An Effective Architecture and Algorithm for Detecting Worms with various Scan Techniques
Jiang Wu, Sarma Vangala, Lixin Gao & Kevin Kwith

Presented by: Namratha Hundigopal
CIS 859 - Advanced Topics in Network Security

Purpose
- Analyze various scan techniques ➔ Comparing spreading speed
  - Selective Random Scan
  - Routable Scan
  - Divide and Conquer Scan
  - Extreme scan
    - DNS Scan
    - Complete Scan
- Propose a generic worm detection algorithm
- Propose and evaluate ‘Victim Number Based Algorithm’

Worms
- mobile code
- autonomous
- spreads thru scan/exploit/transfer cycle

DANGERS
- fast spread
- install malicious code
- destroy/ corrupt valuable data
- access confidential information
- scan traffic takes lot of network resource

Related Work
- Worm propagation Models
  - Kephart and White designed an epidemiological model to measure virus population dynamics
  - Zou et. al. designed a two-factor worm model
  - Chen et. al. designed a concise discrete time model. The model can adapt parameters such as scan rate, patch rate and victim death rate.

Related Work
- Detection and Defense
  - LaBrea tool designed by Liston holds TCP sessions with victims for a long time
  - Somayaji and Forrest’s solution uses system call sequence database and compares new sequences with them
  - Williamson’s ‘virus throttle’ checks on the rate of SYN packet to new address
  - Cheung’s proposed scan activity graph inferred from the traffic

Scan Methods
- Selective Random Scan (SRS)
  - Is scanning the whole IP address space at random a good idea? NO
  - a set of address that more likely belong to existing machines are selected
    - Slapper worm picks 162 /8 networks
  - unallocated/reserved address blocks should not be selected
- must carry info about selected address
- increase in code size ➔ slows down spreading/infection time
Scan Methods

- **Routable Scan (RS)**
  - reduced address space + knowledge about address that are routable
  - unassigned IP address are no more included
  - same issues as selective random scan

- **Design and analysis**
  - globally routable prefixes can be got from BGP routing tables – so can we use BGP table as the database?
  - how many of addresses needs to be scanned – \( |\Omega| \)
  - what will be the minimum size of the database – 3K

Scan Methods

- **Divide-Conquer Scan (DCS)**
  - infected host divides its address database among its victims
  - code size of the worm is reduced

- scan traffic generated by the victims is reduced → hard to detect

- what if a infected machine crashes/turned off database that was passed to it is lost – possibility of single point failure

Scan Methods

- **Hybrid Scan (HS)**
  - a combination of simple scan methods
  - routable scan + random scan (at later stage)
  - divide-conquer + hitlist
  - why – specific address database may miss many vulnerable machines

- can infect more machines even if the propagation has been detected

Scan Methods

- **Complete Scan (CS)**
  - attacker uses only assigned IP address in target address database → hard to distinguish

- large address list needs to be carried → propagation delay

- distinct host address is > 100 million – list size 400M bytes

- by choosing a specific vulnerability the list size can be reduced

Extreme Scan Methods

- **DSN Scan (DNSS)**
  - IP address in DSN server are used to build target database → most of the address are in use

- hard to get complete list of address that have DSN records

- restricted to machines having public domain name

- worms should carry large database → propagation speed is reduced

Extreme Scan Method

- **Complete Scan (CS)**
  - attacker uses only assigned IP address in target address database → hard to distinguish

- large address list needs to be carried → propagation delay

- distinct host address is > 100 million – list size 400M bytes

- by choosing a specific vulnerability the list size can be reduced

Propagation speed analysis

- But what is the effect of address space selection on the worm spread?
- Worm propagation model used → AAWP model

\[
N_{i+1} = N_i + \left| N - N_i \right| \left( 1 - \frac{1}{T} \right)^{\alpha i}
\]

- \( N \) = total no of vulnerable machines
- \( T \) = address space used
- \( \alpha \) = no of machines infected up to time tick \( i \)
- \( s \) = scan rate
Propagation speed for SRS

- $N = 50,000$
- $S = 2$ (scans/sec)
- $t = 1$ (sec)
- $T = 2.7 \times 10^8$

Increase in propagation speed by using a target address pool.

Propagation speed for RS

- $N = 50,000$
- $S = 2$ (scans/sec)
- $t = 1$ (sec)
- $T = 10^8$

Code-Red like worm infects in 7 hrs.

Routable scan is clearly better than Random scan.

Propagation speed for DCS

- $N = 50,000$
- $S = 2$ (scans/sec)
- $t = 1$ (sec)
- $T = 10^8$

Faster than Routable scan.

Propagation speed for DNSS

- $N = 50,000$
- $S = 2$ (scans/sec)
- $t = proportional$ to code size due to transmission time
- $T = 10^8$

No spread seen due to huge address database that has to be carried.

Propagation speed for CS

- $N = 50,000$
- $S = 2$ (scans/sec)
- $t = proportional$ to code size due to transmission time
- $T = 10^8$

code size > 6M CS is worse than RS.

Comparison

- How do scan methods differ? - the way in which they select target addresses
- larger the database → delay in spread

Routable scan + Random scan.
Generic Worm Detection Architecture

- Detection Components (DC): monitors network behavior at different places (can be monitors or traffic analyzers)
- Control Center (CC): can determine the presence of worm attacks – How?
  by gathering info from DC
- Detection Network (DN): consists of address monitored by DC.

Victim Number Based Algorithm (VNBA)

- Detection System tries to track down the number of victims – But what are victims?
- Host/addresses from which a packet is sent to an inactive address – But how will this help?
- During worm attacks it is possible that large num of distinct hosts scan ports on inactive addresses
- How to determine if a host is a victim or not? – Victim Decision Rules
- OSDR – at least one scan packet is received by an inactive host
- TSDR – two scans

Victim Number Based Algorithm (VNBA)

- with TSDR, the num of victims detected by the system
  \[ V_k = \sum_{i=1}^{k} (m_i - m_{i-1}) \left[ 1 - \rho^{i-1} \right] \left[ 1 - \rho^{i-2} (\rho^{i-2} - \beta^{i-1}) \right] \]
  \[ m_i = \text{num of victims up to time tick } i \text{ and } p = (1 - D/T) \]
- assumed that each new victim uses the same target address space
- for DCS if victims are randomly distributed around the address space used by the attacker and \( s \geq 1 \)
  \[ V_k = \frac{D^{s-1}}{s^s} \]

If there is a sudden in num of distinct victims during monitoring of inactive address evidence of worm attack.

VNBA consists of 3 steps,
- gather all packets with inactive destination address
- retrieve the victims
- use a threshold to determine if there is sudden increase in num of victims – how will this help?
- If the rate of increase of victims is greater than threshold abnormality present
- continuous abnormalities worm attack

Validation of VNBA

When attack rate is more serious, detection time will be shorter

Validation of VNBA

Performance of VNBA for various sizes of DN

Increase rate of distinct sources

Validation of VNBA with internet traffic traces (N = 500k, 100k & /16-network)
Detection Performance

Future work

- To develop an integrated approach to further improve the proposed technique and develop an efficient mechanism to fight worm attacks

Discussion /Comments /Questions

- Any suggestions on improvements
- Any ideas on how to defend the attack
- Any comments on the paper

Thank You