Computer worm

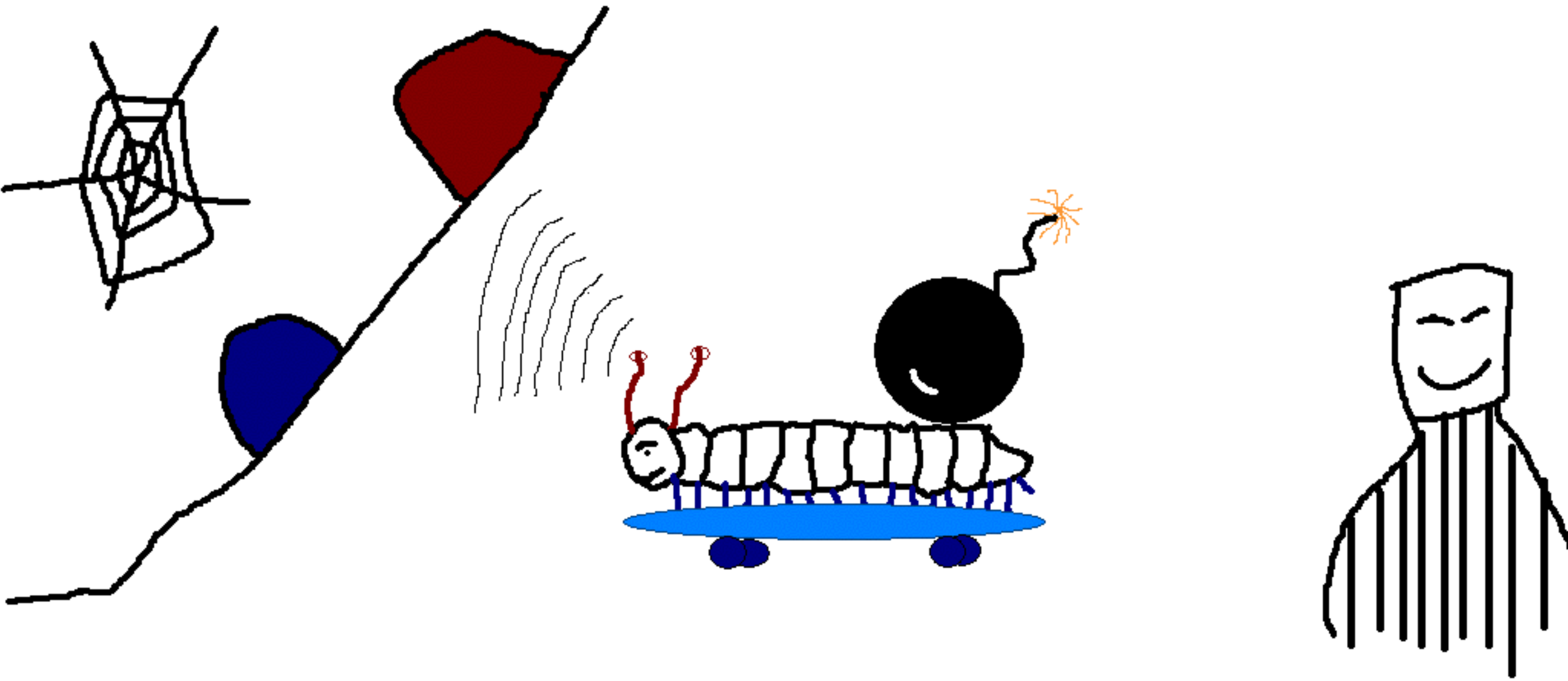
Introduction

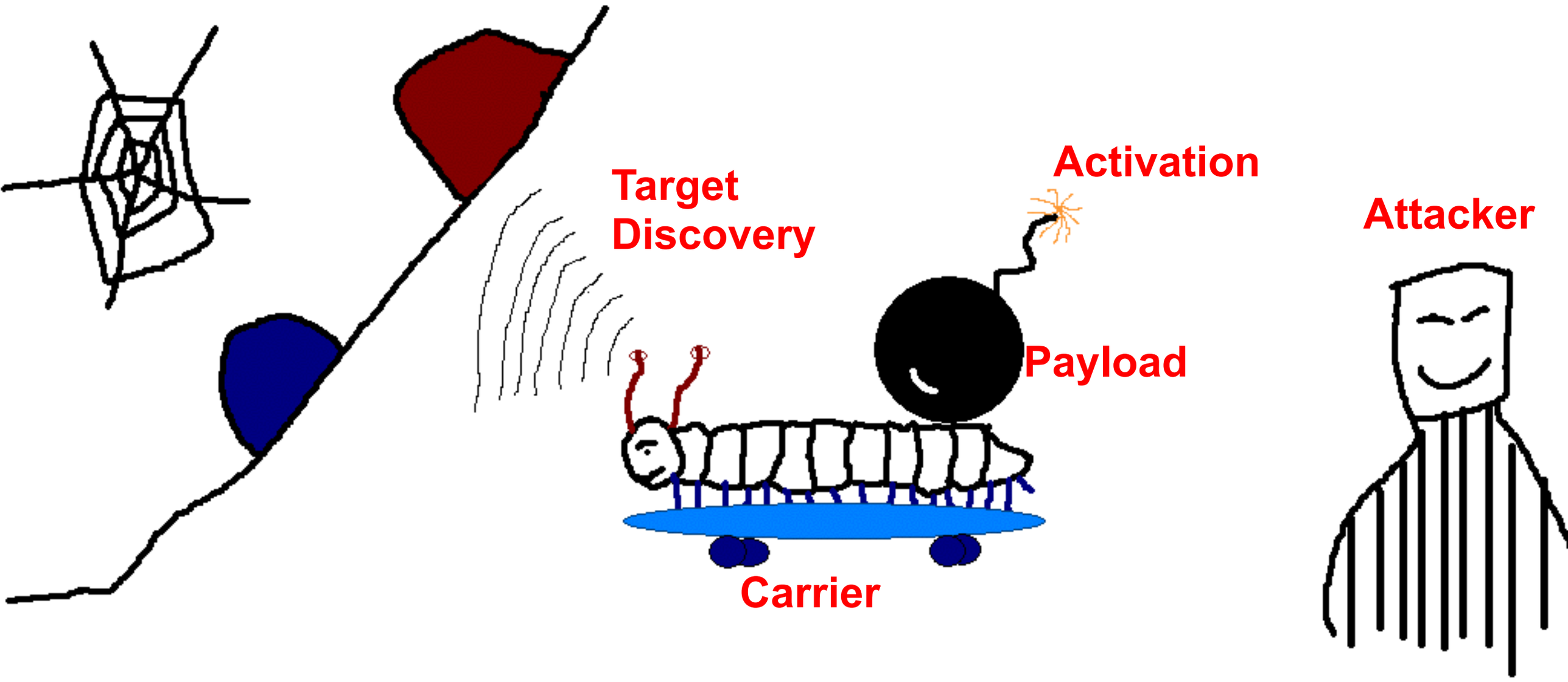
- What is a worm?
  - Piece of software that propagates using vulnerabilities in software/application
  - *Self-propagating (distinct from a virus)*
  - *Self-replicating*
  - *Spread through the Internet easily due to its open communication model*

Classification of Worms

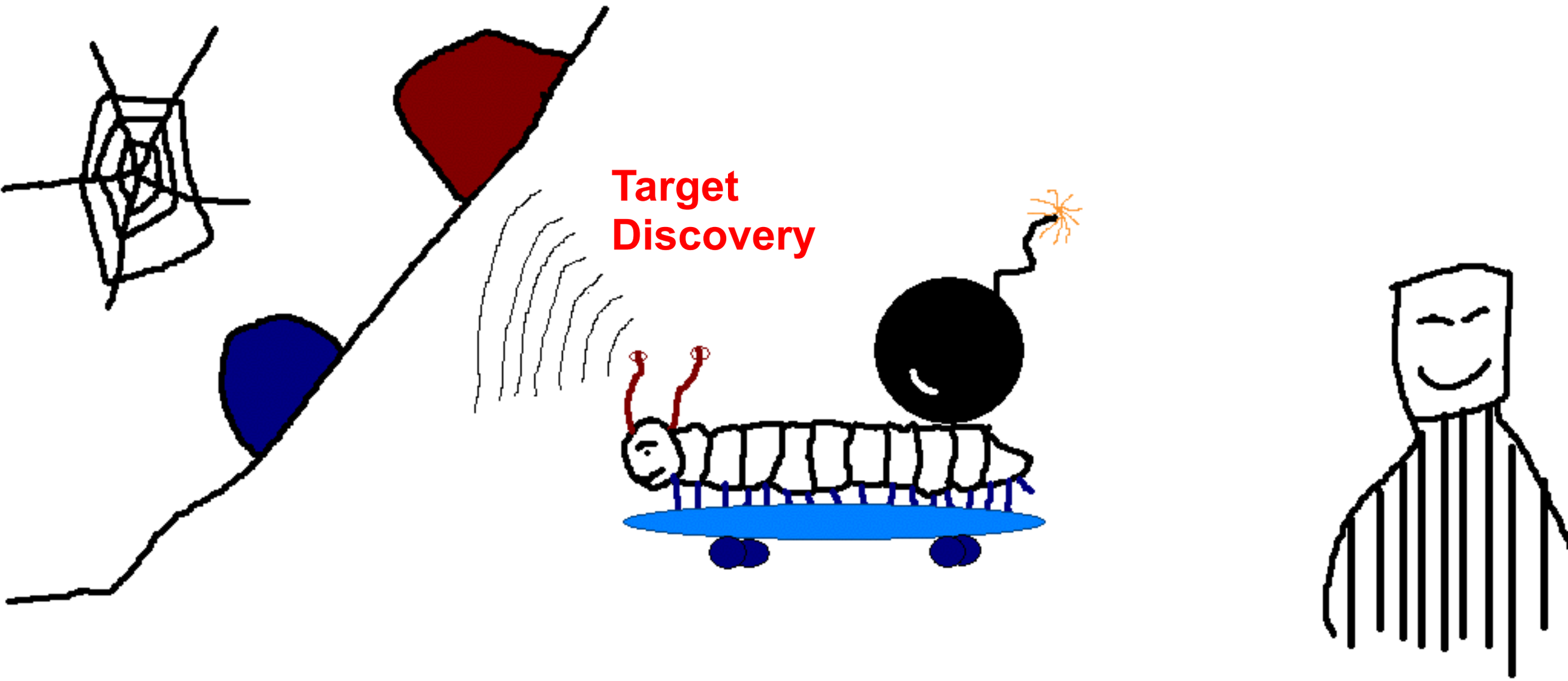
- Target Discovery
  - How does a worm find new hosts to infect?
- Carrier
  - How does it transmit itself to the target?
- Activation
  - Mechanism by which the worm operates on the target
- Payloads
  - What the worm carries to reach its goal

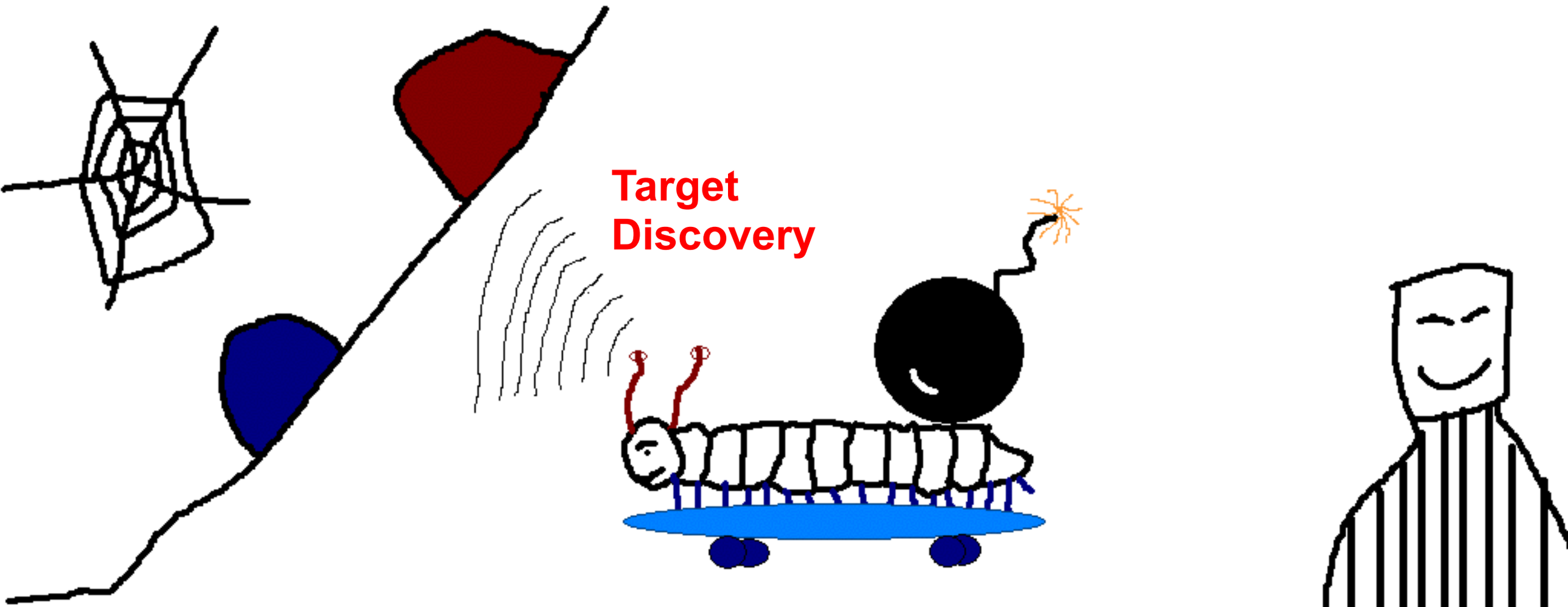
N.Weaver, V.Paxson, et al, “A taxonomy of computer worms”, Proc. Of the ACM workshop on Rapid Malcode, pp.11-18, 2003.





OVERVIEW


Target
Discovery



- **Scanning**
  - sequential, random,
  - Optimization
    - Preference for local addresses: Same OS and applications in a sub-network
    - Permutation scanning: Utilize distributed coordination to more effectively scan
    - Bandwidth-limited scanning: Do not wait for response

Target Discovery

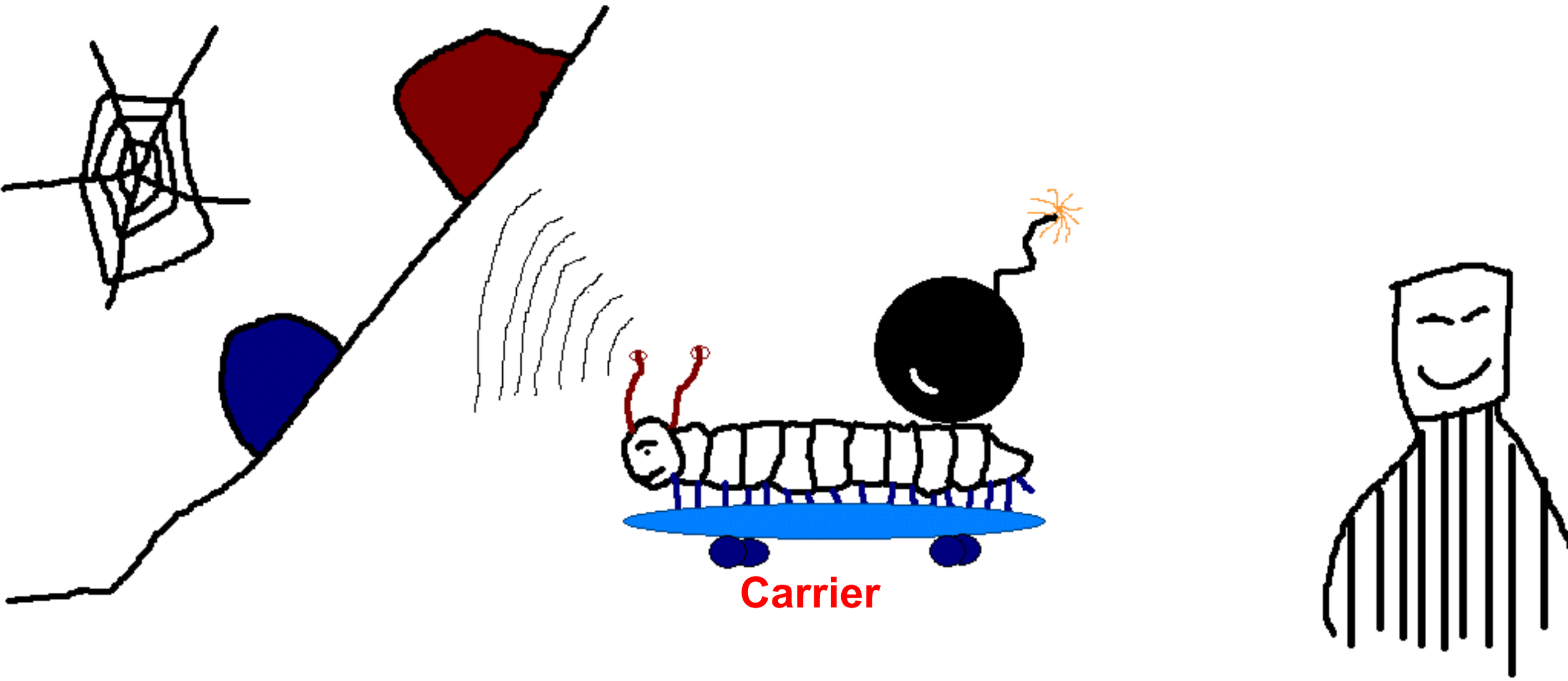
- Pre-generated Target Lists
  - Attacker made a target list in advance

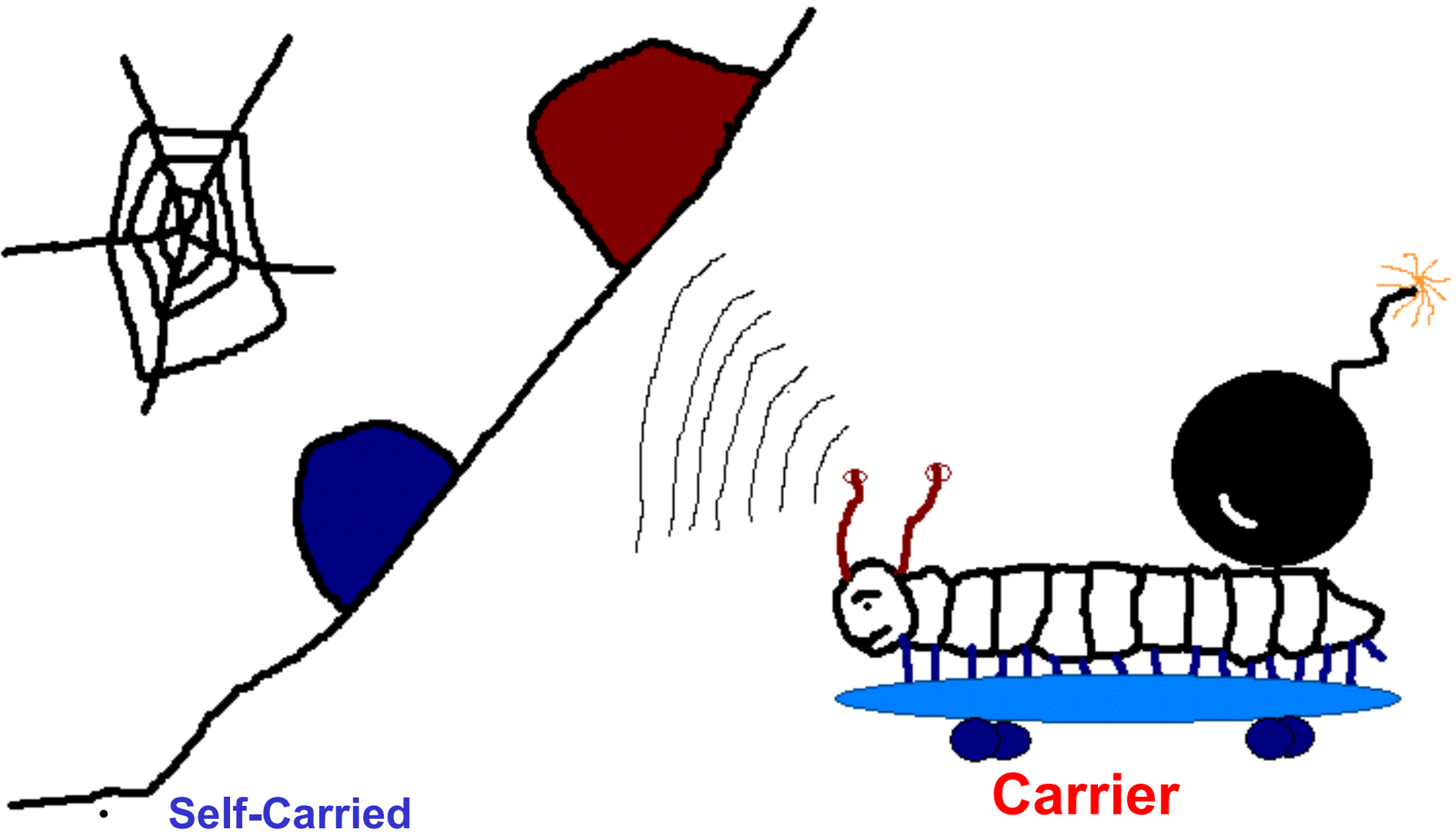
- Internal Target Lists
  - Discover the local communication topology
  - Difficult to detect
    - Suggests highly distributed sensors

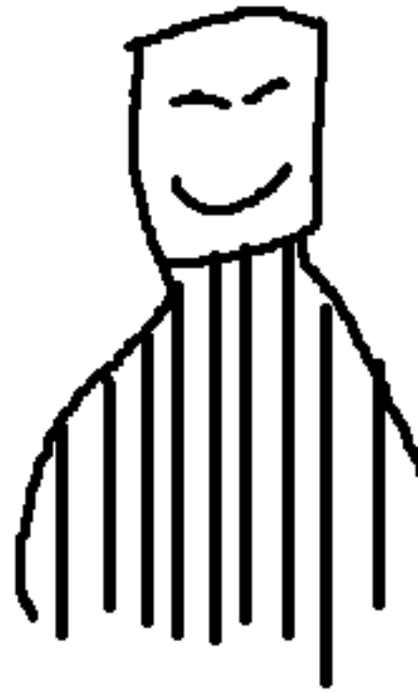
- Externally Generated Target Lists
  - Metaservers keep a list of all the servers which are currently active (Ex. Online game)

Target Discovery

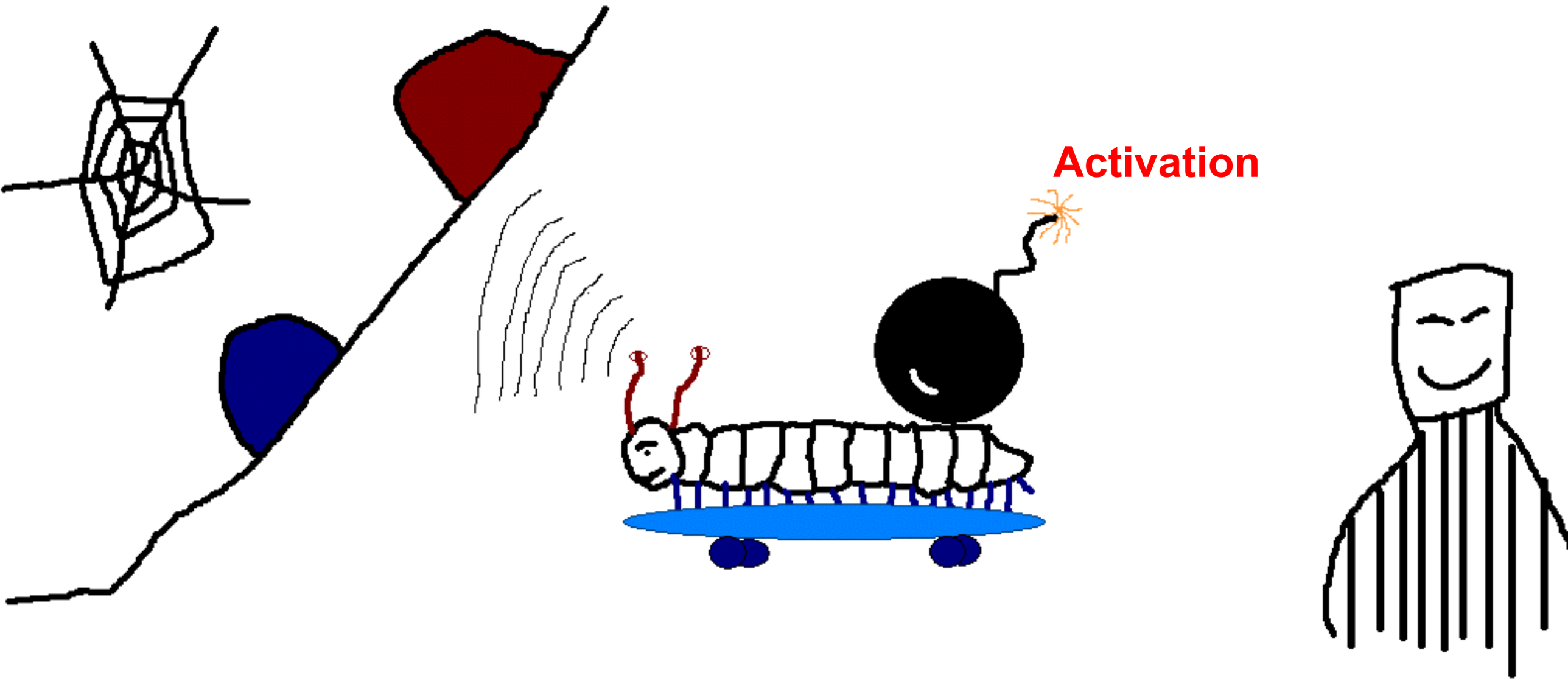
- Passive
  - Wait for potential victims to contact the worm (Ex. Un-patched browser)
  - Rely on user behavior to discover new targets
    - **Contagion** worms rely on normal communication to discover new victims
  - No anomalous traffic patterns during target discovery

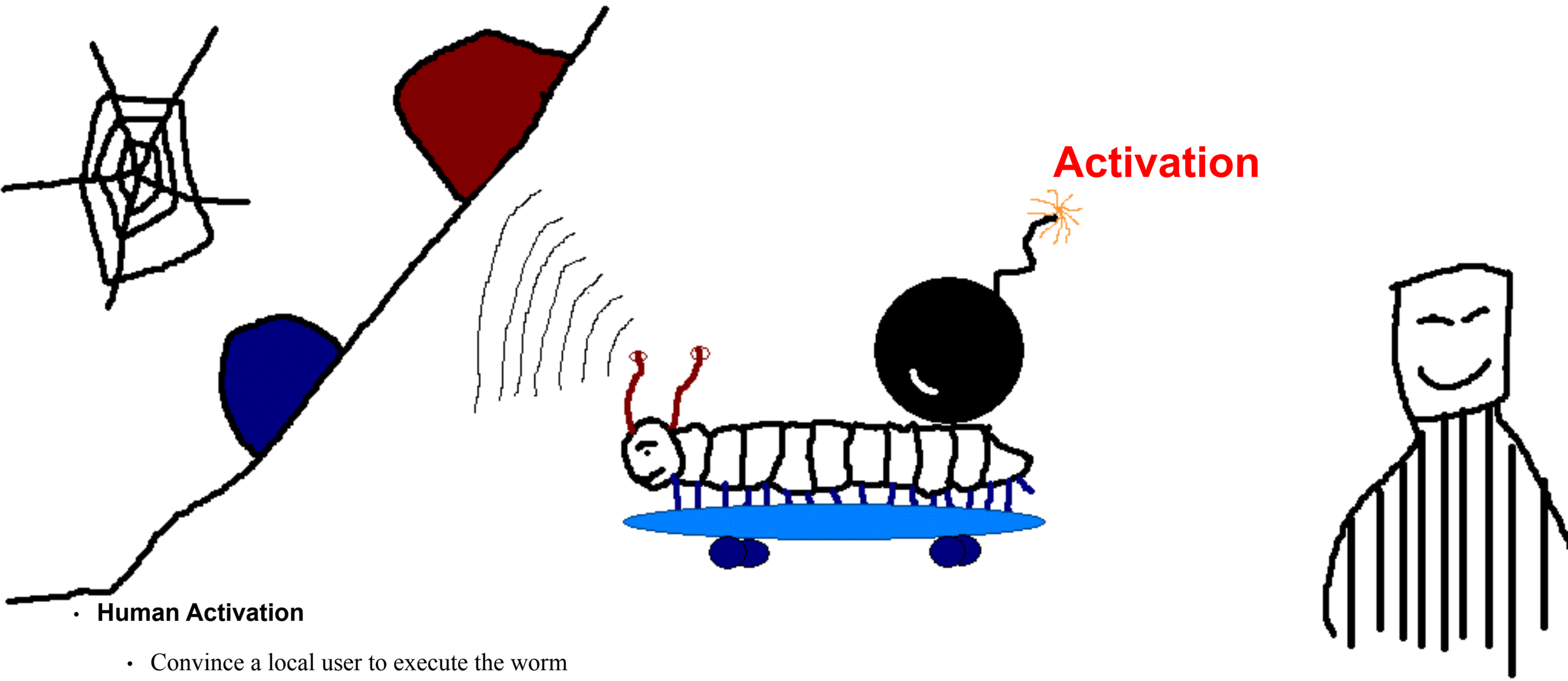

Carrier





- **Self-Carried**
  - Actively transmits itself as part of the infection process
- **Second Channel**
  - Require a secondary communication channel
  - Example Blaster: primary channel is RPC;
  - secondary channel is TFTP
- **Embedded**
  - **Sends itself as part of a normal communication channel, either appending to or replacing normal messages (e.g. web requests)**
  - **Usually used by passive worms**
  - **Relatively stealthy**


Activation



- **Human Activation**
  - Convince a local user to execute the worm
  - The slowest activation approach
    - e.g. MyDoom
- **Human activity-based activation**
  e.g. Activated when the user performs some activity not normally related to a worm (Ex. resetting the machine, logging in)
- **Scheduled process activation**
  - Unauthorized auto-updater programs
  - Ex. Use DNS redirection attack to serve a file to the desktop system to infect the target
- **Self Activation**
  - Initiate their own execution by exploiting vulnerabilities in services that are always on and available
  - The fastest activation approach

"Homepage" From: Gerald Klose

Message Edit Attachment Tools Window Help

From: [redacted] Date: Wed, 9 May 2001 12:47:22 +0200

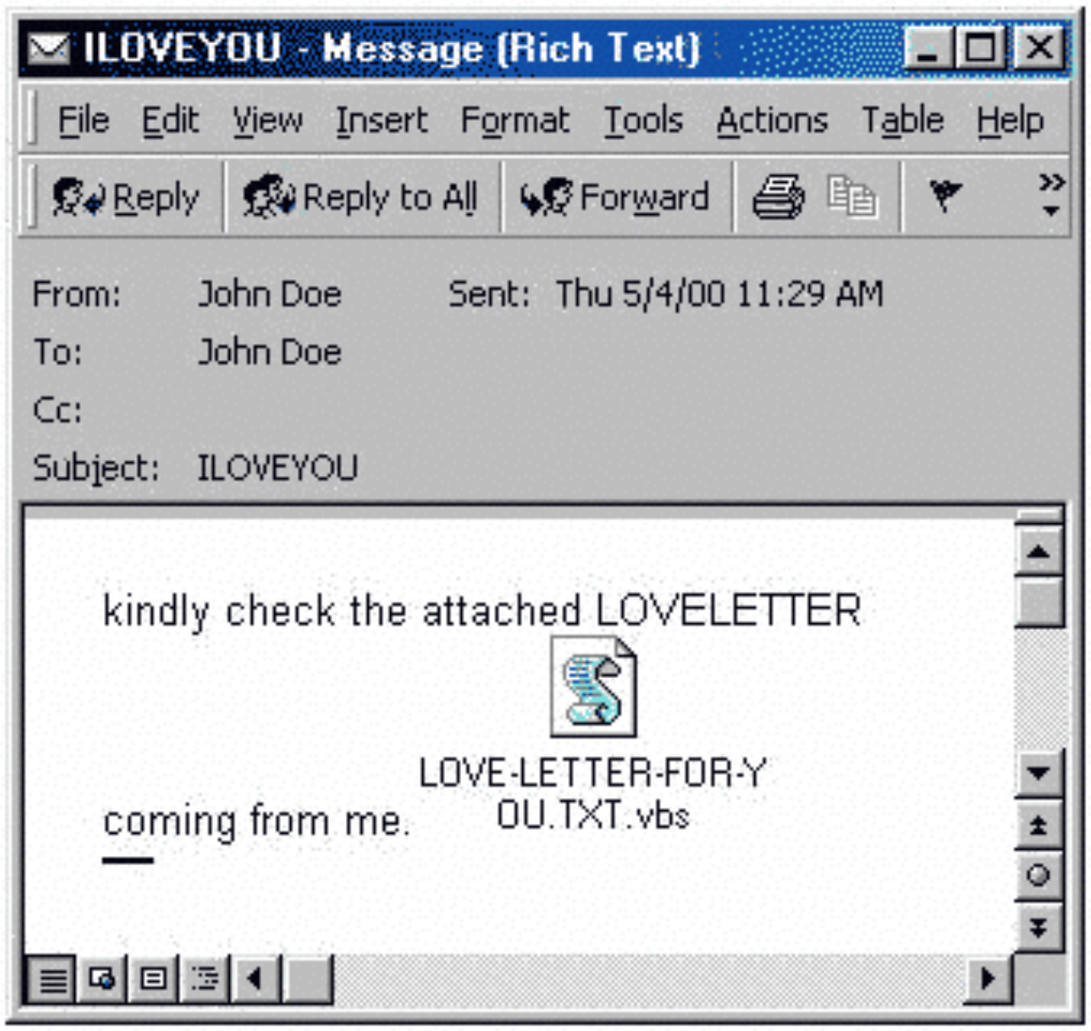
To: pierre@datarescue.com <pierre@datarescue.com> Cc:

Subject: Homepage

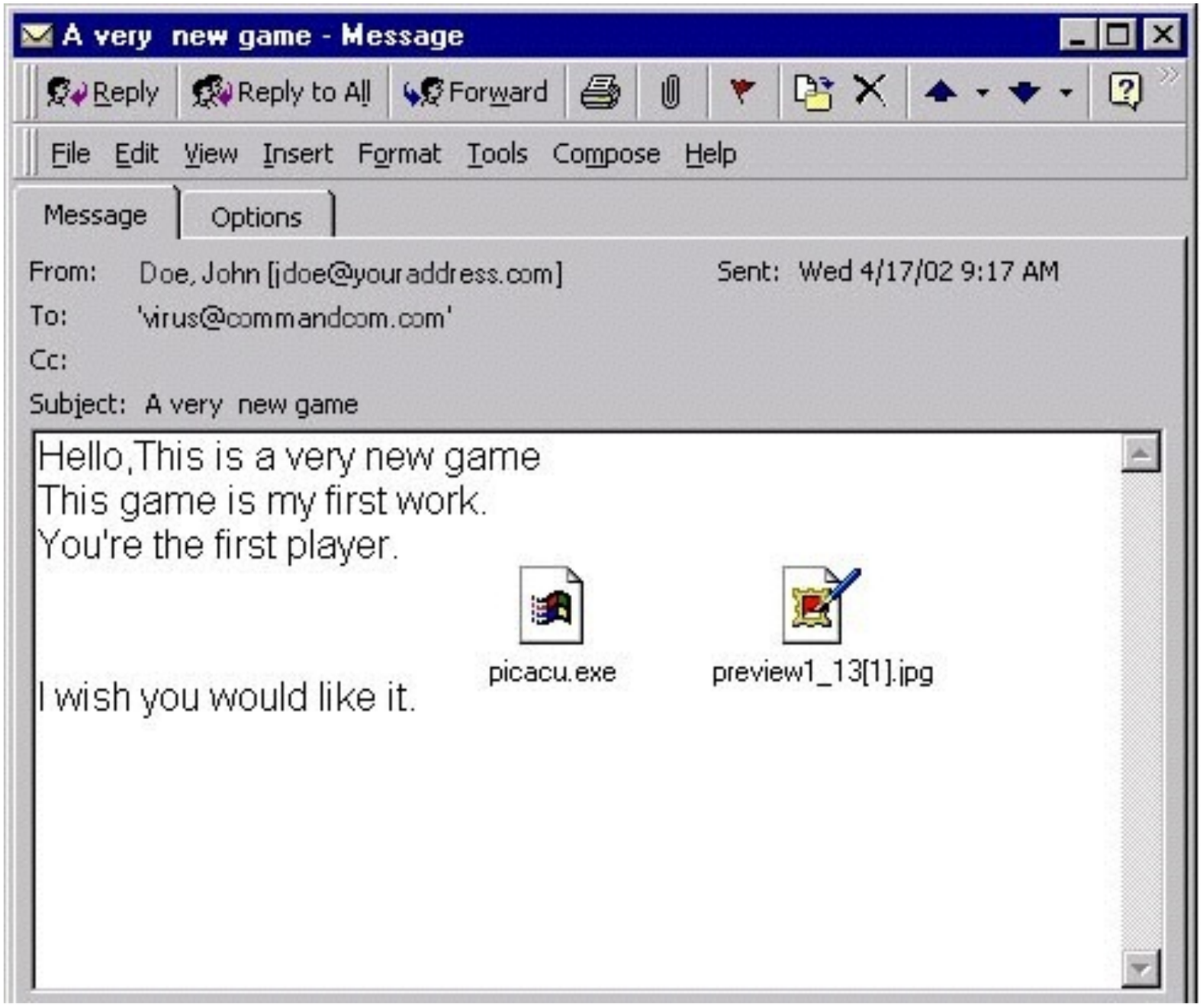
Hi!

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.

Microsoft

All Products | Support | Search | Microsoft.com Guide

Microsoft Home

MS User

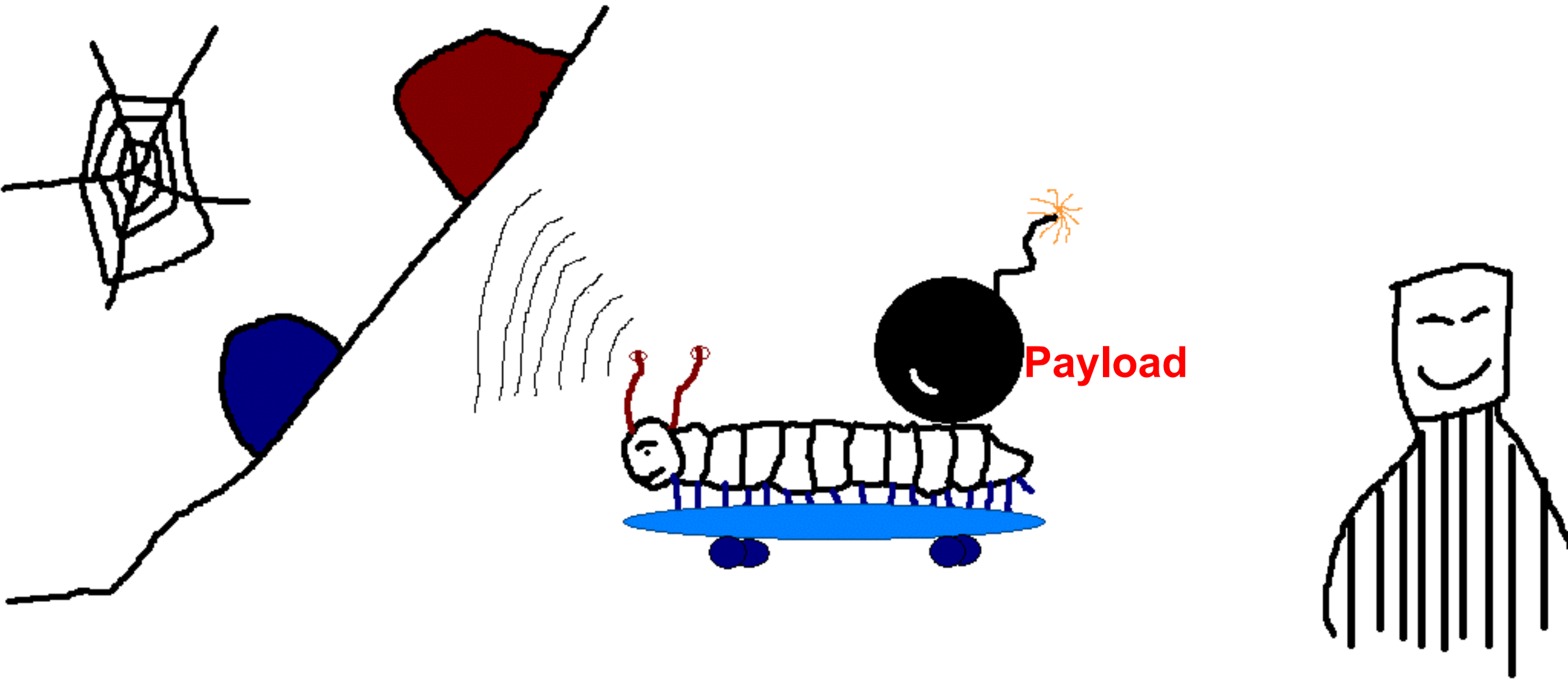
this is the latest version of security update, the "September 2003, Cumulative Patch" update which eliminates all known security vulnerabilities affecting MS Internet Explorer, MS Outlook and MS Outlook Express. Install now to maintain the security of your computer from these vulnerabilities. This upcate includes the functionality of all previously released patches.

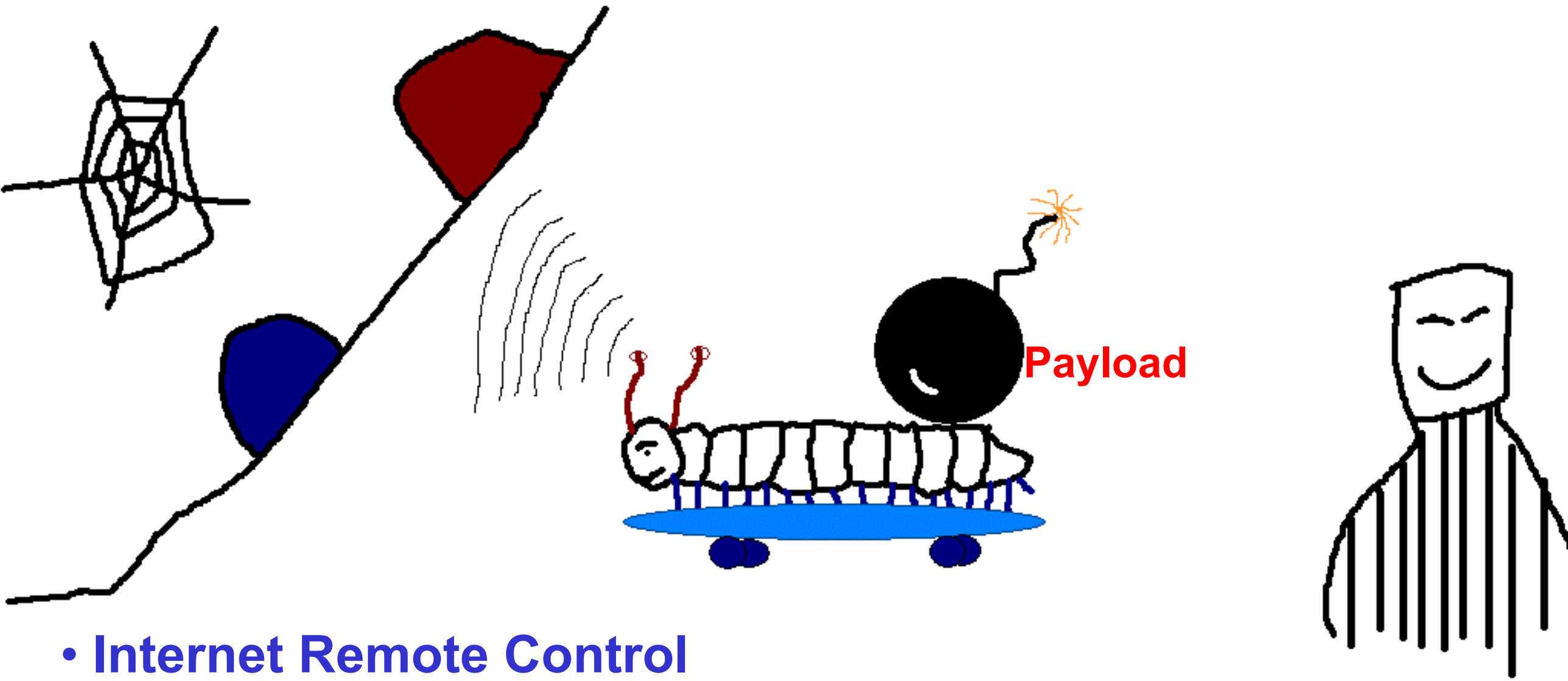
System requirements	Windows 95/98/Me/2000/NT/XP
This update applies to	MS Internet Explorer, version 4.01 and later MS Outlook, version 8.00 and later MS Outlook Express, version 4.01 and later
Recommendation	Customers should install the patch at the earliest opportunity.
How to install	Run attached file. Choose Yes on displayed dialog box.
How to use	You don't need to do anything after installing this item.

Microsoft Product Support Services and Knowledge Base articles can be found on the Microsoft Technical Support web site. For security-related information about Microsoft products, please visit the Microsoft Security Advisor web site, or Contact Us.

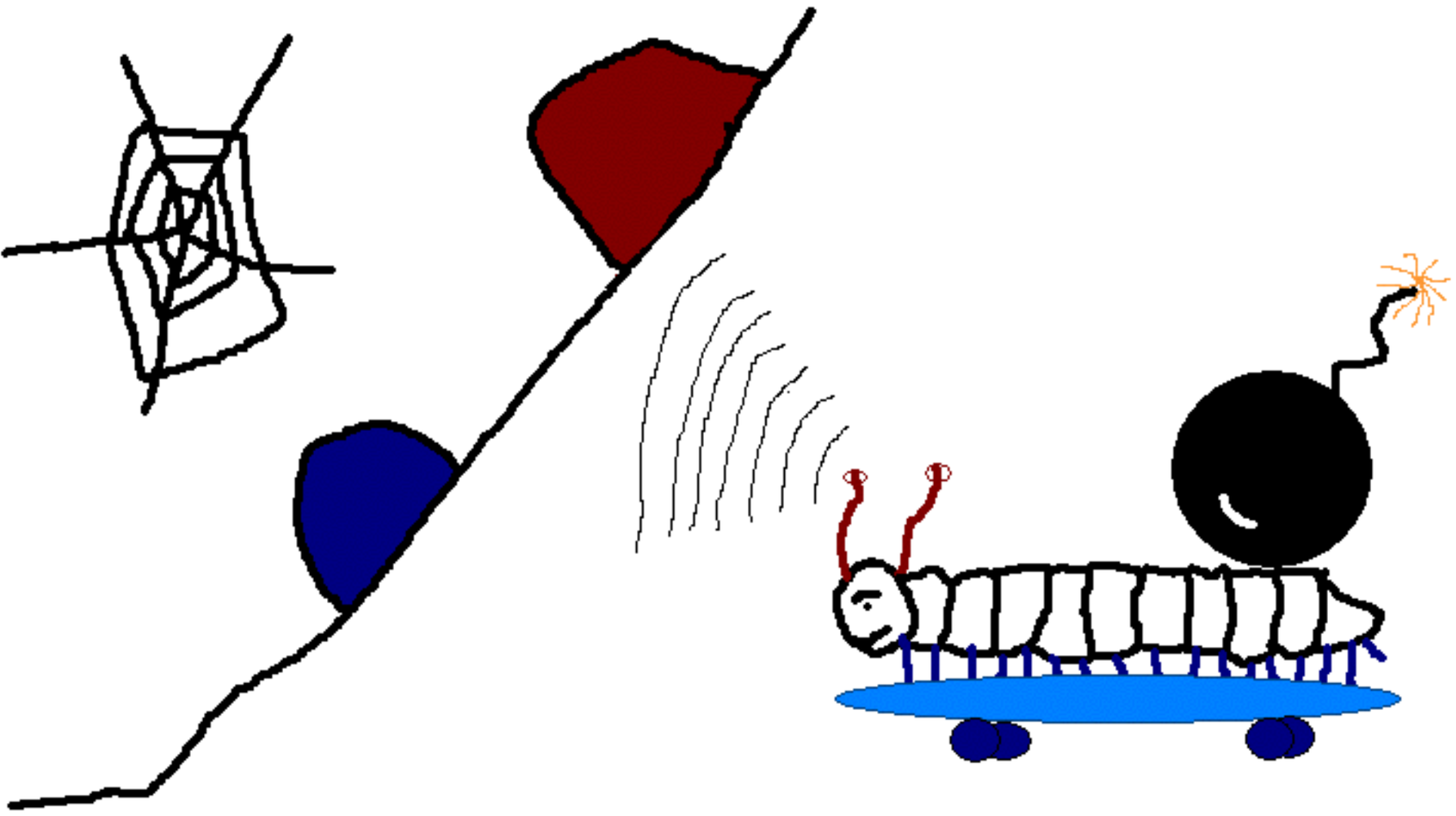
Thank you for using Microsoft products.

Please do not reply to this message. It was sent from an unmonitored e-mail address and we are unable to respond to any replies.

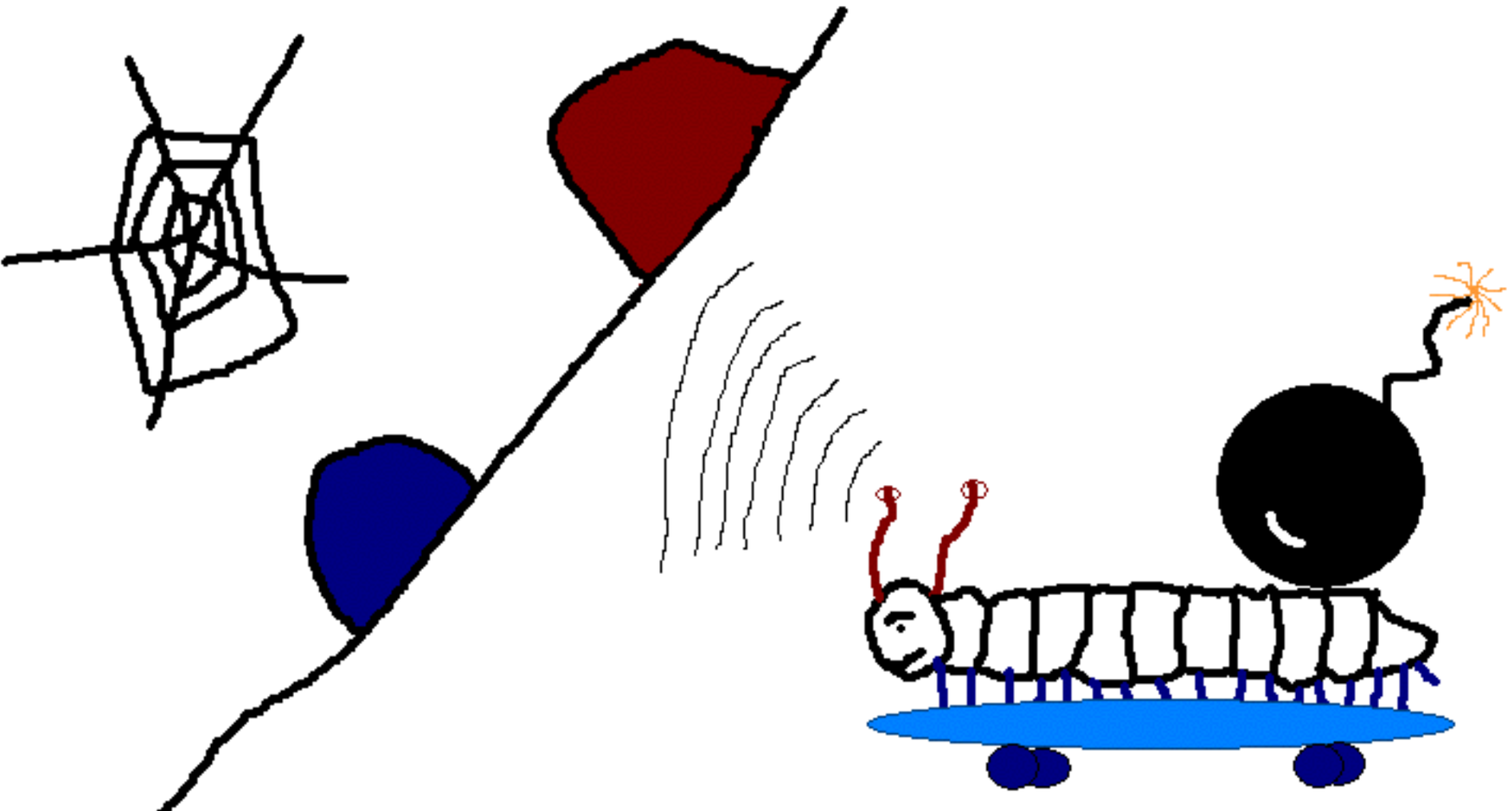

Payload

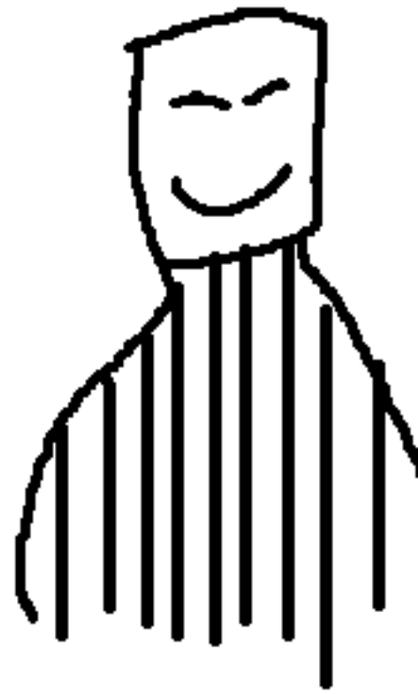


- **Internet Remote Control**
- **Internet DOS : paper's dream realized**
- **Data Damage:** Chernobyl , Klez
- **Physical World Damage**
- **Human control → Blackmail !**



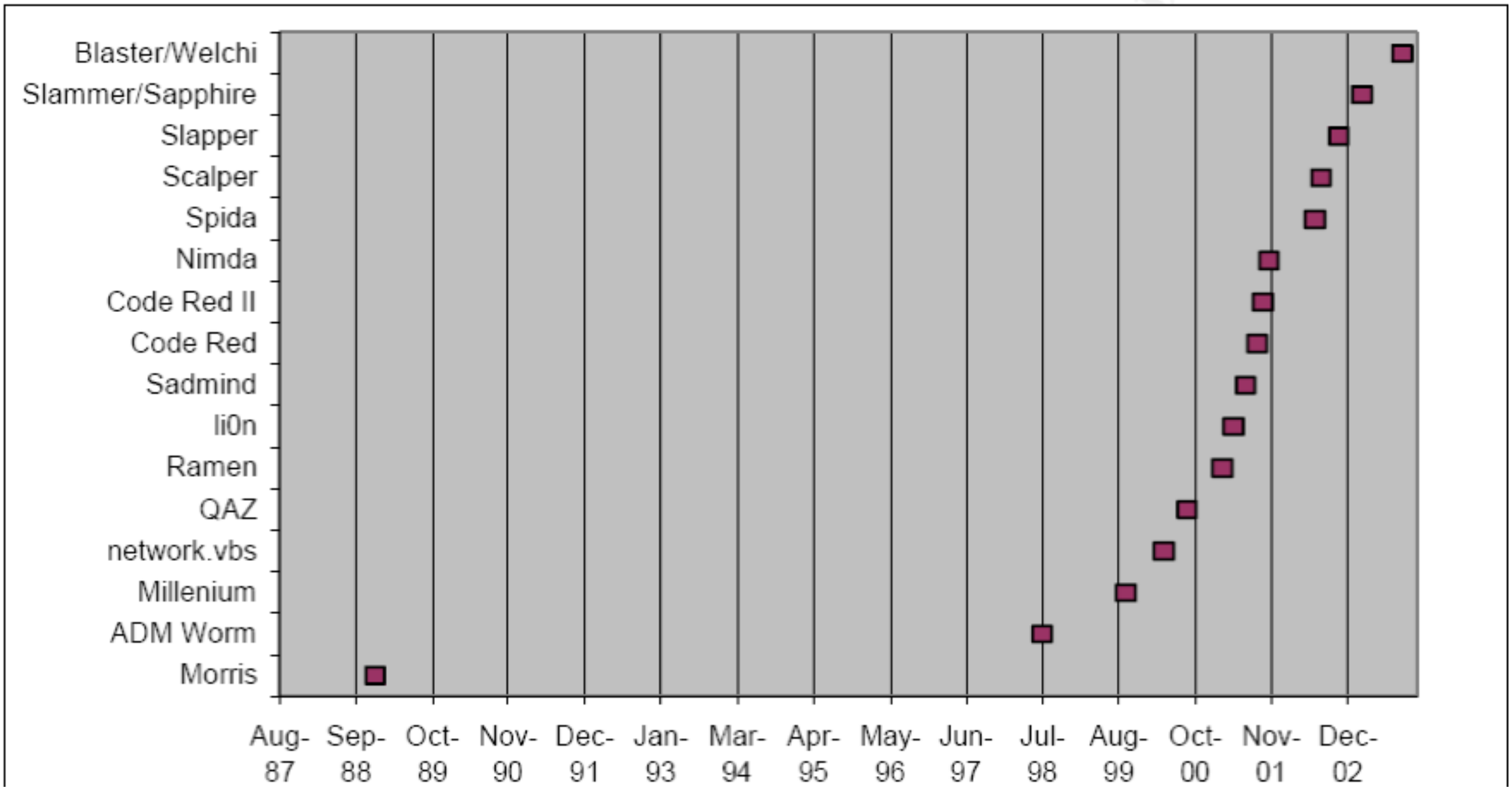

Attacker





- Experimental Curiosity
  - Continual tendency for various individuals to experiment with viruses and worms
- Pride and Power
  - A desire to acquire power, to show off their knowledge and ability to inflict harm on others
- Commercial Advantage
  - Profit by manipulating financial markets via a synthetic economic disaster
- Extortion and Criminal Gain
  - Credit-card information
- Random Protest
  - Disrupt networks and infrastructure
- Political Protest
- **Terrorism → Example**
- **Cyber Warfare**

History of Worms



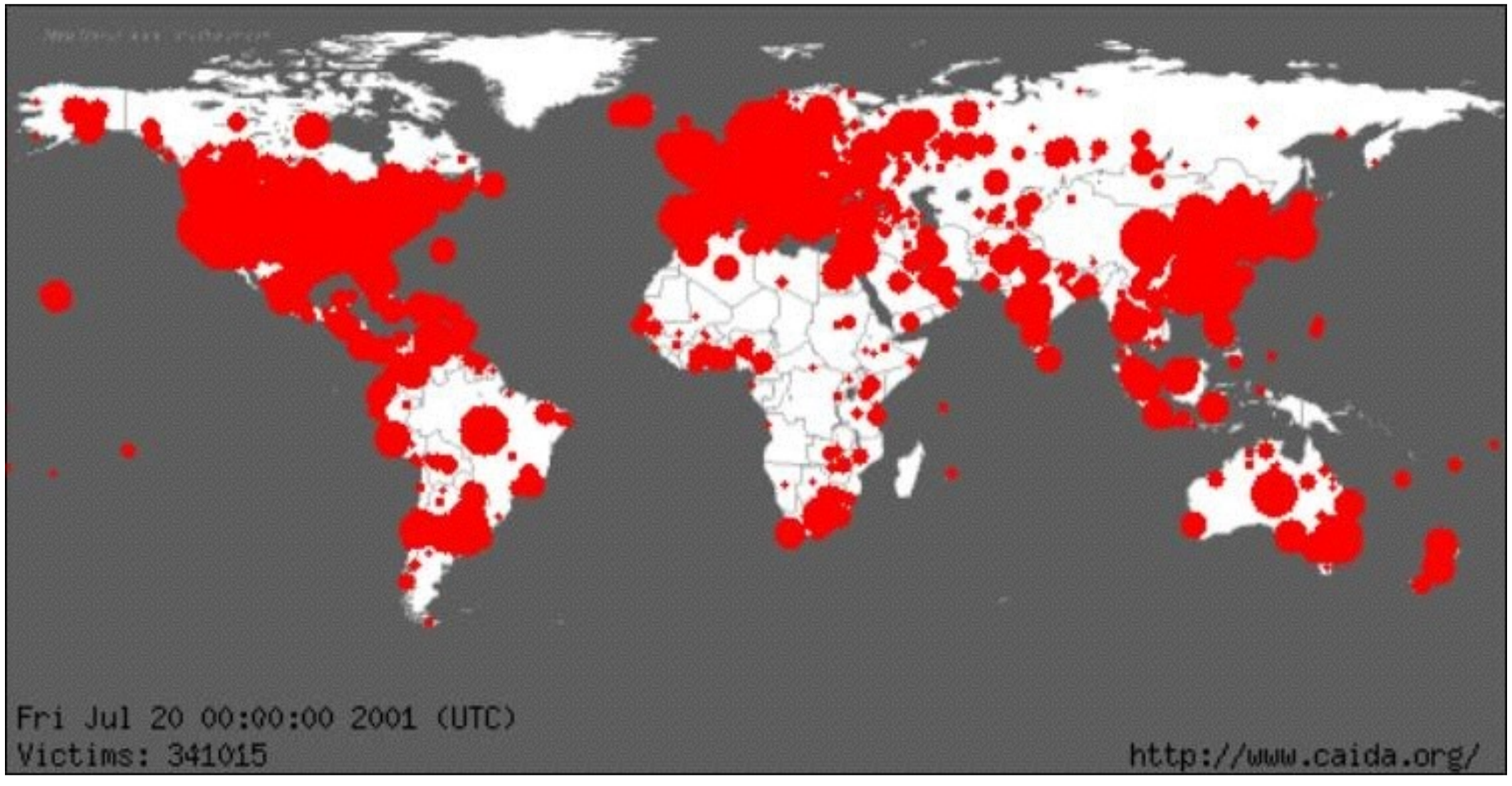
Source:http://www.sans.org/rr/whitepapers/malicious/1410.php

Morris Worm

- Topological Worm (*6-10%* of all Internet hosts infected)
- First large-scale worm that targeted VAX, Sun Unix systems
- Target Discovery
  - Scanning the local subnet
- Activation
  - Self Activation
- Propagation Mechanism (Self Carried)
  - Exploiting a *fingered* *buffer overflow*
- Payload
  - None

Code Red I

- July 19, 2001: more than **359,000** computers connected to the Internet were infected by Code-Red I v2 worm in *less than 14 hours*



Code Red I

- Target Discovery
  - Scanning
- Activation
  - Self Activation
- Propagation Mechanism (Self Carried)
  - Exploiting a *Microsoft IIS Web Server buffer overflow*
- Payload
  - Defacement of websites

Code Red I

- Exploited *buffer overflow* in Indexing Service in Microsoft IIS Server
- Days 1-19 of each month
  - displays ‘hacked by Chinese’ message on English language servers
  - tries to open connections to infect randomly chosen machines using 100 threads
- Day 20-27
  - stops trying to spread
  - launches a denial-of-service attack on the IP address of www1.whitehouse.gon
- Code Red I v1
  - July 12, 2001
  - Used *static* seed for random number generator
  - Each infected computer tries to infect always the same IP addresses
  - Not very damaging, spread slowly
  - Memory resident
- Code Red I v2
  - July 19, 2001
  - Used *random* seed for random number generator

Code Red Damage

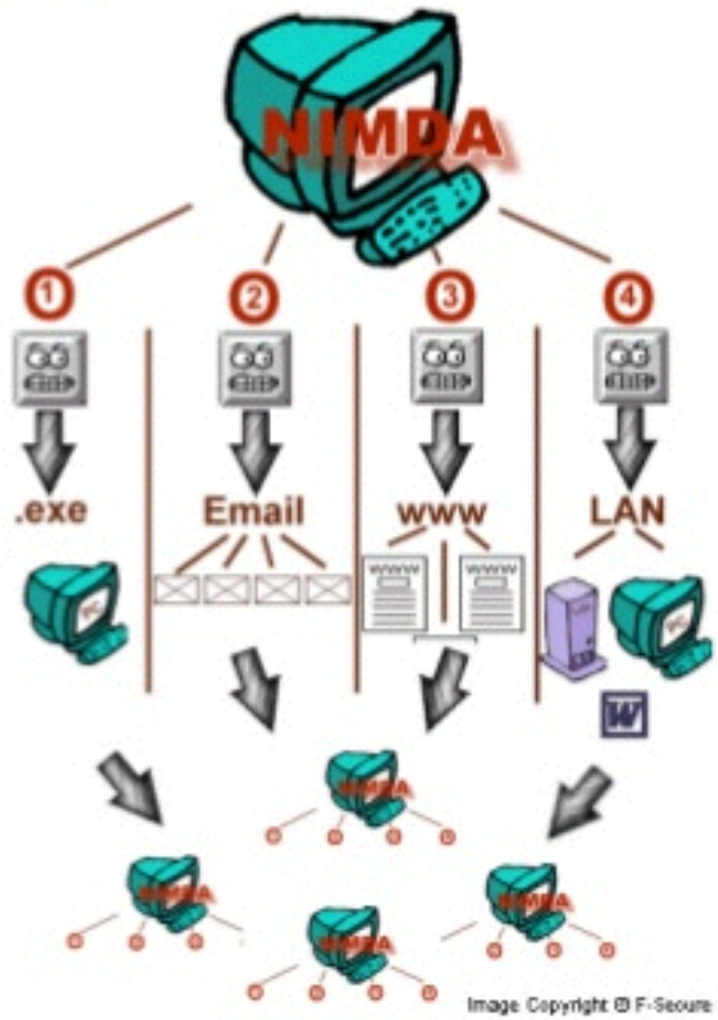
- 359,000 hosts infected in 24 hour period
- Between 11:00 and 16:00 UTC, the growth is exponential
- 2,000 hosts infected per minute at the peak of the infection rate (16:00 UTC)

Nimda (September 18, 2001)

- Target Discovery
  - Scanning, Email
- Activation
  - Self Activation, User action
- Propagation Mechanism (Self Carried)
  - Exploiting a *Microsoft IIS Web Server* *buffer overflow*
- Payload
  - Defacement of websites
- Multi-mode spreading:
  - attack IIS servers via infected clients
  - email itself to address book as a virus
  - copy itself across open network shares
  - modifying Web pages on infected servers w/ client exploit
  - scanning for Code Red II backdoor
- Spread across firewalls.

Nimda outbreak spreads worldwide (September 18, 2001)

- The worm spread by emailing itself as an attachment, scanning for--and then infecting--vulnerable Web servers running Microsoft's Internet Information Server software,
- Copying itself to shared disk drives on networks, and
- Appending Javascript code to Web pages that will download the worm to Web surfers' PCs when they view the page.
- Caused $530 million worth damages with in just first week of outbreak



ARIBA

SASSER Worm (2004)

- April 29, 2004

- Target Discovery
  - Random Scanning of IP addresses on TCP port 445,
  - can scan up to 1,024 addresses simultaneously

- Mode of Transmission
  - *Buffer Overflow* in Windows Local Security Authority Service Server (LSASS)

- Payload
  - Rootkit potential
  - Escalation of privileges

Witty (2004)

- March 19, 2004
- *Buffer overflow* vulnerability in ISS PAM module
- Single UDP packet exploits flaw in the *passive analysis* of Internet Security Systems (ISS) products.
- “Bandwidth-limited” UDP worm like Slammer.
- Vulnerable pop. (12K) attained in 75 minutes.
- Payload: *slowly corrupt random disk blocks*.
- Detailed telescope analysis reveals worm *targeted a US military base* and was launched from a European retail ISP account.

Slammer Worm – Before

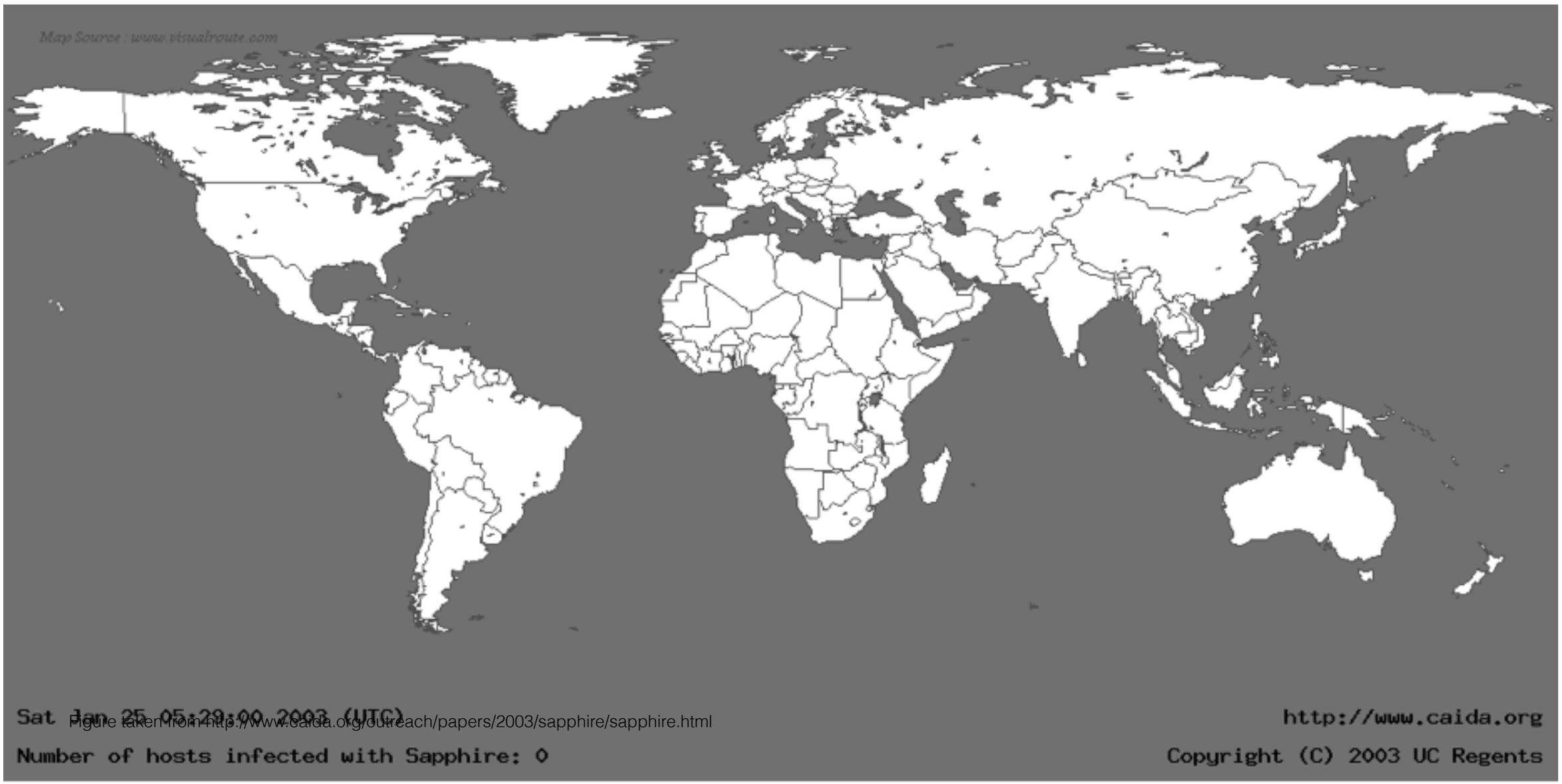
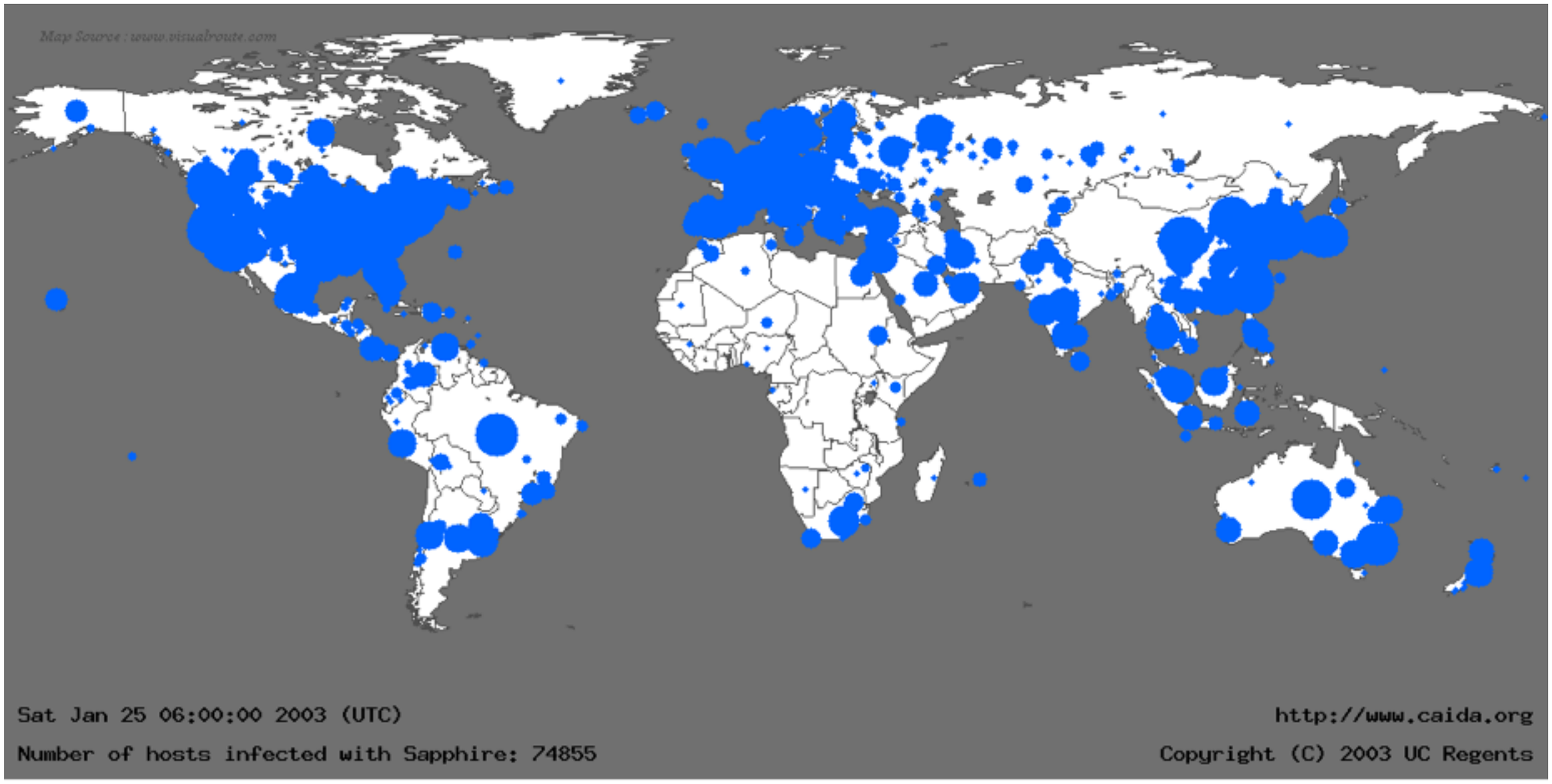


Figure taken from http://www.caida.org/outreach/papers/2003/sapphire/sapphire.html

Slammer Worm - After



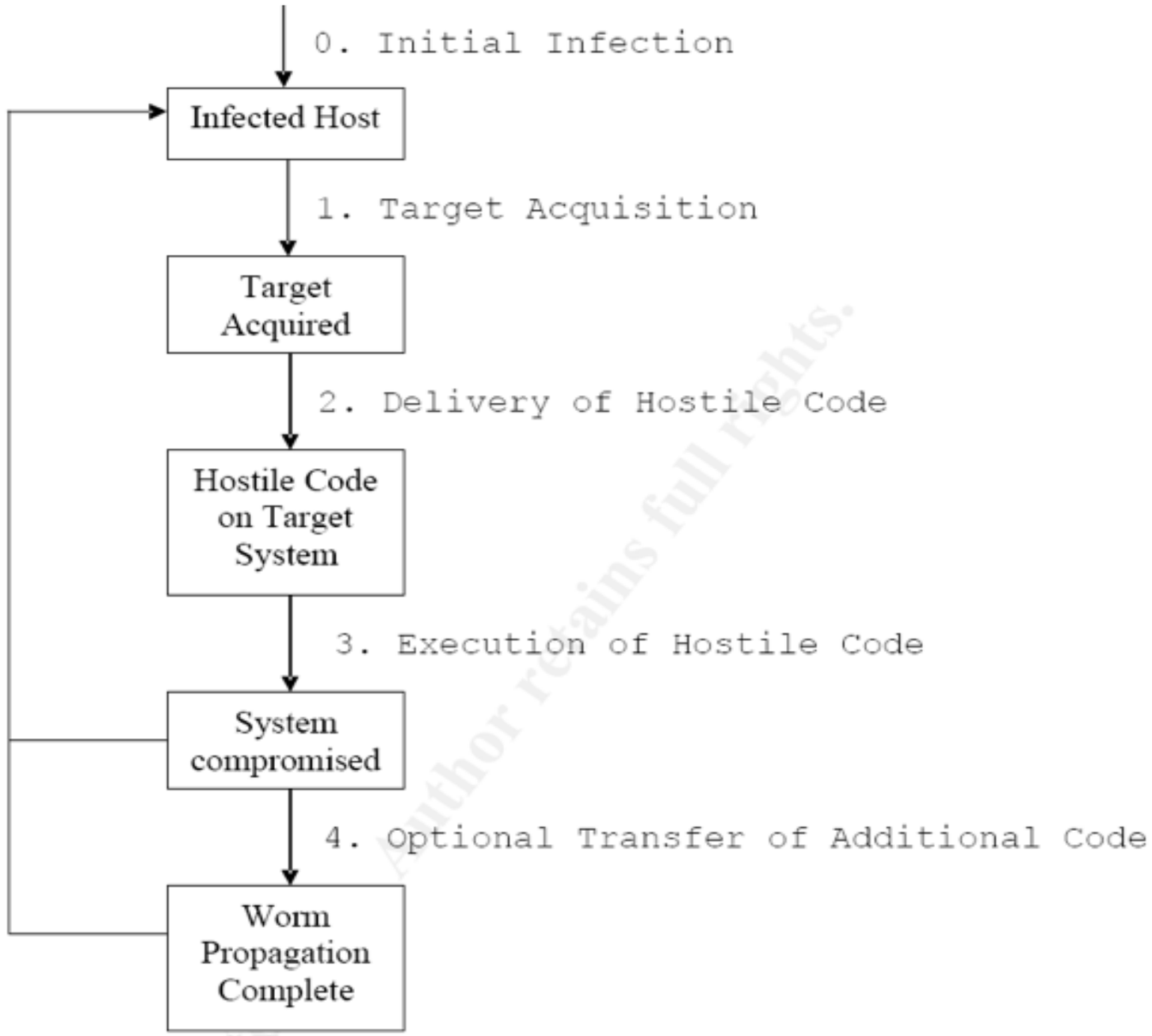
SQL Slammer

- The Slammer worm (also called Sapphire worm) consists of an IP scanner combined with an exploit for MS SQL Server, written in 376 bytes of code.
- Slammer exploited connectionless UDP service, rather than connection-oriented TCP.
- *Entire worm fit in a single packet!*
- Worm infected 75,000+ hosts in 10 minutes (despite broken random number generator).
  - At its peak, ***doubled every 8.5 seconds***

Slammer Worm

- Propagation speed was Sapphire's novel feature: in the first minute, the infected population doubled in size every 8.5 (±1) seconds.
- The worm achieved its full scanning rate (over 55 million scans per second) after approximately three minutes, after which the rate of growth slowed down somewhat because significant portions of the network did not have enough bandwidth to allow it to operate unhindered. Most vulnerable machines were infected within 10-minutes of the worm's release. Although worms with this rapid propagation had been predicted on theoretical grounds, the spread of Sapphire provides the first real incident demonstrating the capabilities of a high-speed worm.
- By comparison, it was two orders magnitude faster than the Code Red worm, which infected over 359,000 hosts on July 19th, 2001. In comparison, the Code Red worm population had a leisurely doubling time of about 37 minutes.

General Model of Worm Propagation



Source:http://www.sans.org/rr/whitepapers/malicious/1410.php

Summary of Worm Propagation

Worm propagation can be broadly described by a 3 (or 4) step process illustrated in the figure before:

0.) Initial Infection: The model begins with the presumption that there exists a system that is already infected by the worm and that the worm is active on this system.

1.) Target Acquisition: In order for the worm to propagate itself it must find additional systems to infect. Worms may actively target systems using:
a. IP addresses
b. Email addresses
c. File system traversal
It should also be noted that worms may passively target client system i.e. the trojaned web content delivered by web servers infected with the Nimda worm.

Worm Propagation

2.)Delivery of Hostile Code: Once a system has been targeted, it is necessary to transfer the worm to the targeted system in preparation for infection. Code delivery has been observed to take place via the following:

a. Network file systems b. Email

c. Web clients d. Remote command shell (or equivalent)

e. As part of packet payload associated with buffer overflows and similar programmatic exploits.

3.) Execution of Hostile Code: The presence of hostile code on a system is not sufficient for worm propagation; execution of the code must be triggered in some fashion. Code may be executed via:

a. Direct invocation from the command line (or equivalent)

b. Buffer overflow or other programmatic attack

c. Email clients d. Web clients

e. User intervention f. Automatic execution by target system.

4.) Some worms may only transfer a portion of their code in step 3. In that case it is necessary for them to transfer the remaining code once the target system has been compromised. This can be achieved via

a. FTP/TFTP

b. Network file systems

Benchmarks and Metrics

- Infection Size
  - Percentage of nodes infected
- Reaction Time
  - Time between detection of a worm and deployment of worm control measures
  - Obviously the lower the better
- Penetration Ratio
  - Number of nodes infected compared to the size of the possible domain
  - Related to infection ratio
- False Positives/Negatives