
Network Worms, What is possible.

By Jonathan Wignall
(jwignall@dnscon.org)

For Defcon 11, August 2003

What is a network worm?

■ An independent program that seeks out new hosts, from an existing host in order to further spread itself.

■ Other definition:

■ Programs which are able to replicate themselves (usually across computer networks) as stand alone programs (or sets of programs) and which do not depend on the existence of a host program are called computer worms [5]


■ Differences can be minor between a worm and a virus, as many worms can be ‘carried’ to infect systems in the same way as a virus.

History of worms: The originals

- ❑ **Science fiction** references (i.e. Brunner's "tapeworm" program in "shockwave rider" 1976)
- ❑ **Xerox** work in 1982, Shock and Hepp coined the use of the term "worm" and carried out experiments with worm like programs [6]
- ❑ **CHRISTMA EXEC** from 1987 that spread via email and required the user to execute it.
- ❑ **Internet worm** in 1988. The morris worm infects close to 10% of the then internet (6000 machines).
- ❑ **IRC Worms**, from 1987 to present day worms have targeted IRC clients (Mirc and PIRC)

History of worms: Email

Melissa(March 1999)

 A worm/virus hybrid that sent mail to the first 50 users in the outlook address book, containing the worm/virus. Could also spread as a conventional macro virus.

KAK(February 2000)

 A VBS worm similar to bubbleboy that exploited a hole in outlook to autoexecute on receipt.

Love Letter (May 2000)

 Another VBS worm that worked like Melissa, but was also able to spread via IRC

History of worms: 24 months ago

☐ In the last two years we have seen a resurgence of **non email** distributed worms

☐ **CODE RED, and variants (Summer 2001)**

☐ Exploited a buffer over flow in IIS [CVE-2001-0500] to compromise over 360,000 systems in 14 hours.

☐ Used both random IP address selection to target sites, and local address scanning (CRv2) to break into targets.








☐ Sites compromised with CRv2 backdoored for further access, and worms broadcast attempts to break into other systems they advertised the vulnerability.

☐ Caused some internet communication problems from traffic levels, and also crashed many NT machines instead of infecting them.

☐ With CRv1 infected machines launched a DoS attack against a specific IP address.






History of worms: 24 months ago

NIMDA (September 18th 2001)

-  An example of a multi-vector worm that used several methods of replication to remain a problem to this day.
 -  Infected web servers via probing using a microsoft IIS vulnerability [CVE-2000-0884]
 -  Scanning for backdoors left by CODERED 2 and sadmind
 -  Using open network shares to replicate
 -  Adding itself to webpages on a compromised box, in the hope that the system was a webserver and unpatched clients viewing the pages would be infected.
 -  By emailing itself out from an infected machine.
-  Was able to infect many windows 2000 systems with a similar propagation speed to CODERED2

History of worms: 6 months ago

Slammer / Sapphire worm, January 2003

-  Used security vulnerabilities [MS02-039],[MS02-061] discovered by David Litchfield who published exploit code at Blackhat 2002.
-  Sent a single packet to UDP port 1434 by random selection of IP address.
-  Small size [376 byte transmission] and coded in assembler leading to rapid transmission with a hit peak rate estimated at 55 million scans per second.
-  Fastest spreading worm yet, most hosts compromised inside the first 10 minutes of launch.
-  Carried no damaging payload, but some parts of the internet suffered significant packet failure due to traffic overload.

[8]

Methods of replication

Two main methods exist for worms to spread:

1. Use legitimate services.

Email

Sircam + previous examples

File Shares

Sircam + Deloder

2. Exploit system vulnerabilities

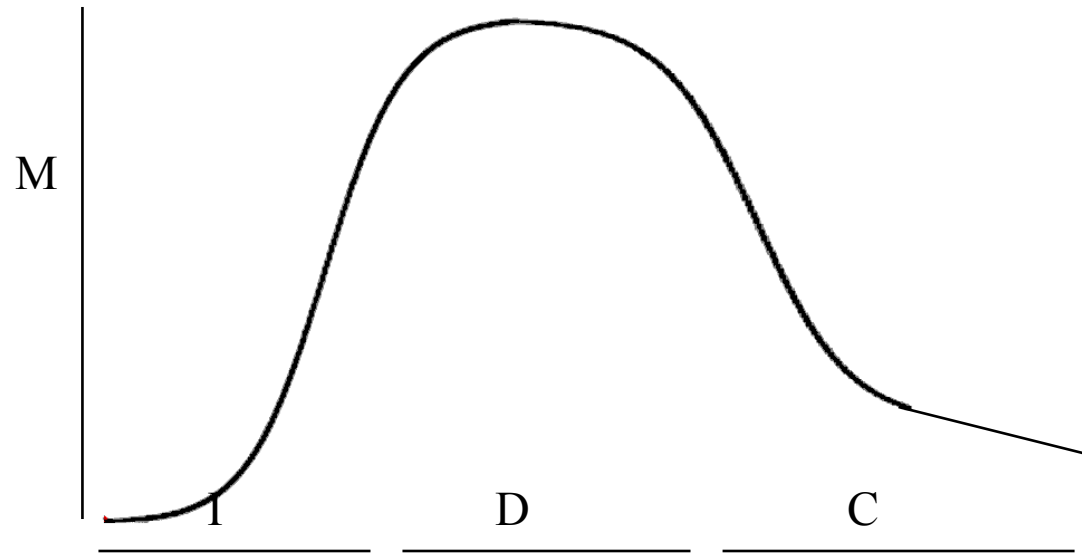
Webservers


NIMDA and CODERED

MS SQLServer


Slammer


The worm life cycle



 I = Infection time to 95% of maximum infections

 M = Maximum number of infected nodes

 D = Duration of peak infections

 C = Containment time needed to reduce infection to 5% of maximum

 The ideal worm has a low I, high M, and a high C

Explanation of life cycle

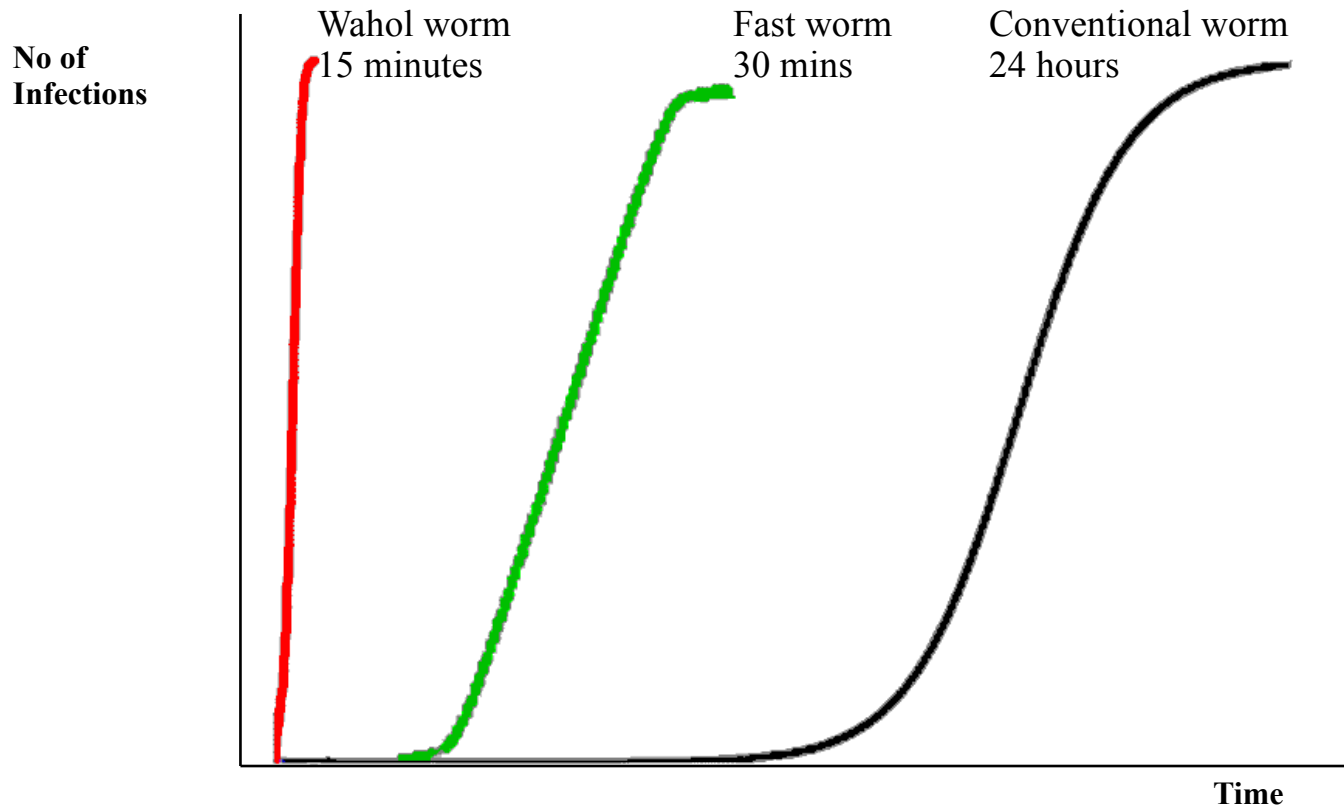
- Low value of I means a faster spreading worm.
- High value of M means a large number of machines has been infected.
- High value of D means a populations that is unaware its infected, or the removal of infections is kept in balance by the number of new infections.
- High value of C means machines infected not cleaned as a matter of priority, or not being taken out of service.

An 'ideal' worm

- ☐ A worm needs an effective replication engine, and ideally an efficient one.
- ☐ Needs to be a small size to prevent its own distribution being hindered by traffic constraints.
- ☐ Target inexperienced users, to reduce chance of removal.
- ☐ Have a large contactable population to infect.
- ☐ Worms payload not easily countered.
- ☐ Avoid detection for as long as possible.
- ☐ Adapt to use new exploits and counter removal methods.[1]

Developments in replication

Time for worm to reach system < (Detection time for worm + Time to counter) [7]



Can we have worms that replicate so fast, that no one has chance to react before they reach their computers?

Fast replication

- ☐ Worms that are able to efficiently and speedily infect many systems.
- ☐ Example being Slammer worm.
- ☐ Can obtain saturation coverage in around 30 minutes.
- ☐ Too fast for human's to detect, process, work out an answer and patch the system.
- ☐ Impossible for most patches to be installed in that time frame.
- ☐ Requires small, well written code.
- ☐ But will result in duplication of infection attempts.

Warhol

“in the future, everybody will have their 15 minutes of fame” Andy Warhol

Named by Nicholas Weaver in 2002[4]

Use a hit list of target machines for the worm to spread to on its first few thousand victims in order to reduce the lead time in building up a population of the worm.

Requires a list of vulnerable targets to be precalculated in advance before the worm is launched.

Warhol (cont)

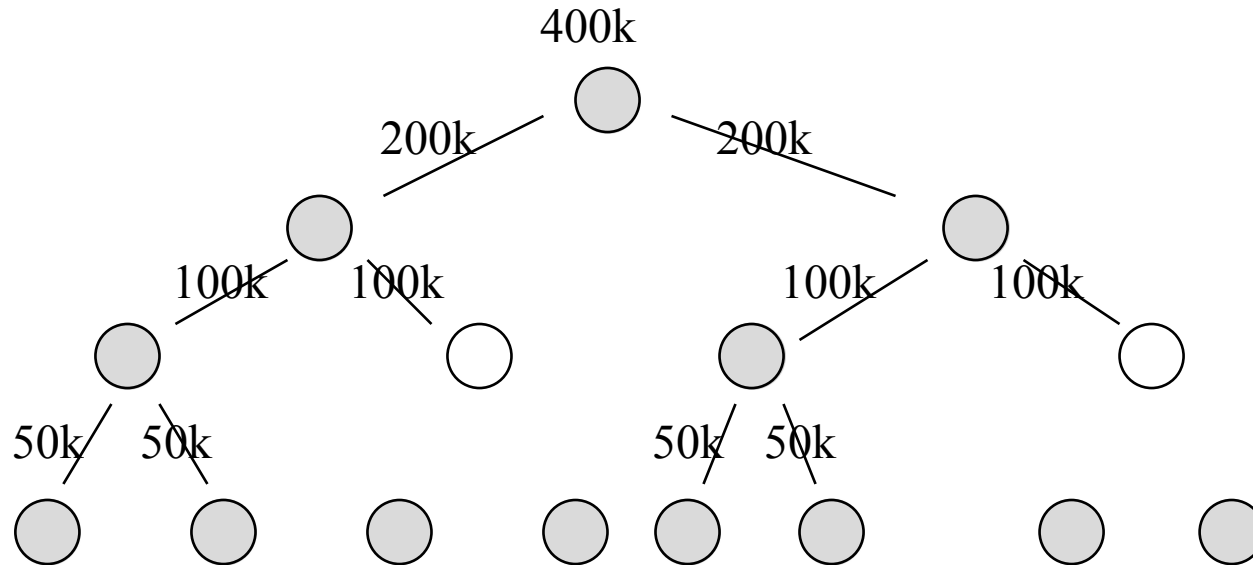
- After the target list is used up, the worm will spread by Permutation Scanning.
- This technique involves the worm scanning a randomly assigned, setsize block of IP address space. If it detects an existing copy of the worm in that address space it will move to another randomly chosen address space in order to locate host computers.
- This reduces the duplication of effort by the worm, provided the random number generator of the worm produces different 'random' selections on different hosts.

Flash worms

Named by Stuart Staniford in 2002, [3]

- Mass coverage inside 15 seconds? Flash worms ‘in theory’ can achieve this.
- Each copy of the worm carries with it a set of targets, it uses some to target new hosts, and sends along a portion of the remaining addresses to each of its ‘children’.
- The result being as a worm makes more copies of itself, each copy get smaller, keeping the worms network load predictable.
- Requires high bandwidth for first few infections (10 million addresses = 40 megs!)

Example of 'Flash' infection



● = Infected computer.

● = computer that could be infected, but hasn't

○ = computer that wasn't successfully infected

Flash worms (cont)

Disadvantages: Requires high speed connections to meet 15 second saturation infection target. Can lose parts of the address space due to errors and miss some precalculated targets. Can only effectively target known vulnerable systems.

Stealth worms

- ❑ Old virus concept of avoiding detection by having a slow replication rate.
- ❑ Hiding your infection mechanism by appearing to be 'normal' traffic.
- ❑ Slowly building up a large infection base, with the hope of payload triggering at a set date.
- ❑ Or store record of infection to allow worms to be 'activated' in future, by an authorised signed message sent instantly along the infection path.
- ❑ Popular with diseases that show no symptoms, harder to achieve in the world of IDS.

Companion worms

- ❑ The concept of carrying of a worm, via another worm.
- ❑ These worms could be for different platforms, operating systems or hardware.
- ❑ Enhancement of the old ‘dropper’ virus principle, but instead of dropping a virus as a payload the system scans a target and deploys the correct worm for that platform.
- ❑ Can use a cluster of worms designed for different systems.

Companion worms (cont)

Example: An infected windows machine has the UNIX worm stored as data. It scans a UNIX host and is able to break in. The UNIX machine is infected with the UNIX version of the worm and the windows version of the worm is copied to the UNIX box as data. The cycle can now repeat with the UNIX worm scanning for both Windows and UNIX targets.

This way each member of the 'cluster' of worms can replicate with the others assistance. The size of the cluster limits how many worms can form the cluster.

Drawback: Requires twice the effort to write than a standalone worm.

Power worms

named by Brandon Wiley 2002 [1]

- The concept of a worm with a modifiable exploit module that can be updated across the internet. The worm forms a distributed patching system that aims to ensure the worm is always able to break into systems and is able to prevent its removal by jamming sites providing any patches.
- The worm would ensure no duplication of scanning of systems, by communicating between the infected machines address information.

Power worms (cont)

■ Many questions exist about the feasibility of such an approach.

■ **Some Drawbacks:** For this system to work, the worm would have to be designed so no one else could introduce a set of dummy instructions that could disable the worm (use digital signatures?), achieve total control of the majority of hosts (require different version of the worm for different platforms?) and would require a constant stream of new exploits (would exploits continue to be published in such an environment?). Also who is going to risk updating this system with new instructions? Wouldn't antivirus companies just distribute patches via CD?

Power worms (cont)

- ❑ Could the worm be modified to find its own exploits? Its a distributed network that's got a large sample of hosts to experiment on.
- ❑ Requires a level of development not seen before in worm construction, but could turn the internet into a single cluster that does only what the worm wants it to do.

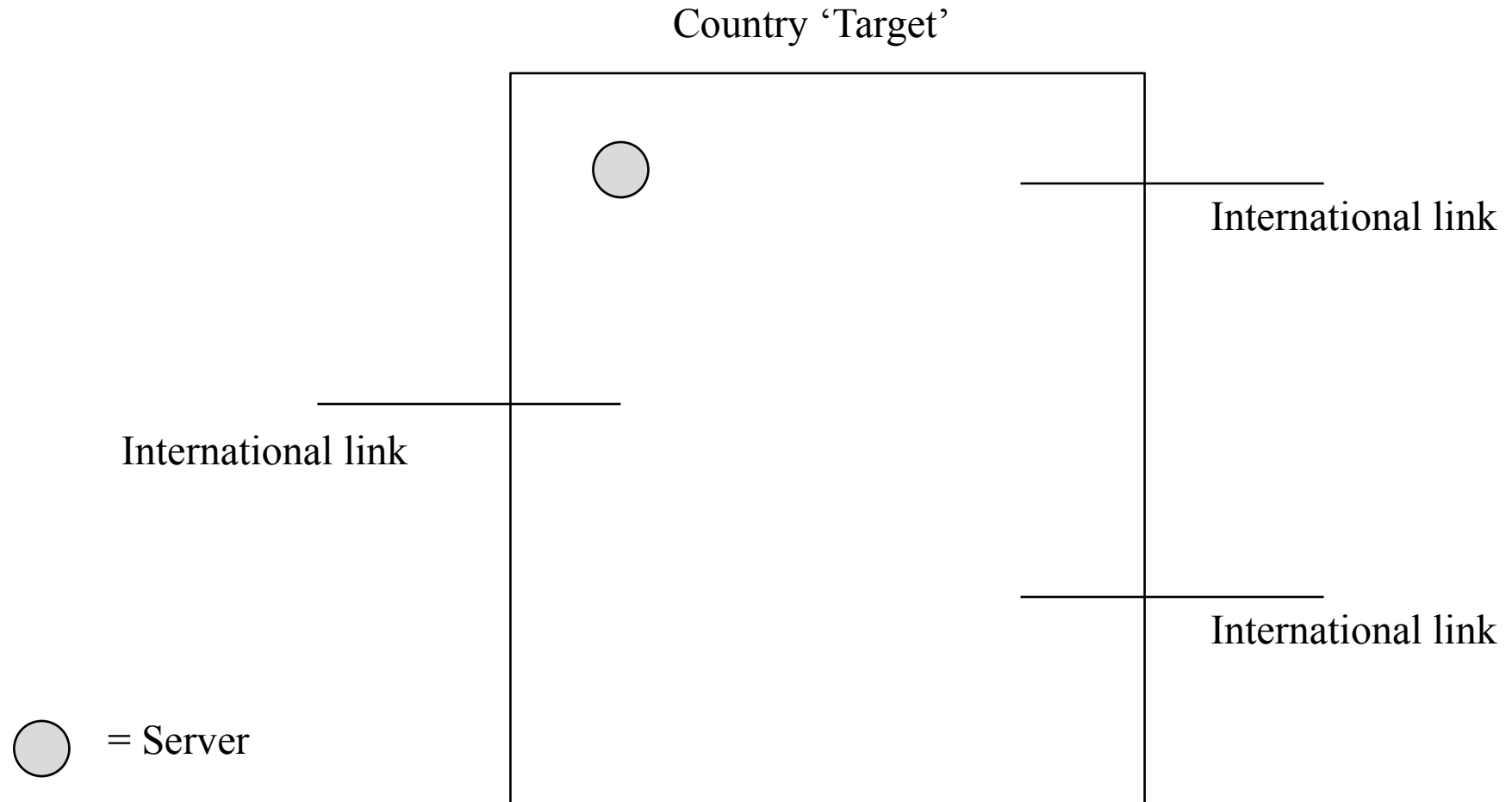
Worms in information warfare

- Most sources dismiss the use of worms in information warfare as they can't be targeted accurately. Network worms can cause as much damage to a high tech attacker as the target.
- This is incorrect, as its possible to develop '**Server Controlled**' worms.
- With this type of network worm the worm doesn't carry a set of target addresses, or its own target finding capability, but instead uses addresses provided by an installed server on request.

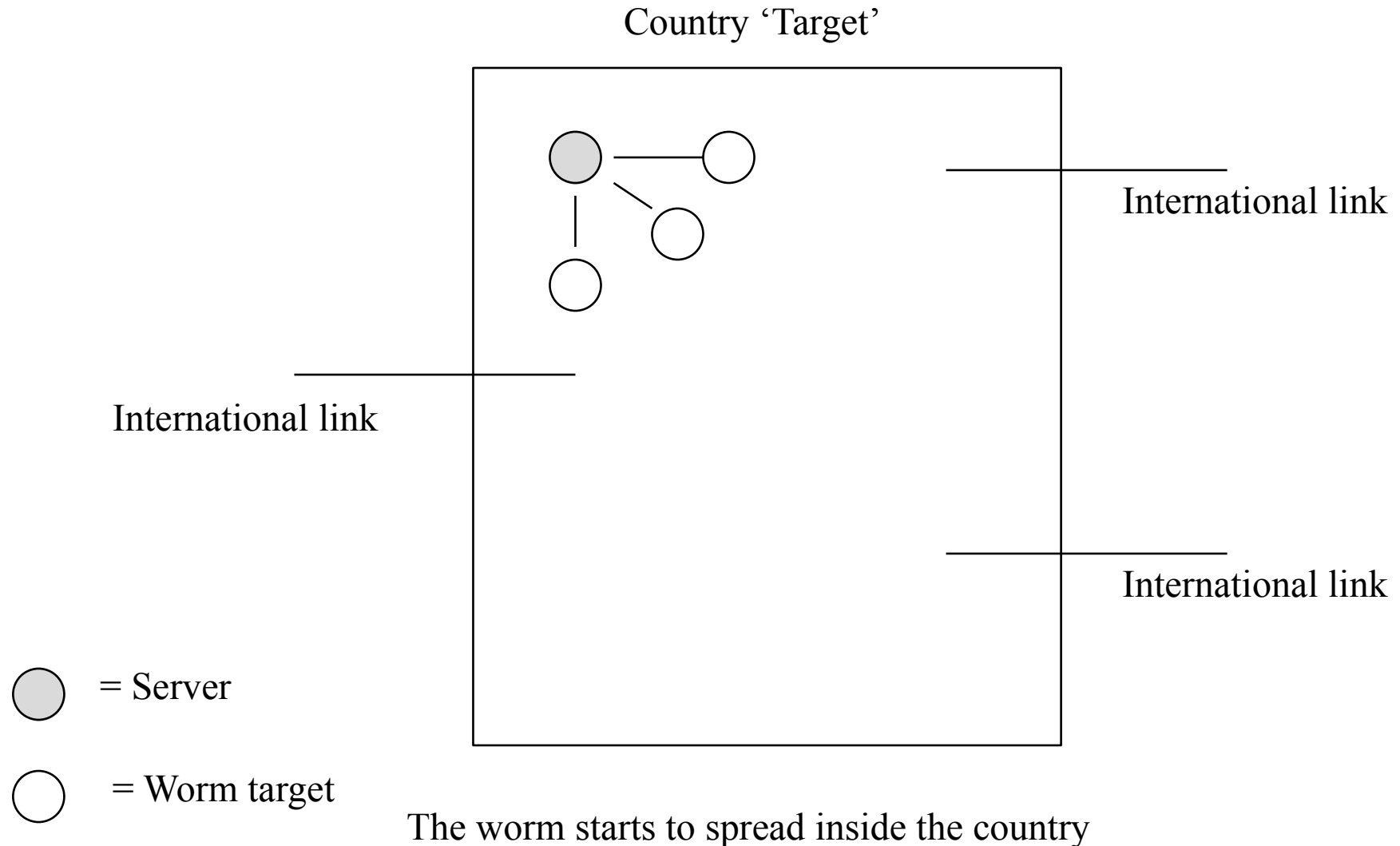
Server controlled

- ☐ The server provides on request to the worm a set of addresses to target for infection, as well as a target address/time delay for any payload. For 100 infection targets addresses, one DoS target and a time trigger this would be a 412 Byte packet. Multiple requests could be made to the server from a worm.
- ☐ This system ensures each computer on the target list is called only once, without missing any on the list.
- ☐ Has similar replication speed to flash worms.

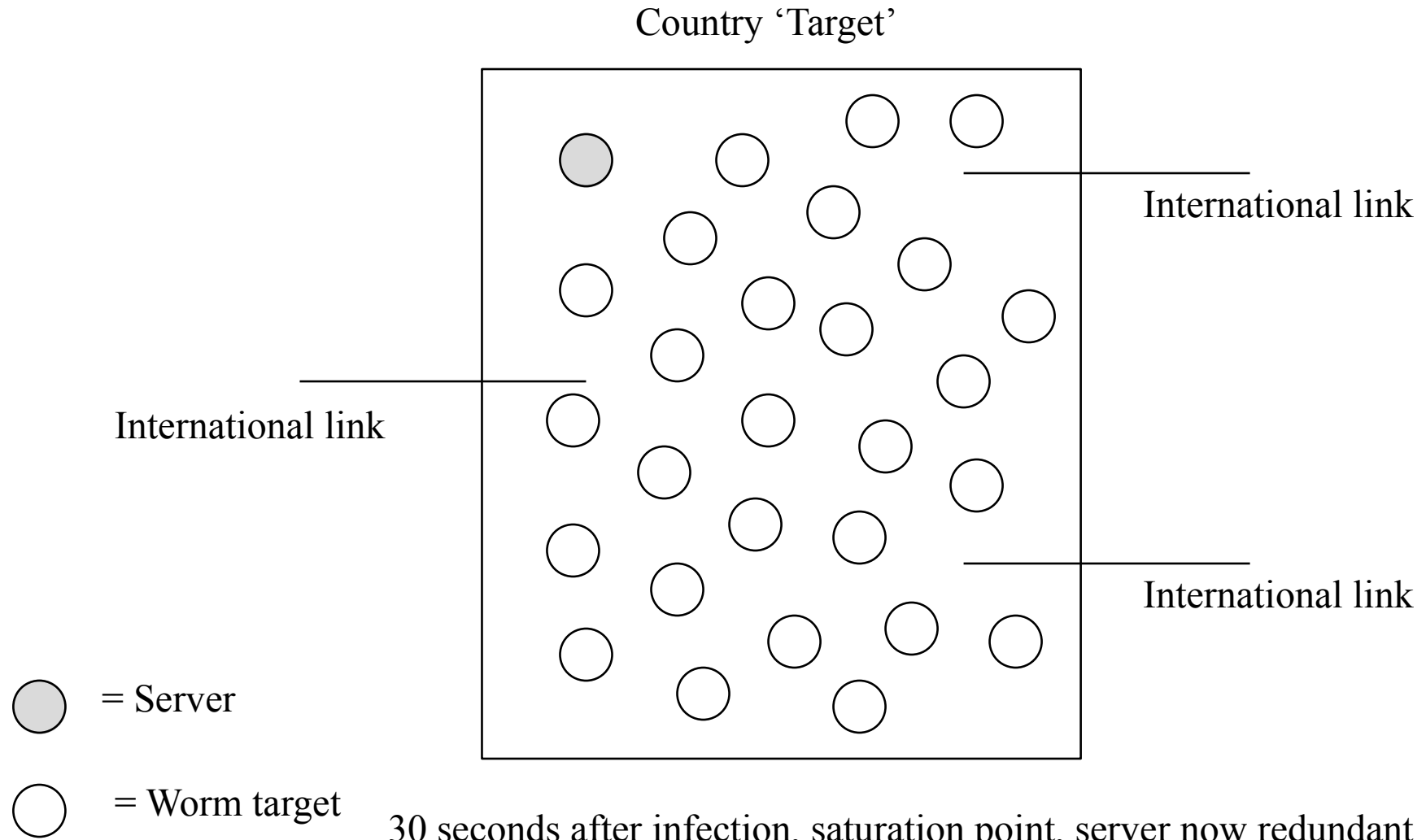
Phase 1: The server is installed



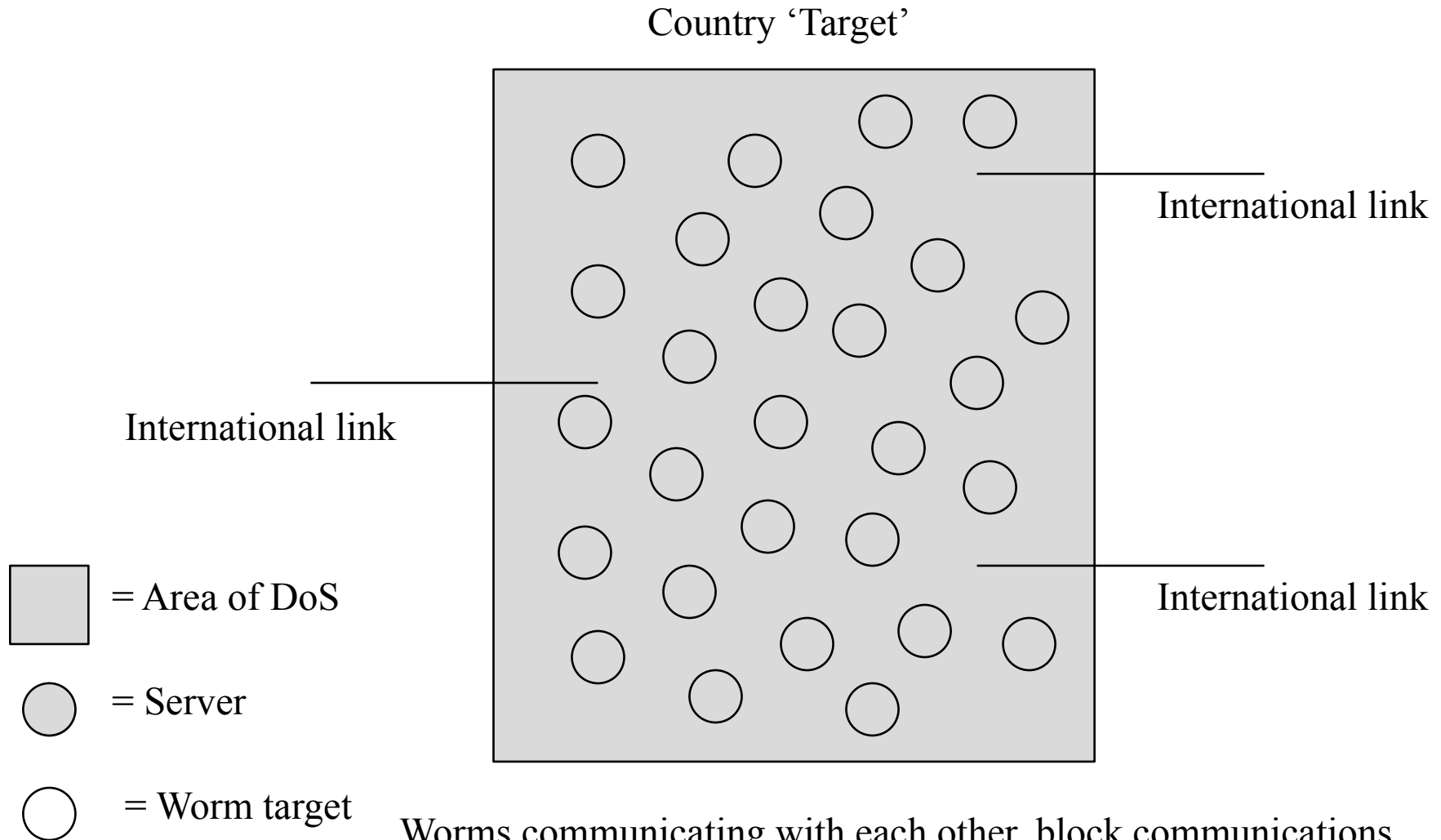
Phase 2: The server is activated



Phase 3: Target number reached



Phase 4: Activation



Worms communicating with each other, block communications inside country, with some leakage via BGP to neighbours.

Server controlled (cont)

Drawbacks: The server would be limited in the number of worm requests it can handle, around 13 thousand worm requests per second, but multiple servers could be used. If the server is taken down within distribution time frame, distribution of the worms will stop. If the server is address blocked by several ISP's within the time frame, worm distribution will be slowed. Some leakage via BGP of traffic will occur to neighbouring countries networks, but not suffer infection by the worm itself.

New ways worms can spread

- ☐ ISP's make common simple mistakes that apply to large number of computers, that are easy to exploit.
- ☐ Same applies to manufacturers.

AOL

FTP for all members web space on AOL.

server: members.aol.com

Id: anonymous

Password: userid@aol.com

cd userid

upload to account web space via 'put' command.

“<http://msdn.microsoft.com/library/default.asp?url=/workshop/author/hta/overview/htaoverview.asp>”

The Power of Trust: HTAs and Security

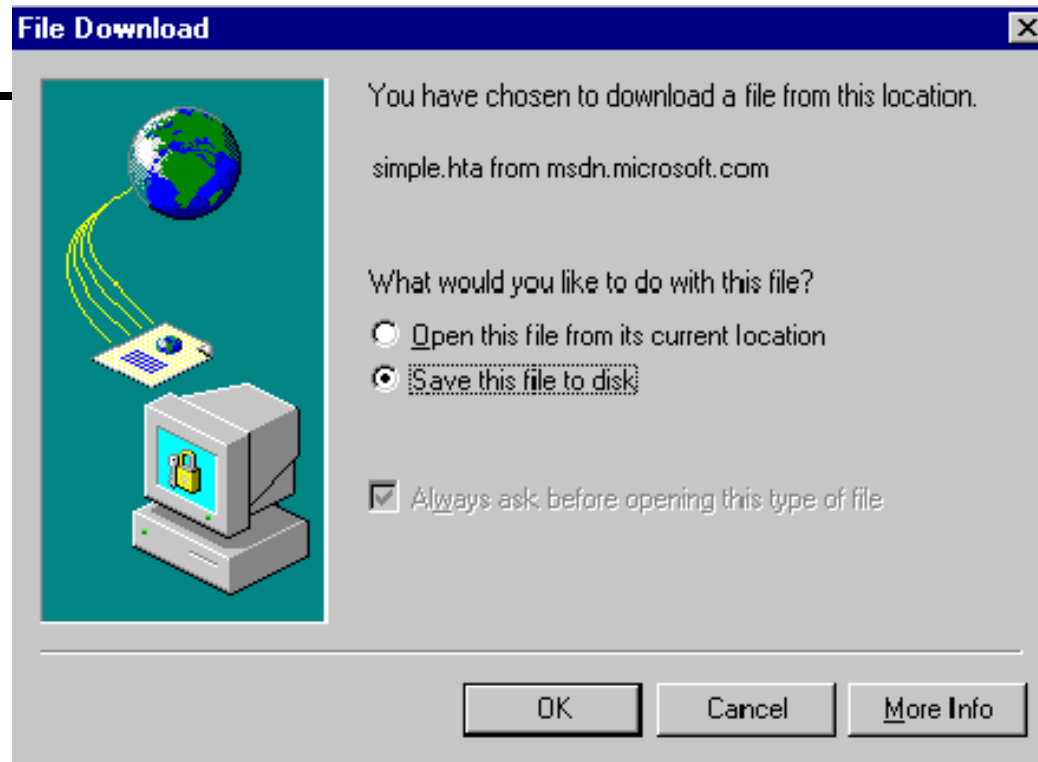
As fully trusted applications, HTAs carry out actions that Internet Explorer would never permit in a Web page. The result is an application that runs seamlessly, without interruption.

In HTAs, the restrictions against allowing script to manipulate the client machine are lifted. For example, all command codes are supported without scripting limitations. And HTAs have read/write access to the files and system registry on the client machine.

The trusted status of HTAs also extends to all operations subject to security zone options. In short, zone security is off. Consequently, HTAs run embedded Microsoft® ActiveX® controls and Java applets irrespective of the zone security setting on the client machine. No warning displays before such objects are downloaded and run within an HTA.

HTA windows can extend the trust relationship to content in other domains.

HTAs allow cross-domain script access between window objects and cookies. ...




Click on a .HTA you see this from a remote site,
No warning if run locally

Needs MSHTA.EXE to run

Other methods

Wireless networks

 See other defcon talk on this subject, but includes the new world of 3G phone communications as well as 'conventional' wireless networks.

Peer to Peer networks such as KaZaA

What a worm can achieve

- ☐ Gain publicity
- ☐ Destruction or corruption of data
- ☐ Denial of Service
- ☐ Revealing data
- ☐ Fix a system
- ☐ Trojan a system
- ☐ Hide other activities under the distraction of the worm

Defences

- It is possible to develop defences against the spread of network worms, but it does require tactics not used in computer infections at the present time.
- The main problem is the factor of time, some proposals have called for a WHO (World Health Organisation) approach given conventional worms/viruses are modelled on epidemiology, the problem being the length of time to respond to a network worm infection requires an automated response.
- The tactics for fighting worm infections, if automated could learn from this approach.

Reducing the population at risk

- ❑ Mono-cultures are ideal for any infection, with a large population with the same vulnerability, worms find it easy to spread.
- ❑ Diversity is a natural defence, deploy non standard systems
 - ❑ **Disadvantage:** Can cause compatibility problems.

Reducing the population at risk

- ☐ The recent SARS virus has been limited by good health practice.
- ☐ If systems maintained good security, they would be less vulnerable to being infected.
- ☐ Most zero day exploits won't work on systems if the system administrators have taken basic precautions.
- ☐ Many standards available, especially from manufacturers, or other bodies.(example INSECS standard, <http://www.dnscon.org/standard.rtf>) [2]
 - ☐ **Disadvantage:** Requires effort to implement, and unless people see it as a priority, it won't be done

Reducing spread of infection

■ When reducing the spread of an infection, we limit its ability to infect a large population.

■ One suggested method is **bandwidth throttling**

■ Mathew Williamson, March 2003, [9]

■ Data rate limit for new machines talking to host, or host to new machines, slows down spread of worm infection, if worm of large size without impacting on normal machine usage.

■ **Disadvantages:** Will not hinder small worms such as Slammer, unless limit is placed on number of machines communications can be sent to at a time. ‘connection throttling’. Will not hinder worms using topological scanning.

Reducing spread of infection

☐ Can be done via **containment** of worms, a workable approach for email worms being held in mailboxes until scanned via a heuristic scanner for example.

☐ **Disadvantage:** In the world of flash worms, can we content scan all IP transmissions?

☐ We could **blacklist** communications from known infected hosts, if we could get a small manageable list.

☐ **Disadvantage:** Can we deploy any blocking ruleset to a large population, before the worm reaches them? Can we get a small list?

Treating infected machines

☐ If you know your system may have been compromised you can obtain software to both patch and disinfect your system of an infection.

☐ **Disadvantage:** Takes time, and often impossible due to the number of downloads occurring from antivirus sites / manufacturer patch sites during an outbreak.

Spreading the word

Time for worm to reach system < (Detection time for worm + Time to counter)

☐ If this equation isn't met, the worm can be beaten we need a method of identifying a worm, and reacting to it before it reaches the bulk of our computers

☐ One method often suggested is to set a worm after the original worm,

☐ **Disadvantages:** Will fail unless the 'counter worm' replicates faster than the original worm. It also is also illegal.

Multi node detection

- Development of a network of computers that cooperate to provide defence information.
- Each computer as part of the network deploys a form of personal firewall called a **protection node** with a remotely modifiable ruleset.
- If a suspicious communication is transmitted to the system, the protection node prevents it being passed to the O/S and instead transmits it to an **analysis node**.

Multi node detection (cont)

- The analysis node (probably from a commercial supplier) determines if its a possible threat. Passing a digitally signed communication, via a web of trust, instructions for protection nodes to block similar transmissions.
- If the communication is not to be blocked, the protection node allows the communication to enter the computer, but it is monitored. If the computer then starts transmitting similar traffic the protection node can block further transmissions, effectively neutering the worm, and notify the analysis node.

Multi node detection (cont)

- The Analysis node then can, if it agrees, notify all protection nodes to block identical communications both incoming, and outgoing. This means any currently infected nodes will be notified and neutered.
- Such a distribution network can, in models, achieve 100 thousand notifications inside two seconds.
 - **Disadvantage:** The effect of a false positive, could result in a denial of service on the network.

Questions?

References

- [1] "Curious Yellow: The First Coordinated Worm Design", Brandon Wiley, 2002
http://blanu.net/curious_yellow.html
- [2] "Worms what is possible" (inc proof of concept web worm), Jonathan Wignall, August 2001, HAL2001
- [3] "How to Own the Internet in Your Spare Time", Stuart Staniford, Vern Paxson, Nicholas Weaver, 11th Usenix security Symposium, 2002
<http://www.icir.org/vern/papers/cdc-usenix-sec02/index.html>
- [4] "Warhol Worms: The Potential for Very Fast Internet Plagues", Nicholas Weaver, 2002,
<http://www.cs.berkeley.edu/~nweaver/warhol.html>
- [5] "Methodology of Computer Anti-Virus Research", Doctorial Thesis, Vesselin Bontchev, University of Hamburg, 1998
- [6] "An Environment for Controlled Worm Replication and Analysis"
Ian Whalley etal. IBM Research Centre 2000
www.research.ibm.com/antivirus/SciPapers/VB2000INW.pdf
- [7] Adapted for worms, from an equation published in "Time Based Security", Winn Schwartau, Impact press, 1999
- [8] BGP Impact of SQL Worm, 1/25/2003, Tim Griffin
http://www.research.att.com/~griffin/bgp_monitor/sql_worm.html
- [9] "Virus Throttling" Mathew Williamson, Virus Bulletin March 2003