

## What is a Worm?

- A program that:
  - Scans network for vulnerable machines
  - Breaks into machines by exploiting the found vulnerability
  - Installs some piece of malicious code
  - Moves on
- Unlike viruses, worms don't need any user action to spread – they spread silently and on their own
- Unlike viruses, worms don't attach themselves onto other programs – they exist as a separate code in memory
- Sometimes you may not even know your machine has been infected by a worm

1

## What is a Worm?

- A program that:
  - Scans network for vulnerable machines
  - Breaks into machines by exploiting the found vulnerability
  - Installs some piece of malicious code – backdoor, DDoS tool
  - Moves on
- Unlike viruses, worms don't need any user action to spread – they spread silently and on their own
- Unlike viruses, worms don't attach themselves onto other programs – they exist as a separate code in memory
- Sometimes you may not even know your machine has been infected by a worm

2

## Why Are Worms Dangerous?

- They spread **extremely** fast
- They are silent
- Once they are out, they cannot be recalled
- They usually install malicious code
- They clog the network

3

## First Worm Ever – Morris Worm

- Robert Morris, a PhD student at Cornell was interested in network security
- He created the first worm with a goal to have a program *live* on the Internet in November 1988
  - Worm was supposed only to spread, fairly slowly
  - It was supposed to take just a little bit of resources so not to draw attention to itself
  - But things went wrong ...
- Worm was supposed to avoid duplicate copies by asking a computer whether it is infected
  - To avoid false “yes” answers, it was programmed to duplicate itself every 7<sup>th</sup> time it received “yes” answer
  - This turned out to be too much

4

## First Worm Ever – Morris Worm

- It exploited four vulnerabilities to break in
  - A bug in sendmail
  - A bug in finger daemon
  - A trusted hosts feature (/etc/.rhosts)
  - Password guessing
- Worm was replicating at a much faster rate than anticipated
- At that time Internet was small and homogeneous (SUN and VAX workstations running BSD UNIX)
- It infected around 6,000 computers, one tenth of then-Internet, in a day

5

## First Worm Ever – Morris Worm

- People quickly devised patches and distributed them (Internet was small then)
- A week later all systems were patched and worm code was removed from most of them
- No lasting damage was caused
- Robert Morris paid 10,000\$ fine, was placed on probation and did some community work
- Worm exposed not only vulnerabilities in UNIX but moreover in Internet organization
- Users didn't know who to contact and report infection or where to look for patches

## First Worm Ever – Morris Worm

- In response to Morris Worm DARPA formed CERT (Computer Emergency Response Team) in November 1988
  - Users report incidents and get help in handling them from CERT
  - CERT publishes security advisory notes informing users of new vulnerabilities that need to be patched and how to patch them
  - CERT facilitates security discussions and advocates better system management practices

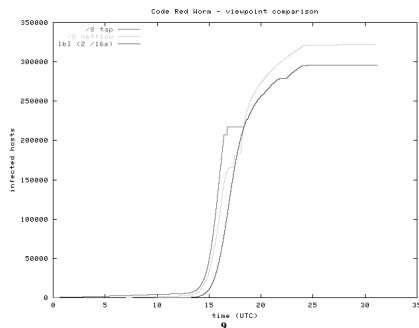
7

## Code Red v2

- Spread on July 19, 2001
- Exploited **the same vulnerability** in Microsoft Internet Information Server that allows attacker to get full access to the machine (turned on by default)
- Two variants – both probed random machines, one with static seed for RNG, another with random seed for RNG
- Infected more than 359,000 computers in less than 14 hours
- It doubled in size every 37 minutes
- At the peak of infection more than 2000 hosts were infected each minute

8

## Code Red v2



9

## Code Red v2

- 43% of infected machines were in US
- 47% of infected machines were home computers
- Worm was programmed to stop spreading at midnight, then attack [www.1.whitehouse.gov](http://www.1.whitehouse.gov)
  - It had hardcoded IP address so White House was able to thwart the attack by simply changing the IP address-to-name mapping
- Estimated damage ~2.6 billion

10

## Sapphire/Slammer Worm

- Spread on January 25, 2003
- The fastest computer worm in history
  - It doubled in size every 8.5 seconds.
  - It infected more than 90% of vulnerable hosts within 10 minutes
  - It infected 75,000 hosts overall
- Exploited buffer overflow vulnerability in Microsoft SQL server, discovered 6 months earlier

11

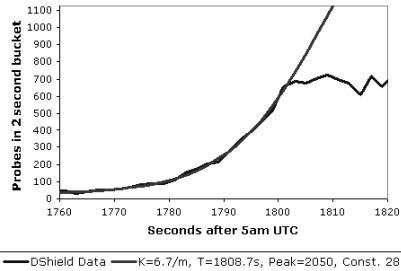
## Sapphire/Slammer Worm

- No malicious payload
- The aggressive spread had severe consequences
  - It created DoS effect
  - It disrupted backbone operation
  - Airline flights were canceled
  - Some ATM machines failed

12

## Sapphire/Slammer Worm

DShield Probe Data

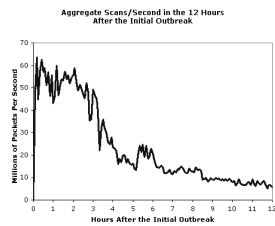


## Why Was Slammer So Fast?

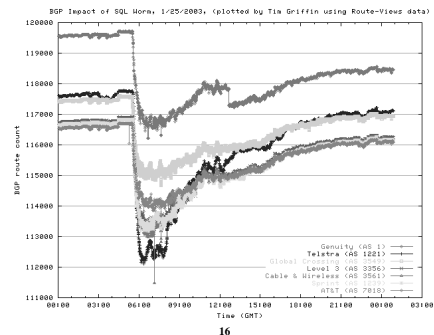
- Both Slammer and Code Red 2 use random scanning
  - Code Red uses multiple threads that invoke TCP connection establishment through 3-way handshake – must wait for the other party to reply or for TCP timeout to expire
  - Slammer packs its code in single UDP packet – speed is limited by how many UDP packets can a machine send
  - Could we do the same trick with Code Red?
- Slammer authors tried to use linear congruent generators to generate random addresses for scanning, but programmed it wrong

## Sapphire/Slammer Worm

- 43% of infected machines were in US
- 59% of infected machines were home computers
- Response was fast – after an hour sites started filtering packets for SQL server port



## BGP Impact of Slammer Worm

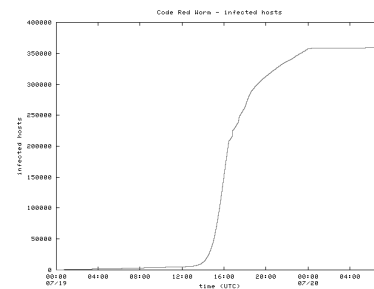


## Scanning Strategies

- All so far discovered worms use **random scanning**
- This works well only if machines have very good RNGs with different seeds
- Getting large initial population represents a problem
  - Then the infection rate skyrockets
  - The infection eventually reaches saturation since all machines are probing same addresses

"Warhol Worms: The Potential for Very Fast Internet Plagues", Nicholas C Weaver

## Random Scanning

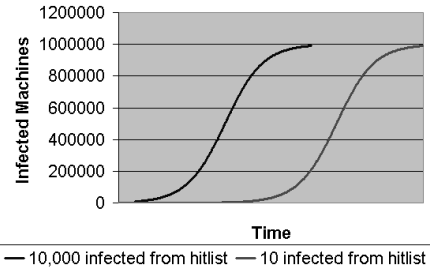


### Scanning Strategies

- Worm can get large initial population with **hitlist scanning**
- Assemble a list of potentially vulnerable machines prior to releasing the worm – a *hitlist*
  - E.g., through a slow scan
- When the scan finds a vulnerable machine, hitlist is divided in half and one half is communicated to this machine upon infection
  - This guarantees **very** fast spread – under one minute!

### Hitlist Scanning

Effect of hitlist size

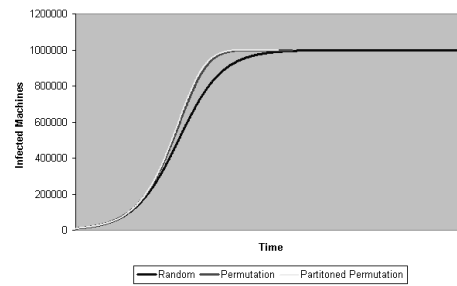


### Scanning Strategies

- Worm can get prevent die out in the end with **permutation scanning**
- All machines share a common pseudorandom permutation of IP address space
- Machines that are infected continue scanning just after their point in the permutation
  - If they encounter already infected machine they will continue from a random point
- **Partitioned permutation** is the combination of permutation and hitlist scanning
  - In the beginning permutation space is halved, later scanning is simple permutation scan

### Permutation Scanning

3 infection modes



### Scanning Strategies

- Worm can get behind the firewall, or notice the die-out and then switch to **subnet scanning**
- Goes sequentially through subnet address space, trying every address

### Worst Case Warhol Worm

- Hypothetical worm
- Uses vulnerabilities in Microsoft IIS (to spread to many places) and Microsoft Exchange (to spread beyond firewalls)
- Use hitlist scanning, subnet and permutation scanning, could spread in 15 minutes
- Malicious payload is activated upon installation but guaranteed not to slow down worm spread
  - E.g., overwrite random pieces of non-system files
  - Then DDoS-es some targets

## Infection Strategies

- Several ways to download malicious code
  - From a central server
  - From the machine that performed infection
  - Send it along with the exploit in a single packet