How to Own the Internet in Your Spare Time

Authors: Stuart Staniford, Vern Paxson, Nicholas Weaver
Publication Year: 2002
Presentation: Thomas Marchioro
Contents

Topic: **worms**, i.e., programs that self propagate across the Internet by exploiting security flaws in widely-used services.

Contribution of the authors:
- analyse relevant case studies of worms
- propose ideas for worm design to accelerate outbreaks
- suggest guidelines to contrast worms
Why study worms’ spreading?

Worms aim to spread quickly and compromise as much machines as possible.

Once a worm is identified and analyzed, its signature is made available to antiviruses
→ Game over for the attacker
Theoretical analysis of worms
CodeRed I

CodeRed I is a worm which started spreading across the Internet in July 2001.

Spreading strategy: infected machines generate IP addresses at random and target them.

Two versions of CodeRed I:

▪ V1: fixed seed for RNG → Linear spread: not many machines compromised
▪ V2: variable seed for RNG → Exponential spread: over 350,000 machines infected
Analysis of CodeRed I as an epidemic

The spread of CodeRed I v2 can be analysed as an epidemic where:

- There is no recovery
- Compromised machines willingly infect other machines

Assumption for the theoretical analysis:
Homogeneous mixing, which in this context means

- Every machine can possibly get in contact with every other machine
- Every compromised machine selects its targets uniformly at random
Compartmental model

\[ N = \text{total vulnerable machines (compromised and not)} \]
\[ A = \text{number of compromised machines ("zombies")} \]
\[ S = \text{number of vulnerable but not compromised machines} \]
\[ K = \text{initial compromise rate (# of machine that a zombie can contact in one unit of time)} \]
Intuition

(Assume $K = 1$)

$\Delta A = \# \text{ of machines contacted by one zombie (} K \text{)} \cdot \# \text{ of zombies (} A \text{)}$

$t = 0 \rightarrow A(t) = 1$

$t = 1 \rightarrow A(t) = 1 + 1 \cdot 1 = 2$

$t = 1 \rightarrow A(t) = 2 + 1 \cdot 2 = 4$

$t = 2 \rightarrow A(t) = 4 + 1 \cdot 4 = 8$

... Exponential growth at the beginning
Intuition

(Assume $K = 1$)

What happens when half of the hosts are infected?

$$A(t) = \frac{N}{2} \rightarrow A(t + 1) = \frac{N}{2} + \frac{N}{2} \cdot 1 \cdot \frac{1}{2} = \frac{3N}{4}$$

Saturation phenomenon:

$$A(t) = 0.99N \rightarrow A(t + 1) = \frac{99N}{100} + \frac{99N}{100} \cdot 1 \cdot \frac{1}{100} = 0.9999N$$
Rigorous analysis

New zombies created in a time interval $\Delta T \ll 1$

Number of machines contacted in an interval $\Delta T$

$$\Delta A = (KA \cdot \Delta T)^{\frac{S}{N}}$$

Fraction of non infected machines
(Probability of getting in contact with a non infected machine)
Continuous model

When $N \gg 1$ we can approximate the discrete model with a continuous model:

- $a =$ fraction of compromised machines
- $s =$ fraction of vulnerable but not compromised machines
- $K =$ initial compromise rate (# of machines infected per hour by a single machine)

\[
\begin{align*}
\begin{cases}
d a &= K a s \, d t \\
s + a &= 1
\end{cases}
\Rightarrow d a = K a (1 - a) d t
\end{align*}
\]
Analysis of CodeRed I as an epidemic

\[ da = Ka(1 - a)dt \]

\[ \int \frac{1}{a(1-a)} da = \int K \, dt \]

\[ \int \left( \frac{1}{a} + \frac{1}{1-a} \right) da = \int K \, dt \]

\[ \log a - \log(1 - a) = Kt + c \]

\[ \log \left( \frac{a}{1-a} \right) = K(t - T) \]

\[ -1 + \frac{1}{1-a} = e^{K(t-T)} \]

\[ 1 - a = \frac{1}{1 + e^{K(t-T)}} \]

\[ a(t) = \frac{e^{K(t-T)}}{1 + e^{K(t-T)}} \]
Analysis of CodeRed I as an epidemic

Epidemic curve for $K = 2.6, T = 5.52$

Exponential growth

saturation
Analysis of CodeRed I as an epidemic
ANY QUESTIONS?
Can we do “better”?
Localized scanning

CodeRed I v2 generates IP addresses at random.

Problem 1: some addresses are unreachable because of firewalls.

Idea: **Localized scanning** → scan IP addresses within the local network of the infected host.
CodeRed II

Example of worm employing localized scanning

Spreading strategy:
- with probability $\frac{3}{8}$ choose target from class B address space ($/16$)
- with probability $\frac{1}{2}$ choose target from class A address space ($/8$)
- with probability $\frac{1}{8}$ choose a random IP
Multi-vector worms

Another idea: use more than one strategy → Multi-vector worms
Example: Nimda

Spreading strategies:
1. Infect Web servers (hubs) from infected clients
2. Add malicious code to the Web pages of infected servers
3. Spam emails with the worm as an attachment
4. Copy the worm across open network shares
5. Scan for backdoors left by other worms
Can we do “better”? (Theory)

CodeRed II v2 generates IP addresses at random.

Problem 2: at the beginning exponential growth is “slow” $\rightarrow$ It takes a lot to infect the first 10,000 hosts

Ideas:
- Hit-list Scanning
- Permutation Scanning
- Topological Scanning
- Flash worms
Hit-list scanning

Strategy:
- scan in advance for a list of vulnerable hosts and make a list
- when infecting a host, send half of the list to it
Scanning strategies

How to scan vulnerable hosts without drawing attention?

- Stealthy scans → do portscans at random (takes a lot of time)
- Distributed scanning → infect few hosts and use them for scanning
- DNS searches → check for domain spamlists available online
- Spiders → use engines for Web crawling
- Public surveys → e.g., Netcraft survey
- Just listen → P2P networks often end up advertising their servers
Permutation scanning

Prevent inefficiency of random scanning (hosts targeted multiple times)

Assumption: the worm can detect if a target is infected or not

Strategy:
- all the infected hosts share the same permutation of the address space
- if the targeted host is not infected, infect it and contact the next
- If the targeted host is already infected, choose a new starting point at random
Permutation scanning

Example of execution:
- Try to contact IP 1 → IP 1 is not infected → Infect IP 1 and go to IP 2
- Try to contact IP 2 → IP 2 is not infected → Infect IP 2 and go to IP 3
- Try to contact IP 3 → IP 3 is already infected → Go to IP r
- Try to contact IP r →...
Warhol worms (hit-list + permutation)
Topological Scanning

Alternative to hit-list scanning exploiting network topology information.

Strategy: scan the infected host’s memory looking for possible targets (e.g., email contacts, URLs)
→ More likely to target hubs
Flash Worms (infect the Internet in 30 sec)

Strategy:
- do a complete scan of the Internet
- create a huge list with all the possible hubs

How large would be the list?
→ According to the article (2002) \(\approx 48\) MB, which would make the worm really slow in its early stage

Solution: target first the hubs with “fastest” links (in terms of capacity)
Alternative to rapid spread: stealth worms

Remember: the techniques seen before aim to make a worm spread as quickly as possible, assuming that once a worm is detected and countermeasures are taken it is game over.

Possible alternative: not having your worm detected

→ “Stealth” worms
Alternative to rapid spread: stealth worms

Contagion model:

▪ Assume we have 2 exploits: $E_S$ for servers, $E_C$ for clients

▪ Servers infected with $E_S$ follow the protocol:
  ▪ when a new vulnerable client contacts the server
    1. compromise it with $E_C$
    2. send the pair $(E_S, E_C)$ to it

▪ Clients infected with $E_C$ follow the protocol:
  ▪ when the legitimate user contacts a new server
    1. compromise it with $E_S$
    2. send the pair $(E_S, E_C)$ to it

No anomalous traffic is generated
P2P stealth worm

P2P systems are among the best targets for stealth worms because:
1. Peers are both clients and server → $E_s = E_c$
2. They interconnect many different peers → lot of targets
3. Protocols are view as “not mainstream” → receive less attention
4. Often execute on user’s desktop → easier to retrieve sensitive data
5. Used to transfer “grey” content → users are less inclined to report anomalies
Example: KaZaA
Fig. 4. TorrentNet degree distribution using the full network and one with the top 100 IPs filtered.
Update and control

How can an attacker control its “zombies” after the dissemination of a worm?

Centralized control? → The source can be identified and its communications blocked
Updates on a Web page? → It can be removed

Solution: distributed control
Update and control

Distributed control:
- each zombie must have a list of other zombies to propagate the update
- each copy of the worm must be able to create encrypted communication channels
- each update must have a unique ID and to be signed by the attacker’s key

Programming updates:
- main code and updates must be written in a flexible language
ANY QUESTIONS?
How to contrast worms?
Identification of outbreaks

Warhol and flash worms create anomalies in the network traffic.

Human-driven analysis is arguably too slow to be effective before a worm has infected the entire network

→ Identification must be **automatized**
Analysis of malicious code

Once a worm is identified, we need to understand
1. how it spreads
2. what it does in addition to spreading

→ Development of a signature to be propagated rapidly to enable the identification of the worm by vulnerable machines
Anticipation of new vectors

Need to study and identify possible worm signatures before an attacker actually employs them.

This can be done by analyzing well-established and emerging systems:

- What kind of network they create? What is its topology? Are there hubs or points of failure?
- What are likely exploit strategies to be used by an attacker?
Proposal of the authors: CDC

Center for Disease Control (CDC)

“Its mission would be to monitor the national and worldwide progression of various forms of disease, identify incipient threats and new outbreaks, and actively foster research for combating various diseases and other health threats.”
Nowadays there is still no CDC

How to Save the Net: A CDC for Cybercrime

The Internet may be made up of software and hardware, but it is an ecosystem that depends on a key human value: trust. The networks and systems must be able to trust the information we are sending, and in turn we have to be able to trust the information we receive.

This system of trust has allowed businesses around the world to share data rapidly and reliably on almost every issue—except their own security. Too many firms are still unwilling to share crucial information about the network attacks, data breaches, and outright cybertheft they’ve experienced—and what they do to defend themselves. Companies keep everything from basic facts to crucial technical details from one another and, notably, from the government, largely because they’re suspicious and fearful about what others might do with that information. The fear runs the gamut:

Do We Need a CDC for Cybersecurity?

A centralized cybersecurity agency could leverage the mission and approach of the U.S. Centers for Disease Control and Prevention to proactively respond to cyber threats in real-time.

By Madeline Weiss

Although the U.S. Centers for Disease Control and Prevention (CDC) has emerged from the Ebola outbreak with a somewhat diminished reputation, its mission (“works 24/7 to protect America from health, safety and security threats, both foreign and in the U.S.”) and approach (“conducts critical science and provides health information that protects our nation against expensive and dangerous health threats and responds when these arise”) are as vital as ever.

Would such a proactive mission and approach help protect America from cybersecurity threats, both foreign and in the U.S.? David Bray, visiting associate at the University of Oxford, made the case for “a CDC for cybersecurity” at a recent Advanced Practices Council (APC) meeting. Based on his research at Oxford, Bray built up his case by reminding APC members of the current grim reality.
Summing up

- We have done a theoretical analysis the spreading of a worm and shown the exponential increase of the infected machines in its early stage
- We have discussed some strategies to make the spread even faster
- We have listed the main steps to be done to contrast a worm
ANY QUESTIONS?