SuperviZ

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

How Fast Does Malware Leveraging EternalBlue Propagate? The case of WannaCry and NotPetya

SOTERN TEAM - IRISA, IMT ATLANTIQUE

- Do Duc Anh NGUYEN (Presenter) - do-duc-anh.nguyen@imt-atlantique.fr
- Pierre ALAIN - pierre.alain@irisa.fr
- Fabien AUTREL - fabien.autrel@imt-atlantique.fr
- Ahmed BOUABDALLAH - ahmed.bouabdallah@imt-atlantique.fr
- Jérôme FRANÇOIS - jerome.francois@uni.lu
- Guillaume DOYEN - guillaume.doyen@imt-atlantique.fr

SecSoft Workshop 2024, Saint-Louis USA, June 28, 2024

Outline

Outline

Context

Malware is a contraction of *mal*icious and soft*ware* aims to damages to information systems
$\rightarrow$ Especially dangerous when the worm ability is enabled to spread

EternalBlue exploit [1]

Allows attackers to execute a remote code on the infected hosts by sending specially crafted Server Message Block version 1 (SMBv1) packets to unpatched Windows systems

WannaCry and NotPetya are two malware example that leverage EternalBlue to install a backdoor to deliver their payload
$\rightarrow$ Demand ransom after encrypting victims' data

[1] Z Liu et al. "Working mechanism of eternalblue and its application in ransomworm", International Symposium on Cyberspace Safety and Security, 2022

Background: WannaCry and NotPetya

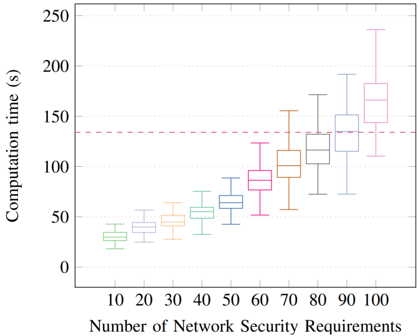
Before exploitation, they send SMBv1 packets to a target to check existence

- Vulnerability: based on response of target to an invalid request
- Infection: based on value of a field in responses
  $\rightarrow$ The backdoor modifies value the field

Characteristics	WannaCry	NotPetya
SMB field	Multiplex ID = 0x81	Reserved = 0x11
Propagation method	EternalBlue	EternalBlue EternalRomance Collected credentials

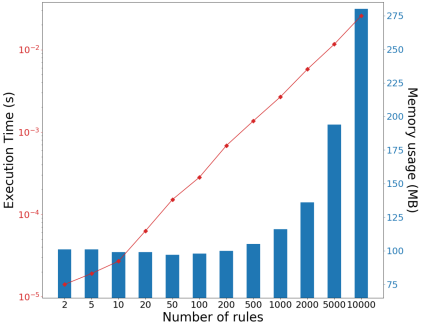
Problematic

Many malware detection and mitigation methods [2] but mostly local-only decisions



VEREFOO firewall rule computation [3]

$\rightarrow$ Long reaction time



Conflict detection in I2NSF [4]

$\rightarrow$ Follow polynomial complexity

To propose effective solution, understanding of malware propagation strategies is important

$\Rightarrow$ Current analysis do not cover knowledge of propagation behavior at scale

Contribution

Our contribution

- Conduct experiments on a 50-host network $\rightarrow$ study WannaCry and NotPetya propagation behavior
- Measurement of propagation speed
- Discussion on their propagation strategies

$\rightarrow$ Providing meaningful insights on malware propagation in a local network

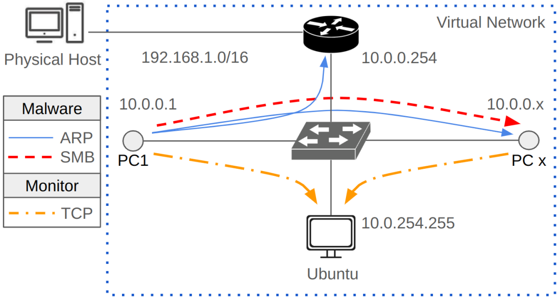
Outline

1. Introduction
   - Context
   - Background: WannaCry and NotPetya
2. Experiments
   - Environment Setup and Measurement Method
   - Observation on the NotPetya Sample
   - Observation on the WannaCry Sample
3. Conclusion and Future Works

Environment Setup

Environment setup

- GNS3 provides network simulation
- 50 Windows 7 hosts (2GB of RAM and 1 vCPU) start from 10.0.0.1 to 10.0.0.50
- The PC1 contains malware binaries
- An Ubuntu machine counts the number of infection
- Two samples are selected



→ A star topology can maximize the propagation speed

For accurate speed measurement, place a monitor at each host

Measurement Method

Two processes run on startup

- Check availability of hosts' IP address
  - if they are ready for connection
  - if they reboot
- Detect a malware process

Different detection is used

- `mssecsvc` is the first process of the WannaCry sample
  $\rightarrow$ Monitor running process name
- The normal process `rundll32.exe` is used to run the NotPetya sample
  $\rightarrow$ Monitor a full command: `rundll32.exe c:\Windows\notpetyafilename.dll,#1`

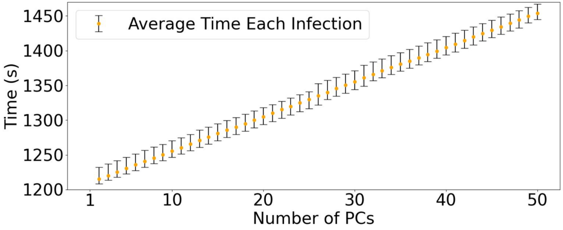
Assumption: The starting time t = 0 when the PC1 first reports

Each experiment is repeated 10 times

$\rightarrow$ More results from the WannaCry experiments are exposed

Average time for each host infected by NotPetya

Total average time to infect 50 hosts



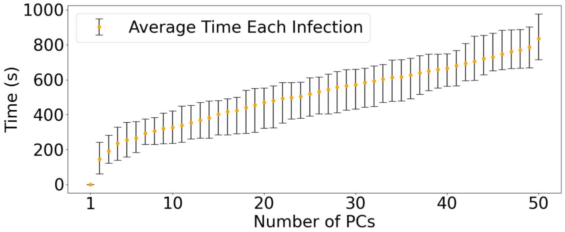
Total average time: 1454.08 $\pm$ 6.31 s (95% CI)
CI: Confidence Interval

Observation

- Strategy: Complete scan then explore
- Sequential scan: prefix /24 ($\sim$3 s / IP address) + 5 min delay
  $\rightarrow$ After 1200 s, the 2nd host is infected
- Then, $\sim$4.97 s / host $\rightarrow$ Increases linearly
- The order of infected hosts: 10.0.0.1 to 10.0.0.50
  $\rightarrow$ Follows the order of scanning
- The *PC*1 infected 49 hosts (no competition)

Average time for each host infected by WannaCry

Total average time to infect 50 hosts



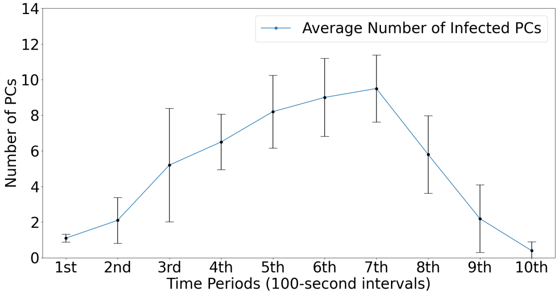
Total average time: 836.11 $\pm$ 62.48 s (95% CI)

Observation

- Strategy: Scan while exploring
- Sequential scan: prefix /24
  $\rightarrow$ After 60-250s s, the 2nd host is infected
  $\rightarrow$ $>$ The infection time if only a 2-host network is considered ($\sim$50 s)
- Then epidemic spread, but not exponential increase
  $\rightarrow$ Perhaps 50 hosts are not enough
- The order of infected hosts: Random
- Some hosts reboot due to `srvnet.sys`

Number of Infected Hosts in Each 100-Second Period

How did the speed change?

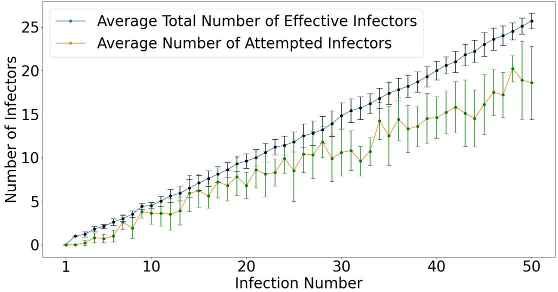


Observation

- Speed increases in the first 700 seconds
- Speed slows down in the 8th period
  $\rightarrow$ The number of infectors and remaining hosts affect the speed

Number of Attempted and Effective Infectors

How infectors compete with each other?



Definition

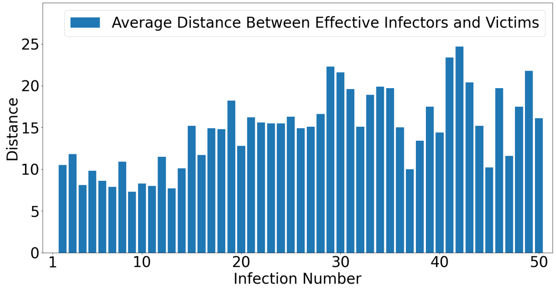
- Effective infector: The first one sending malware binary
- Attempted infector: Executed incomplete exploitation

Observation

- $mean_{attempted\_infectors} = 18$
- $mean_{sum\_effective\_infectors} = 25$
  $\rightarrow$ At least one host is infected by a new infector

Average Distance of IP Addresses Between Infectors and Victims

How the propagation strategy affects the infection order?



Example: distance 10.0.0.1 and 10.0.0.9 = 8

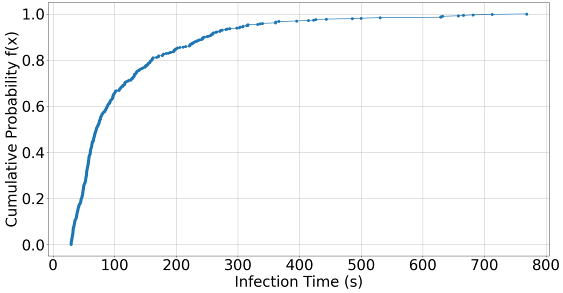
Observation

Due to sequential scan, most effective infectors have low IP

- 2nd-14th infection: Closer victim is infected
  $\rightarrow$ New infectors are more effective than older one
- After 14th infection: Father victims is infected
  $\rightarrow$ Old infectors become effective in infecting higher IP victim

Infection Time of Effective Infectors

What is the time needed to protect a system?



We compute

- The time to infect $t2i(i) = t_{victim} - t_{infector\_i}$
- Empirical CDF of $t2i$ for 490 infections in 10 repetitions

Observation: $\sim$ 20% of infections $\leq$ 50 s
$\rightarrow$ Save $\sim$ 80% of hosts if the reaction time $\leq$ 50 seconds

CDF: Cumulative distribution function

Discussion

WannaCry propagate faster than the NotPetya

Characteristics	WannaCry	NotPetya
Scanning behavior	Sequential	Sequential
Propagation strategies	Parallel to the scan	After the scan
Competitors	✓	×

The results imply that

- Mitigating malware propagation is challenging
- Propagation speed of the WannaCry increase non-linearly
  $\rightarrow$ The need for reducing reaction time

$\Rightarrow$ Determine the time interval to deploy appropriate mitigation

Outline

Conclusion and Future Works

A dynamic analysis of WannaCry and NotPetya is presented to

- Understand their propagation behavior
- Measure their propagation speed

The results present

- Challenges for detection and mitigation solutions
- Large confidence intervals
  $\rightarrow$ A dynamic propagation strategy that may vary

Future works

- 1000 hosts and $>$ 10 repetitions
- Propose a fast mitigation approach
  - Leverages microservices and Intent-Based Networking (IBN) systems
  - Opportunistic approach that synchronizes microservices' behavior to autonomously react

Question

Thank you for listening. Any question?

References I

[1] Zian Liu et al. „Working mechanism of eternalblue and its application in ransomworm". In: International Symposium on Cyberspace Safety and Security. Springer. 2022, pp. 178–191.

[2] S Sibi Chakkaravarthy et al. „A survey on malware analysis and mitigation techniques". In: Computer Science Review 32 (2019), pp. 1–23.

[3] Daniele Bringhenti et al. „Automated Firewall Configuration in Virtual Networks". In: IEEE Transactions on Dependable and Secure Computing 20.2 (2023), pp. 1559–1576. DOI: 10.1109/TDSC.2022.3160293.

[4] Do Duc Anh Nguyen et al. „A Robust Approach for the Detection and Prevention of Conflicts in I2NSF Security Policies". In: NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium. 2023, pp. 1–7. DOI: 10.1109/NOMS56928.2023.10154304.

EternalBlue Exploitation

Unpatched Windows versions from XP to 8.1 are vulnerable

- Allows SMB connection without authentication
- Wrongly compute the heap allocation size of SMB requests
- Constant memory addresses used by the (Hardware Abstraction Layer) HAL module, has execution privilege

References

- – How Fast Does Malware Leveraging EternalBlue Propagate? The case of WannaCry and NotPetya