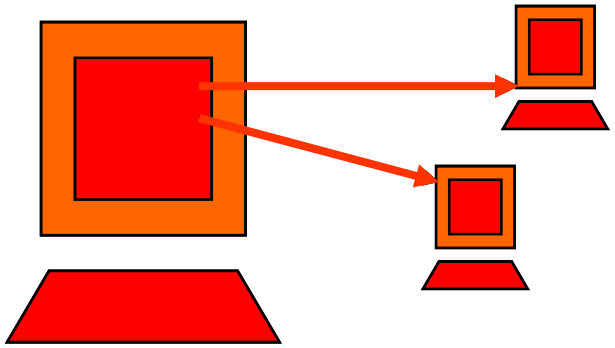
How Fast Are Metaserver Worms?

- Game Metaserver: Use to attack a small population (eg, all Half-Life servers)
  - ~1 minute to infect all targets
- Google: Use to enhance a scanning web worm
  - Each worm conducts initial queries to find URLs



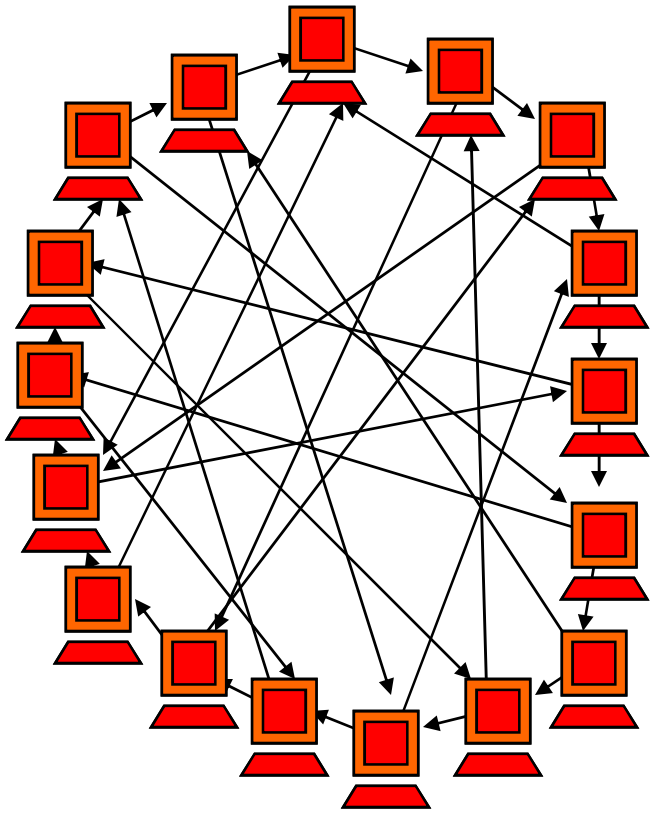
Internal Target Lists: Topological Information

- Look for local information to find new targets
  - URLs on disk and in caches
  - Mail addresses
  - .ssh/known_hosts
- Ubiquitous in mail worms
  - More recent mail worms are more aggressive at finding new addresses
- Basis of the Morris worm
  - Address space was too sparse for scanning to work



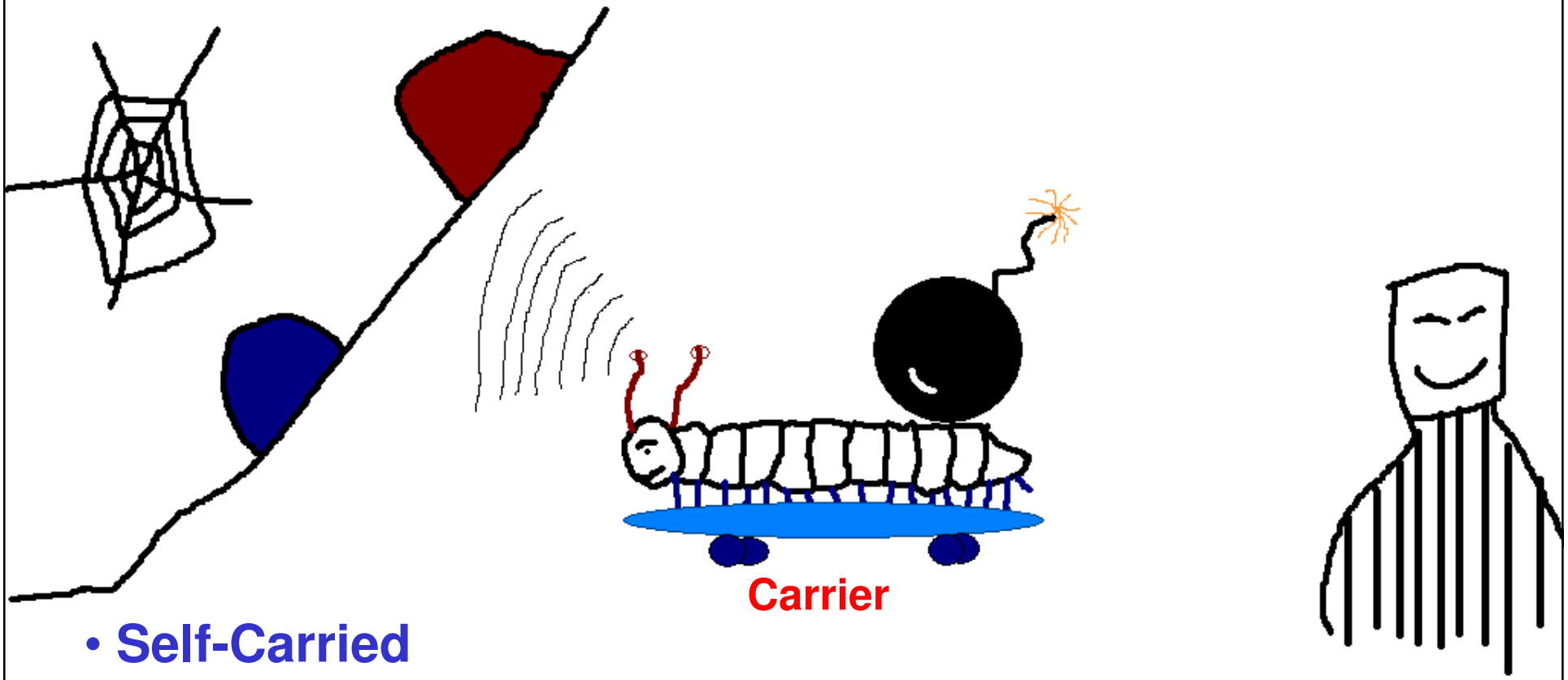
How Fast are Topological Worms?

- Depends on the topology *G = (V, E)*
  - Vulnerable machines are vertices, edges are local information
  - Time to infect is a function of the shortest paths from the initial point of infection
- Power law or similar graph (KaZaA)
  - Depends greatly on the parameters, but generally very, VERY fast



Passive Worms & Contagion Strategies

- Wait for information about other targets
  - CRclean, an anti-CodeRed II worm
    - Wait for Code Red, respond with counterattack
  - Nimda: Infect vulnerable IE versions with Trojan web-page
  - Contagion strategies (not yet seen, see "How to Own"...)
    - Piggyback infection on normal traffic
- Speed is highly variable
  - Depends on normal communication traffic
- Very high stealth
  - Have to detect the act of infection, not target selection



- **Self-Carried**

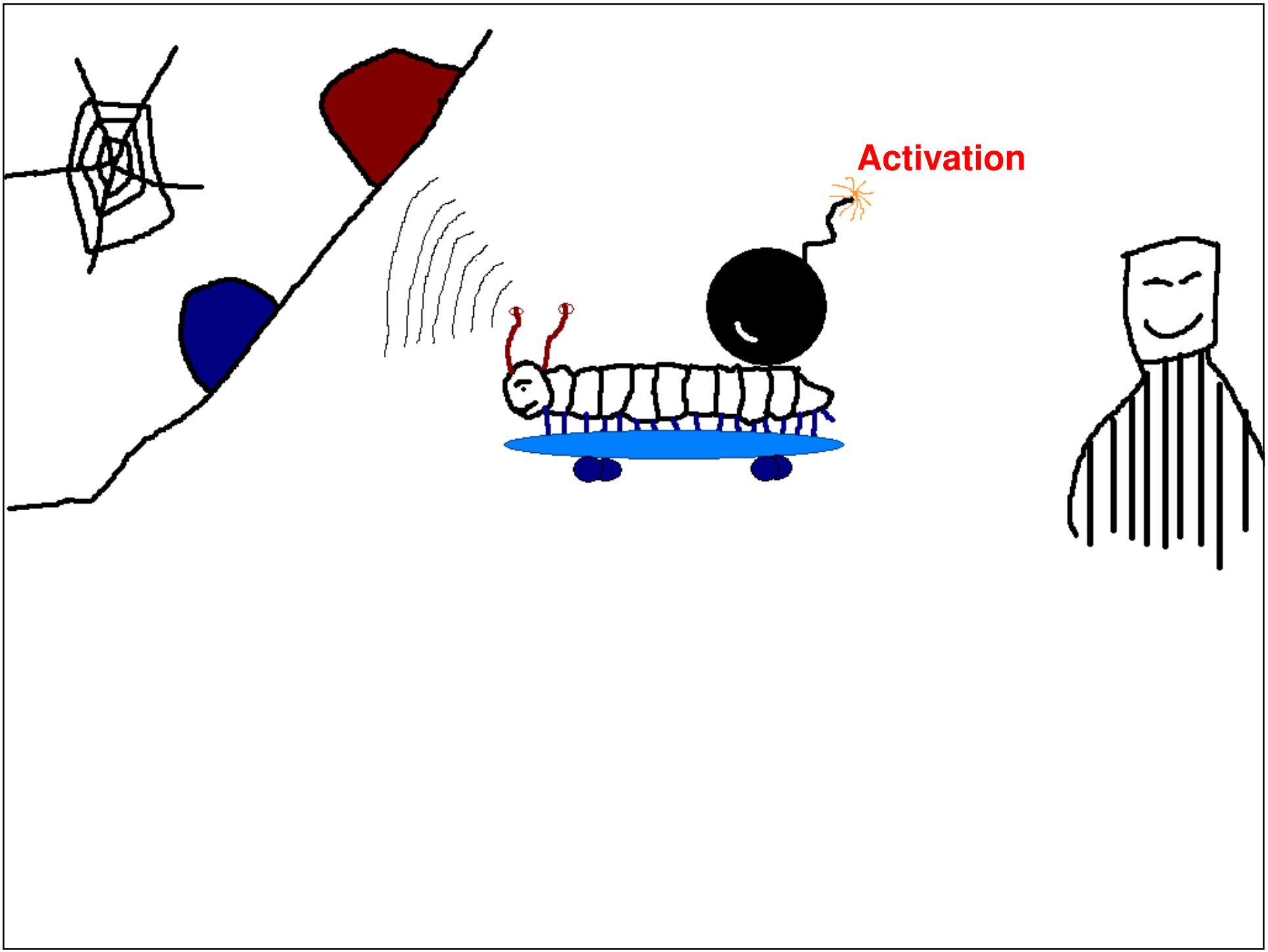
  active transmission

- **Second Channel**

  e.g. blaster worm use RPC to exploit, but use TFTP to download the whole virus body

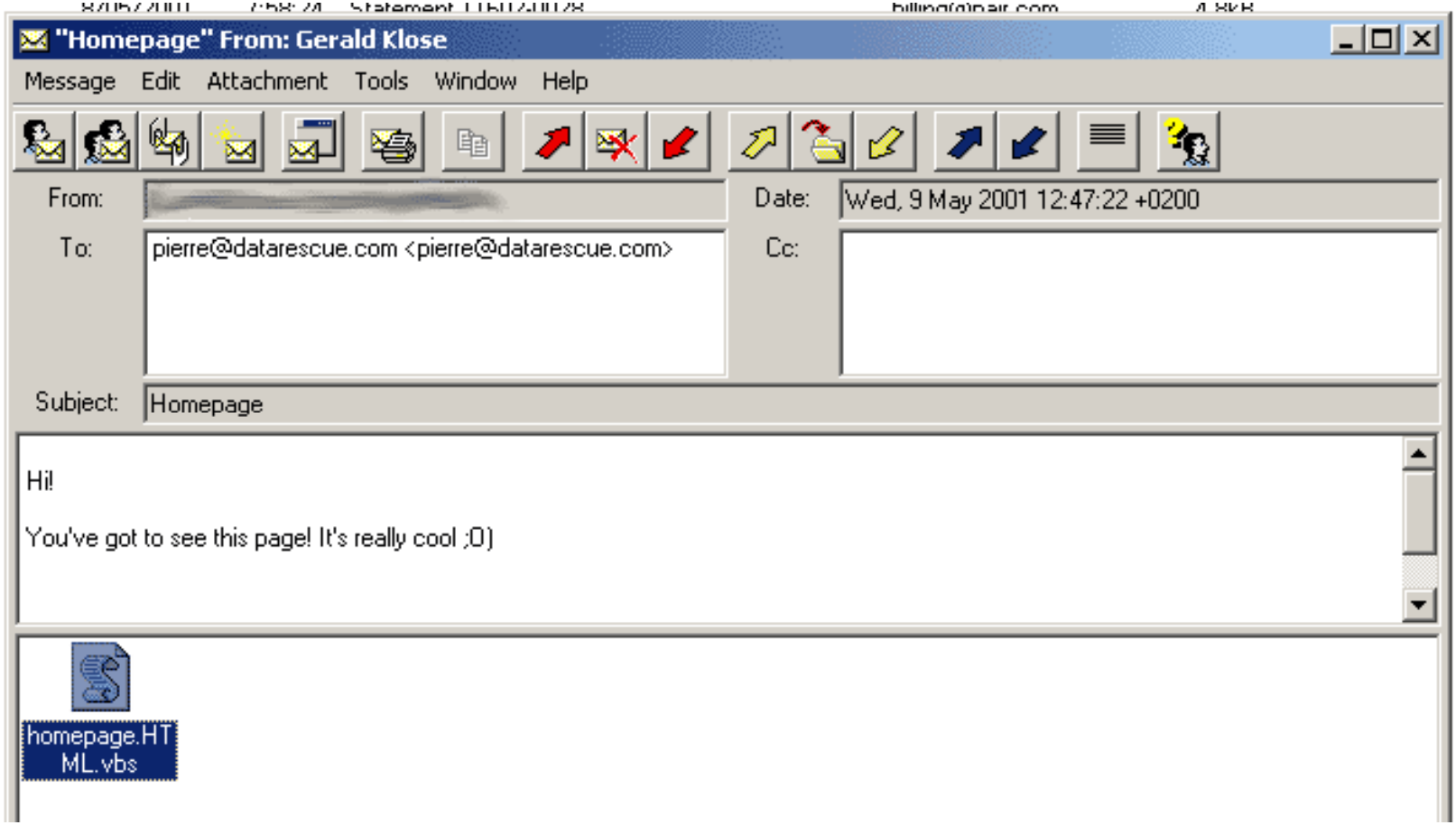
- **Embedded**

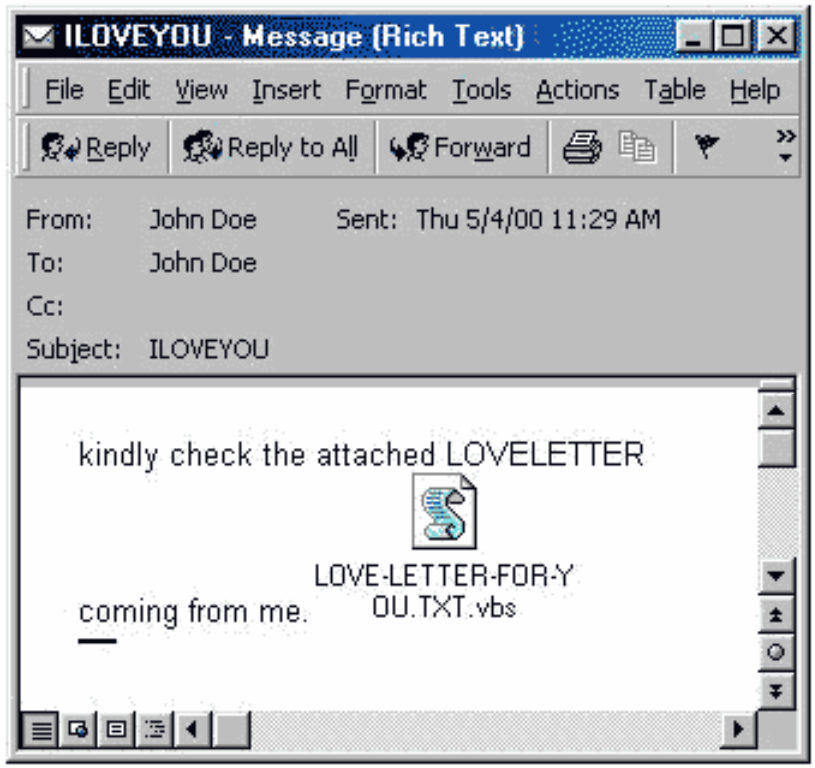
  e.g. web requests


Activation

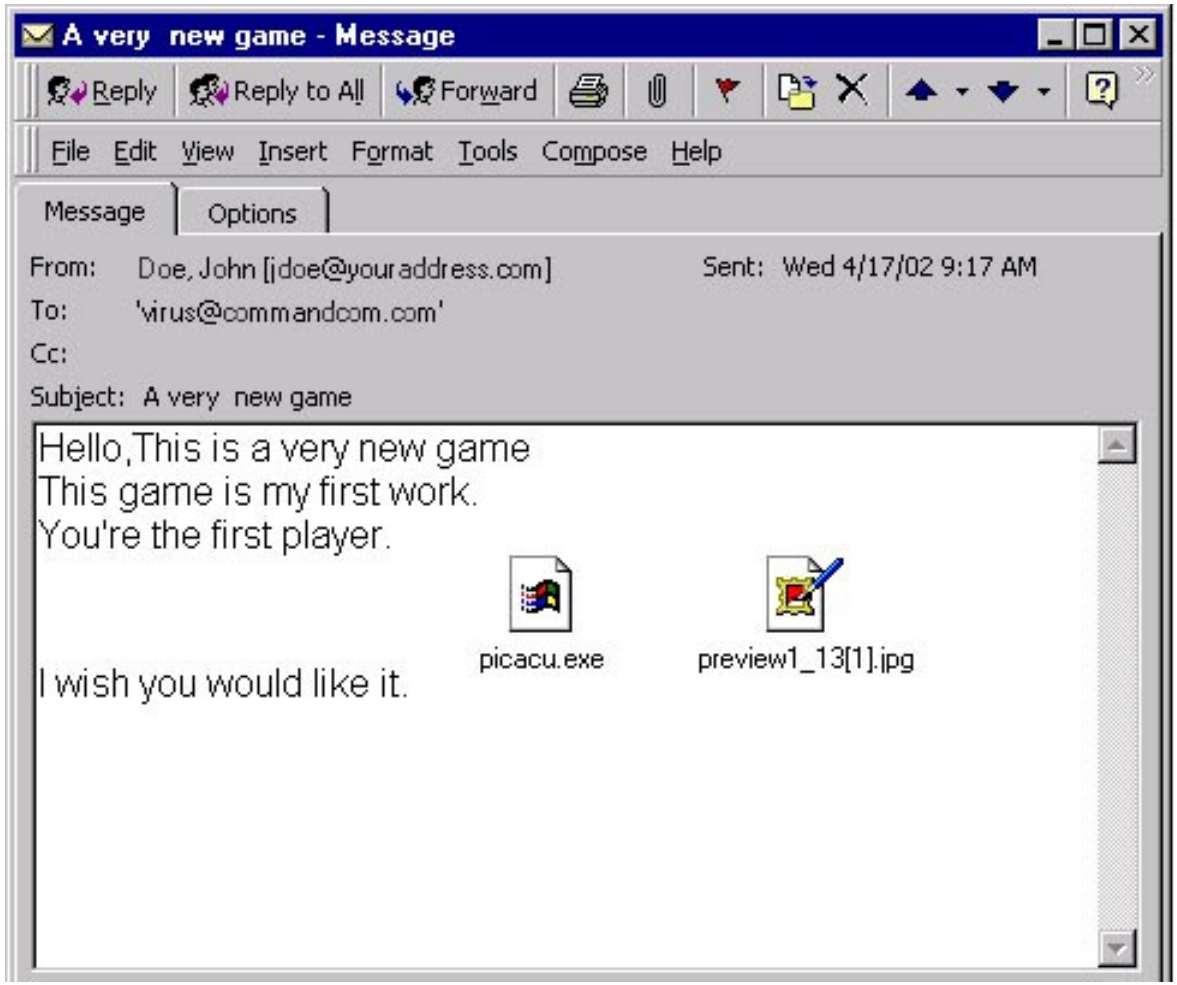
Activation

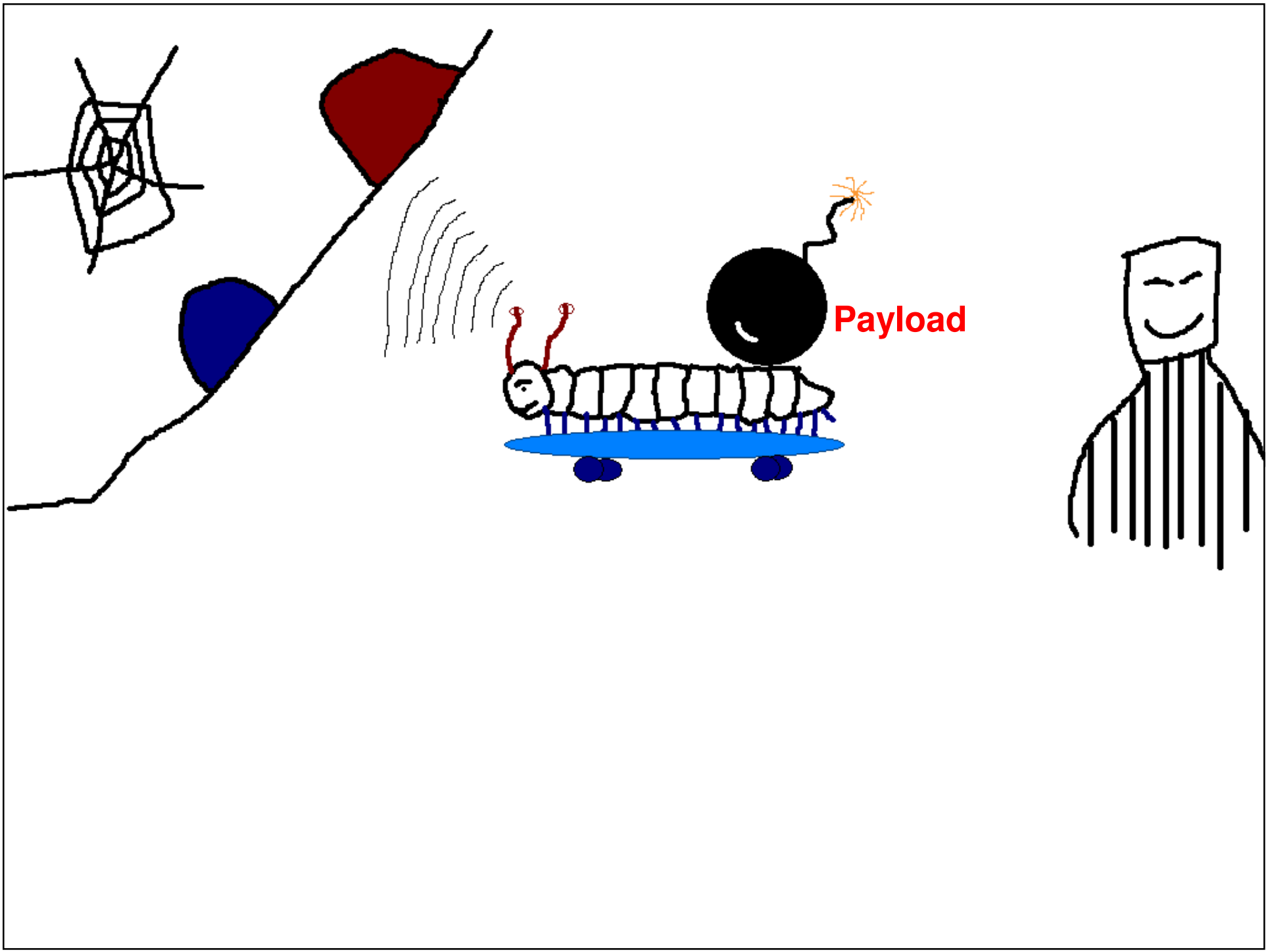
- Human Activation
  - Needs social engineering, especially for email worms
    - Melissa – "Attached is an important message for you!"
    - Iloveyou – "Open this message to see who loves you!"
- Human activity-based activation
  - E.g. logging in, rebooting (Nimda's secondary propagation)
- Scheduled process activation
  - E.g. updates, backup etc.
- Self Activation
  - E.g. Code Red exploit the IIS web servers


"Homepage" From: Gerald Klose
Message Edit Attachment Tools Window Help
From:
Date: Wed, 9 May 2001 12:47:22 +0200
To: pierre@datarescue.com <pierre@datarescue.com>
Cc:
Subject: Homepage
Hi!
You've got to see this page! It's really cool ;0)
homepage.HTML.vbs



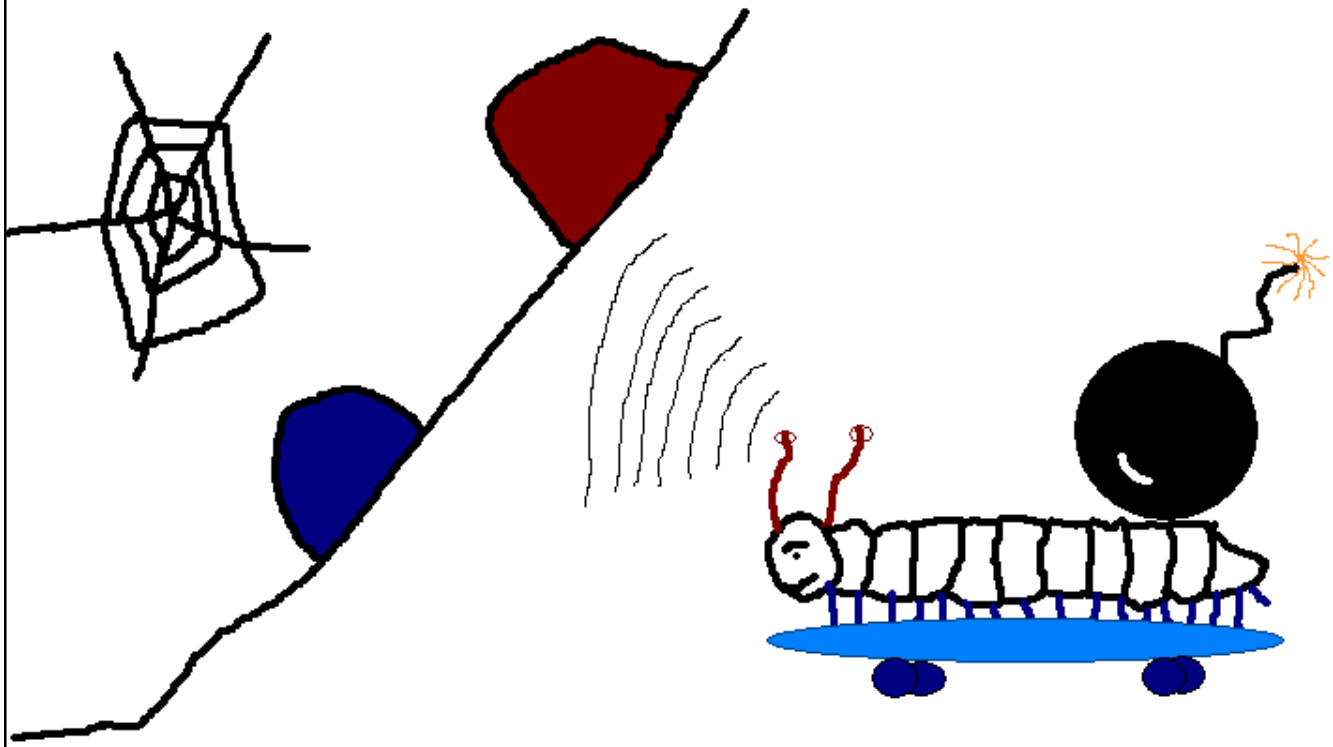
Screenshot courtesy of F-Secure.com

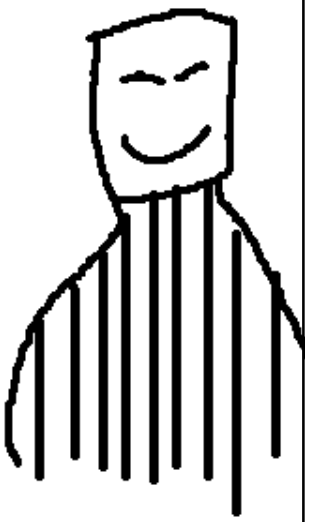

A very new game - Message
Reply
Reply to All
Forward
File Edit View Insert Format Tools Compose Help
Message
Options
From: Doe, John [jdoe@youraddress.com]
Sent: Wed 4/17/02 9:17 AM
To: 'virus@commandcom.com'
Cc:
Subject: A very new game
Hello,This is a very new game
This game is my first work.
You're the first player.
picacu.exe
preview1_13[1].jpg
I wish you would like it.


Payload

Payloads

- None/nonfunctional
  - Most common
  - Still can have significant effects through traffic and machine load (e.g., Morris worm)
- Internet Remote Control
  - Code Red II open backdoor on victim machines: anyone with a web browser can execute arbitrary code
- Internet Denial of Service (DOS)
  - E.g., Code Red, Yaha
- Data Collection
- Data Damage: Chernobyl , Klez
- Worm maintenance



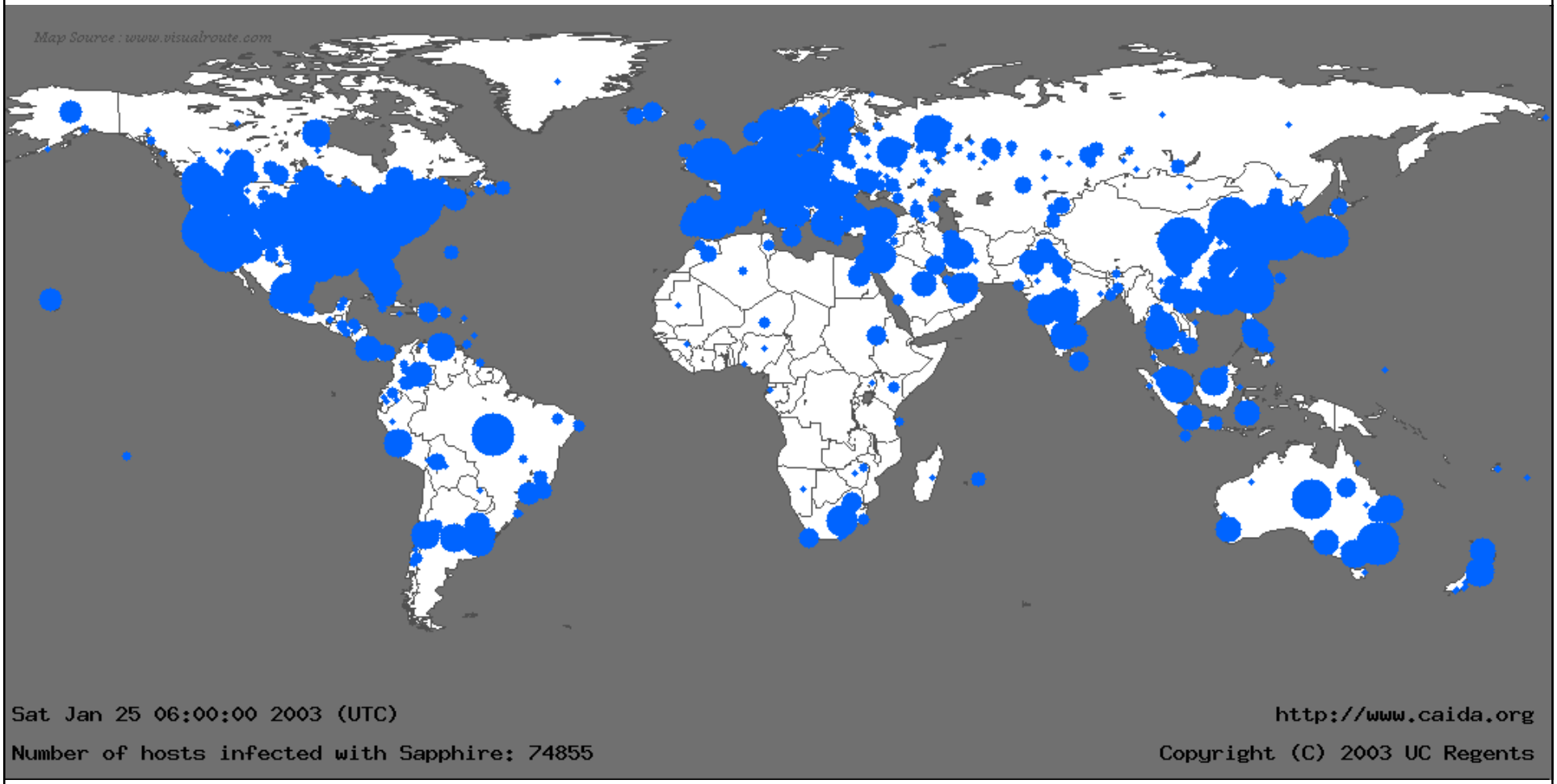


- Experimental Curiosity
- Pride and Power
- Commercial Advantage
- Extortion and criminal gain
- Terrorism
- Cyber Warfare

Some Major Worms

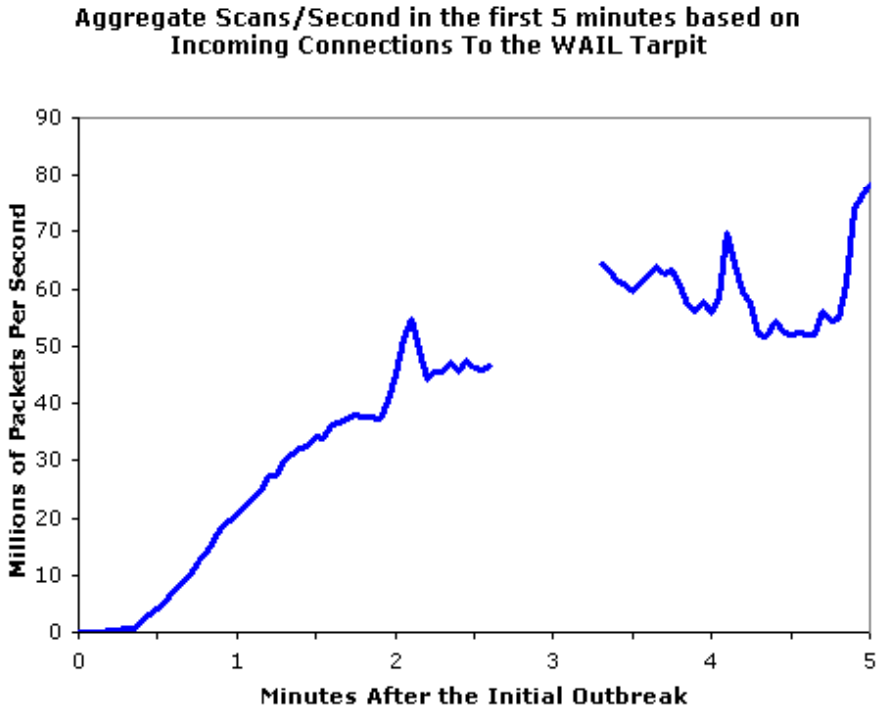
Worm	Year	Strategy	Victims	Other Notes
Morris	1988	Topological	6000	First major autonomous worm. Attacked multiple vulnerabilities.
Code Red	2001	Scanning	~300,000	First recent "fast" worm, 2nd wave infected 360,000 servers in 14 hours
CRClean	2001	Passive	none	Unreleased Anti-Code-Red worm.
Nimda	2001	Scanning IIS, Code Red 2 backdoor, etc	~200,000	Local subnet scanning. Effective mix of techniques
Scalper	2002	Scanning	<10,000	Released 10 days after vulnerability revealed
Slammer	2003	Scanning	>75,000	Spread worldwide in 10 minutes

The Spread of the Sapphire/Slammer SQL Worm



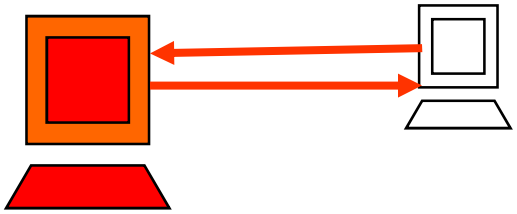
How Fast was Slammer?

- Infected ~75,000 machines in 10 minutes
- Full scanning rate in ~3 minutes
  - >55 Million IPs/s
- Initial doubling rate was about every 8.5 seconds
  - Local saturations occur in <1 minute

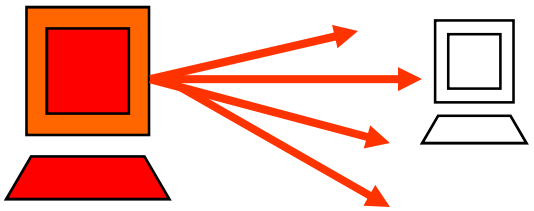


Why Was Sapphire Fast: A Bandwidth-Limited Scanner

- Code Red's scanner is latency-limited
  - In many threads: send SYN to random address, wait for response or timeout
  - Code Red → ~6 scans/second,
    - population doubles about every 40 minutes



- Every Sapphire copy sent infectious packets at maximum rate
  - 1 Mb upload bandwidth → 280 scans/second
  - 100 Mb upload bandwidth → 28,000 scans/second



- Any reasonably small TCP worm can spread like Sapphire
  - Needs to construct SYNs at line rate, receive ACKs in a separate thread

Backup Slides

Fred Cohen's Work: 1983

- First documented work with viruses
  - Cohen's PhD advisor, Leo Adelman, coined the term "virus"
  - Virus: "a program that can infect other programs by modifying them to include a … version of itself"
  - Viruses can quickly (~30 min) spread through a networked file system
- Dissertation (1986) conclusion: "universal" detection of a virus is undecidable
  - No 100% guaranteed detection for virus/worm

Early Mail Virus: Happy99 (1999)

- One of the earliest viruses that propagated automatically when an infected attachment is executed
- Did not infect files, only email user accounts
- Email sent from infected person to others in address book (novelty at the time)

Morris Worm

- best known classic worm
- released by Robert Morris in 1988
- targeted Unix systems
- using several propagation techniques
  - simple password cracking of local pw file
  - exploit bug in finger daemon
  - exploit debug trapdoor in sendmail daemon
- if any attack succeeds then replicated self