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

Do Duc Anh Nguyen*, Pierre Alain‡, Fabien Autrel*, Ahmed Bouabdallah*,
Jérôme François† and Guillaume Doyen*
*SOTERN - IRISA (UMR CNRS 6074), IMT Atlantique, firstname.lastname@imt-atlantique.fr
†SnT, University of Luxembourg and Inria Nancy Grand Est, jerome.francois@uni.lu
‡SOTERN - IRISA (UMR CNRS 6074), Université de Rennes, pierre.alain@irisa.fr

***Abstract*—Malware attacks pose a critical threat to digital infrastructures particularly given their potential for widespread and fast propagation. Mitigating them involves limiting their expansion, which requires a thorough understanding of their propagation mechanisms. However, few studies have been conducted on their propagation behaviors in large-scale networks. In this paper, we present the results of an empirical study focusing on the propagation strategy of WannaCry and NotPetya, two malware instances leveraging EternalBlue, an exploit developed by the NSA and stolen by The Shadow Brokers hacker group, which has been used to implement rapid spreading in some malware instances. Our experiments qualify the speed of infection, epidemic behavior, and spreading strategies in a local network of 50 VMs. We have especially measured for WannyCry that (1) nearly 20% of infections are processed in less than 50 seconds, and (2) up to 16 hosts are infected in a 100-second period. Our results provide meaningful insights on malware propagation to support the design of effective countermeasures.**

***Index Terms*—Malware infection, Propagation characteristics, EternalBlue, WannaCry, NotPetya**

## I. Introduction

Malware, a contraction of *mal*icious and soft*ware*, is a class of program specifically designed to cause various types of damages to information systems, such as altering data, programs, hardware, leaking information, etc. Digital architectures are susceptible to a wide range of malware attacks (5.5 billion reported in 2022[1]). Malware can take many forms (viruses, worms, rootkits, etc.). but they become especially dangerous when the *worm* ability is enabled to spread across entire networks. EternalBlue is a particular dangerous exploit that can be leveraged for very fast propagation, as it only requires sending packets to execute remote code on victim machines by contrast with manual propagation achieved through infected attachments of emails, for instance. The EternalBlue exploit is present in the Server Message Block (SMB) protocol in Windows, the most widely used operating system, and WannaCry and NotPetya are two malware instances that leverage EternalBlue to spread quickly across entire networks, causing, for example, a huge attack on data encryption in 2017[2].

Many methods for detecting and mitigating malware effects have been proposed to date [1]. However, these methods largely involve local-only procedures or decisions (*e.g.,* by modifying firewall rules), and they exhibit a potentially long reaction time. Consequently, it appears essential to propose novel concerted responses at the network level that fit with the distributed and fast propagation strategies of malware to limit their diffusion. To support this objective, a thorough understanding of network-level malware propagation mechanisms is required. But, despite extensive analysis and documentation of the EternalBlue exploit and the related malware instances, there is insufficient knowledge of their propagation behavior at scale.

In this paper, we perform an experimental measurement campaign on a 50-host network to study the propagation behavior of two malware samples, WannaCry and NotPetya, leveraging the EternalBlue exploit. Besides measuring the propagation speed of the two malware instances, the differences in the propagation strategies of these two samples are discussed and confronted with reports from related work. We identified that the WannaCry sample used a more complex propagation strategy than the NotPetya sample. For this reason, more results from the WannaCry experiments are exposed, including the increase in the number of infectors (*e.g.,* the infected hosts that try to propagate the malware) and the IP distance between infectors and victims. As such, the results of this study are not a new propagation model but aim at providing meaningful insights on malware propagation in a local network and thus can support the design of effective countermeasures.

The rest of the paper is structured as follows: The background, the malware propagation models, and the analysis efforts of EternalBlue, WannaCry, and NotPetya are discussed in Section II. Section III presents the experimental scenarios, setup, and propagation speed results of the WannaCry and NotPetya samples. Finally, Section IV summarizes the obtained observations on the experiments and future works.

## II. Background and Related Work

In this section, we provide some background about the EternalBlue exploit, focus on the propagation behavior of WannaCry and NotPetya, and present related works on malware propagation models and analysis reports on WannaCry and NotPetya samples.

[1]Source: 2023 SonicWall Cyber Threat Report.https://www.sonicwall.com/medialibrary/en/white-paper/2023-cyber-threat-report.pdf

[2]Source: Cyberattack Hits Ukraine Then Spreads Internationally,The New York Times, 2017

### A. Background

*1) EternalBlue:* The EternalBlue exploit [2] affects the Windows SMB protocol [3] commonly used for remote file and printer sharing. Unpatched Windows versions from XP to 8.1 [4] are vulnerable due to several reasons: (1) SMB version 1 (SMBv1) allows unauthenticated attackers to send specially crafted SMB packets to victims; (2) The allocation size in the heap memory for SMB data requests is miscalculated, which allows attackers to assign more data than the requested allocation size, which results in a buffer overflow; (3) A system module in Windows, namely the Hardware Abstraction Layer (HAL) [5], uses a heap with a fixed address (*e.g.,* 0xffd00000 in Windows 32-bit and 0xffffffffffd00000 in Windows 64-bit). Taking advantage of these security issues, attackers can send crafted SMB packets to trigger a buffer overflow to write binary code and make it executable on the heap of the HAL module. In the case of WannaCry and NotPetya, this binary code is the DoublePulsar backdoor. The DoublePulsar backdoor then waits for the connection from the attackers to execute their commands.

*2) WannaCry:* In order to propagate, WannaCry relies on ARP scanning. It runs a thread for each scanned remote host. When a victim is discovered, it first establishes a TCP connection to port 445 of the victim. Then, it checks if the victim is vulnerable to the EternalBlue exploit by requesting a non-existent named pile using the SMB protocol. The response from a potential victim is `STATUS_INSUFF_SERVER_RESOURCES` if not patched, `STATUS_ACCESS_DENIED` or `STATUS_INVALID_HANDLE` otherwise. The WannaCry instance then checks whether the victim has already been infected by determining the presence of the DoublePulsar backdoor. To do this, WannaCry requests an SMB session, stealthily embedding its command in the `Timeout` field of a SMB session request [6]. If the backdoor does not exist, the value of the `Multiplex ID` field in the SMB session response will be the same as that of the request. Otherwise, the `Multiplex ID` field value is 0x81. Finally, the EternalBlue exploit is used by WannaCry to copy and execute itself on the victim.

*3) NotPetya:* This malware checks the presence of its backdoor using a different field compared to WannaCry. If a victim has already been infected, the backdoor responds to the SMB session request with the `Reserved` field set to 0x11. Otherwise, the `Reserved` field is 0x0. Then, NotPetya attempts to log in to other victims via SMB using credentials harvested from its infected victim. It is also capable of exploiting EternalBlue and EternalRomance (another exploit in the SMBv1 protocol) [7]. The processing of these two exploits depends on the version of Windows shown in victims' SMB responses [8] (*e.g.,* on Windows 7, only the EternalBlue exploit is used).

### B. Related Work

Since self-propagating malware became a major threat in the 2000s, different studies were conducted to characterize their propagation processes. Many of them rely on epidemiological models inspired from medical research. The most classic model is the SI model, which assumes two states for each host: Susceptible (S) or Infected (I). More advanced models consider the possibility for a host to recover and to be protected from further infection (SI Removed). However, such models lack accuracy even when applied to the Code Red worm [9] from 2001, leading to consideration of different factors such as human decisions or network congestion caused by the worm itself [10]. This paves the way to a large literature leveraging different types of models (continuous-time, discrete-time, stochastic, spatial, etc.) in the attempt to catch the complexity of the propagation and infection mechanisms [11]. In addition, such models need to be adapted to the context, taking into account, for example, the propagation mechanisms, such as emails in [12], or the type of targets, such as IoT in [13]. This demonstrates the difficulty in having accurate models. It is worth noticing that most of these works focus on approximating the infection rate in large networks such as the Internet, while a few of them are dedicated to local environments [14]. As highlighted in [11], [14], network topologies and organizational aspects are major factors in models. Obviously, differences in malware functionalities and implementations, including different techniques, strategies, and actions, require studying them in the wild to precisely understand their behavior.

In [15], static and dynamic malware analyses are conducted on a WannaCry sample to try extracting encryption functions along with some network traffic between the host and the attackers' server. However, only the resource loader (*i.e.,* an encryption component) is executed, excluding the ability to observe the propagation behavior that uses the EternalBlue exploit. In contrast, the authors in [6] analyze a complete sample of WannaCry, including the resource loader and the dropper (*i.e.,* responsible for the resource loader extraction and propagation), but in a 2-host network solely. In [16], beside attempting to extract the digital forensic of WannaCry, the authors monitor the number of SMB packets per second generated by the malware in a 3-host network. Upon analyzing both the dropper and the resource loader, the authors in [17] provide a comprehensive analysis by identifying different phases of the WannaCry malware life cycle: Initial interaction, persistence, configuration load, encryption, recovery prevention, and propagation in a 2-host network.

NotPetya emerged after WannaCry. Two related technical analyses provided in [18] and [8] describe its main characteristics (encryption and propagation). In addition, LogRhythm Labs performed an analysis [7] on two different samples of NotPetya.

According to the aforementioned analyses of WannaCry and NotPetya, multiple binary samples can be found, probably sharing some common code. In our research, we assess their propagation capabilities and speed in a local environment composed of tens of machines. This is also the objective of a very recent similar work [19]. However, the latter relies on an intrusion detection system to identify a host as newly

infected once it starts scanning the network. Our method is more accurate, consisting of monitoring the EternalBlue exploit prior to further malicious activities. Therefore, even if an infected host does not perform scanning, our method can still identify it. This might explain why authors in [19] observed that every vulnerable host has not been infected.

## III. Malware Experiments

In this section, we first present our experimental setup, then provide our insights regarding NotPetya, and finally WannaCry. We finish this section with a discussion related to the two propagation strategies.

### A. Malware Environment Setup

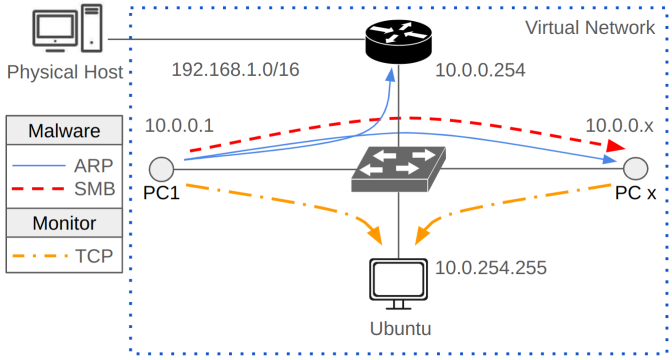


Fig. 1: Environment Setup and Monitor

To construct a large network, the GNS3[3] virtual environment has been selected due to its programmatic and visual configuration methods. It also facilitates the storage of the network's initial state (*e.g.,* prior to malware infection) for later recovery and multi-evaluation purposes. The network, shown in Figure 1, consists of 50 Windows 7 hosts (2GB of RAM and 1 vCPU) connected via an Ethernet switch within the 10.0.0.0/24 local network. Additionally, a router is connected to the switch and a physical server. One host, $PC1$, is assigned the fixed IP address 10.0.0.1 and contains the malware samples. A DHCP service running on the router automatically assigns IP addresses (from 10.0.0.2 to 10.0.0.50 with a subnet mask of 255.255.255.0) to the remaining 49 hosts. At the network level, it is expected to see ARP and SMB packets sent by WannaCry and NotPetya to probe the network and propagate to other hosts, respectively. In terms of malware sample selection, the two samples of WannaCry and NotPetya referred to in Table I have been used as droppers, as they have been analyzed by the related work presented in the previous section.

### B. Measurement Method

As a common approach to monitoring propagation behavior, network traces can be collected to know the relationships between infectors and victims. However, to have accurate temporal information about when a host is infected, we decided to monitor the hosts themselves.

[3]Source: GNS3 Documentation https://docs.gns3.com/docs/

TABLE I: Samples of WannaCry and NotPetya

| Sample | SHA256 |
|---|---|
| WannaCry | 24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c |
| NotPetya | 027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d3a745 |

To collect this information, each Windows host runs a program at startup to detect the presence of a malware process. For WannaCry, the first process that appears is `mssecsvc`. However, NotPetya runs its binary through the normal process `rundll32.exe`, which allows to run Dynamic Linked Library (DLL) files. To overcome this problem, the full command of the `rundll32` process is checked instead. A host is infected if the command `rundll32.exe c:\Windows \[The name of the sample file],#1` is observed. Actually, in our experiment, this command is `rundll32.exe c:\Windows \notpetyafile.dll,#1`.

Once the monitoring tool detects the presence of the malware, it reports the infected host to a dedicated Linux machine with the IP address 10.0.254.255 (shown in Figure 1). As the malware instances only target Windows machines, this Linux machine acts as an observer and not as a potential victim or infector. In addition, because Windows hosts require some time to be assigned an IP address by the DHCP service, another program is run on each host at startup to inform the Ubuntu machine of its connection status if its IP address is available. In this way, we can also observe hosts that reboot accidentally during the malware attack. To obtain relative time measurements, we assume the starting time $t = 0$ when $PC1$ first reports to the observer machine.

This measurement approach, which combines host-based and network-based monitoring, allows for the differentiation of effective and attempted infections as well as delays between scanning and malware installation phases. Furthermore, each experiment is repeated 10 times to obtain meaningful statistical values for the derived metrics (*e.g.,* the number of infected hosts).

### C. Observation on the NotPetya Sample

Figure 2 depicts the average time for each host to be infected by NotPetya. Once the second host is infected (which takes more than 1200 seconds after $PC1$ is infected), the total number of effective victims increases linearly. On average, $1454.08 \pm 6.31$ seconds (95% confidence interval) are required to compromise all 50 hosts. The order of hosts infected over time follows the order of the network scanning phase (*i.e.,* sending ARP packets within the range of 10.0.0.0/24, as the subnet mask configured in $PC1$, involves sequentially incrementing the fourth octet of the IP address).

Besides, all infections are exclusively performed from $PC1$ with no competition from other hosts (*i.e.,* there is no second

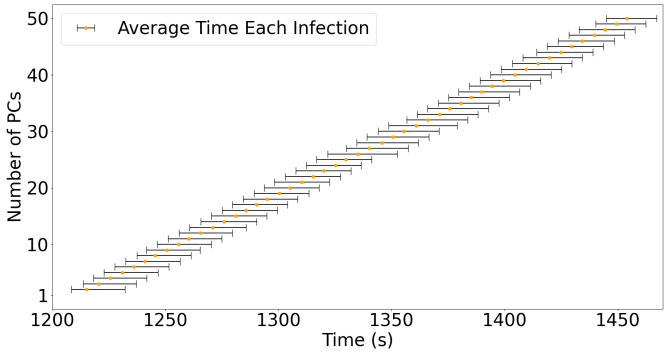


Fig. 2: Average time for each host infected by NotPetya

infector that uses the EternalBlue exploit). Actually, NotPetya performs the EternalBlue exploit once it has scanned the /24 network (taking 3 seconds on average for each IP address) and after a waiting time of about 5 minutes. In contrast, the infection caused by the NotPetya sample is quite fast (approximately 4.97 seconds per host). Hence, such a process leaves enough time for the $PC1$ to compromise all other hosts before any of them can use the EternalBlue exploit. Hence, no epidemic-like spread can be observed. It should also be noticed that during the scan, the NotPetya sample unsuccessfully attempted to bruteforce victims with the credentials gathered from the $PC1$.

### D. Observations on the WannaCry Sample

As a general observation, WannaCry also enumerates IP addresses sequentially during the scan, but unlike NotPetya, the order of infected hosts appears to be random. This is because each infection process runs on a single thread.

*1) Number of Infected Hosts Over Time:* As shown in Figure 3, the second host is infected after approximately 60-250 seconds. This is longer than the infection time if only a 2-host network is considered (approximately 50 seconds in that case).

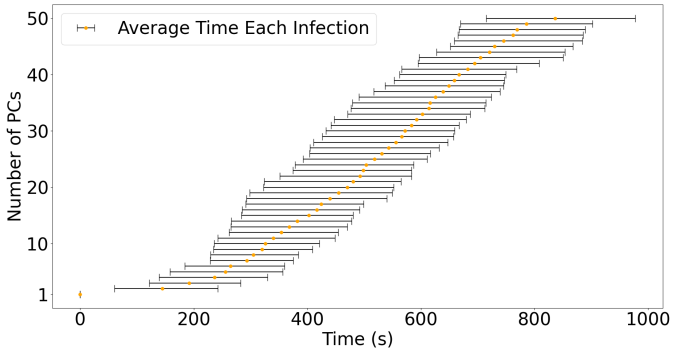


Fig. 3: Average time at which each host is infected by WannaCry

The reason for this observation is still unknown. When a reachable IP address is detected, WannaCry initiates a thread to establish a SMB connection to assess the vulnerability's existence. Therefore, it becomes apparent that in a 50-host network, multiple threads are generated for all 50 hosts, whereas in a 2-host network, only one thread is created.

In addition, WannaCry uses epidemic-like spread: once a host is infected, it starts infecting others by becoming an infector. As a result, the total propagation time of WannaCry is much shorter than that of NotPetya. The average total time for compromising all hosts is 836.11 $\pm$ 62.48 seconds (95% confidence interval). Overall, the number of infected hosts does not follow an exponential increase. It is perhaps due to the fact that 50 hosts in a network are not enough to see this acceleration.

Moreover, some victims crash during the experiment, leading to a blue screen with a message `DRIVER_IRQL_NOT_LESS_OR_EQUAL` and the stop code `0x000000D1`. Those hosts then reboot after receiving a number of SMB packets from infectors. The technical information section of the error message mentions the kernel-level driver `srvnet.sys`, which handles file sharing in SMB. However, they are still vulnerable and can be infected later.

*2) Number of Infected Hosts in Each 100-Second Period:* To measure the propagation speed, Figure 4 displays the average number of infected hosts in successive 100-second periods. The speed increases in the first 700 seconds (*i.e.,* from the 1st period to the 7th period). This can be explained by the increase in the number of new infectors. As this also leads to a reduction in the number of remaining hosts to infect, the speed starts to decrease in the 8th period. This decrease can also be explained by the aforementioned reboot time of crashed hosts.

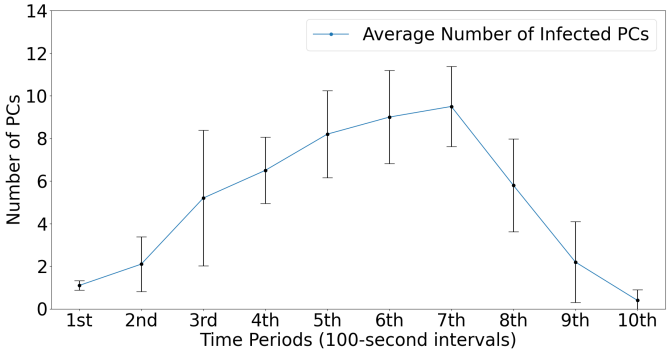


Fig. 4: Number of infected hosts per 100-second period

*3) Number of Attempted and Effective Infectors:* While increasing the number of infectors certainly boosts the propagation speed, they also become competitors, leading to infection attempts in parallel. Given a particular victim, only the infector receiving the SMB response from it (indicating the presence of the DoublePulsar backdoor) is considered as the effective infector, while others are qualified as unsuccessful attempts. Upon detection of the backdoor, the infector proceeds to upload the malware binary. Network traffic analysis reveals that some infectors successfully use the EternalBlue exploit

by activating the DoublePulsar backdoor on the victims, but effective infectors are those that deliver the malware binary. In other words, attempted infectors are those responsible for sending SMB packets related to the EternalBlue exploit, excluding vulnerability and backdoor checks, before the SMB response of their victims is sent to an effective infector.

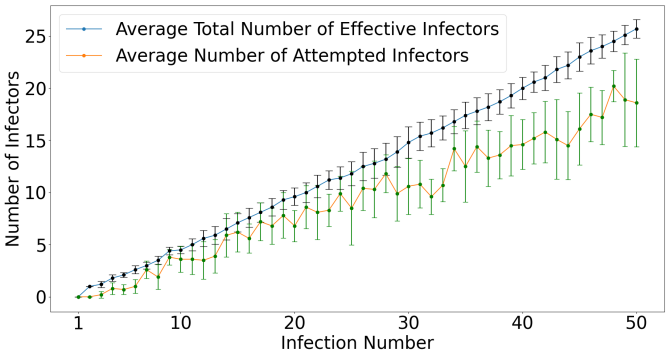


Fig. 5: Number of Attempted and Effective Infectors

The average total number of these effective infectors is computed for each infection number (representing 50 hosts on the network, corresponding to 50 infections) alongside the unsuccessful infectors (attempted infectors).

As depicted in Figure 5, both the average number of attempted infectors and the average total number of effective infectors increase. At the 50th infection (*i.e.,* 49 infectors), the average number of attempted infectors and the total number of effective infectors exceed 18 and 25 infectors, respectively. This means that at least one host is infected by each new infector in the propagation attempt.

*4) Average Distance of IP Addresses Between Infectors and Victims:* To understand the competition between infectors, Figure 6 highlights the average distance between the numerical representation of IP addresses of effective infectors and victims at each infection number. For example, if the IP address of an effective infector is 10.0.0.2 and that of its victim is 10.0.0.10, the distance is 8. Since the first infection number is represented as $PC1$, the distance is always 0. It can be observed that at the beginning of the infection phase (from 2 to 14), most victims are close to infectors in terms of IP distance. However, after the 14th infection, the distances between victims and infectors increase significantly.

Infectors typically initiate network probing from 10.0.0.1 to 10.0.0.254 (10.0.0.0/24). Consequently, most initial victims becoming infectors during the initial infection phase possess low IP addresses. For instance, the maximum IP address among the first fourteen infected hosts is 10.0.0.33 (even if the order of infected hosts is random due to independent infection threads). These low-IP infectors, not having initiated many infection threads yet compared to older infectors, compete with them by simply uploading the malware binary to low-IP victims after confirming the presence of backdoors established by the older infectors. Consequently, during the initial infection phase, the majority of IP address distances

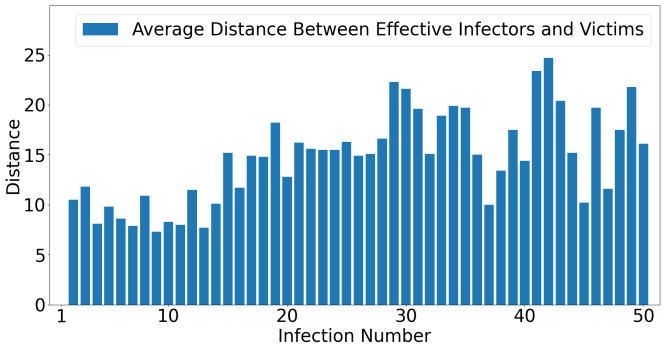


Fig. 6: Average Distance Between Effective Infectors and Victims

between effective infectors and victims predominantly involve low-IP infectors and victims. Conversely, the older infectors have had sufficient time to establish themselves as effective infectors targeting higher-IP victims.

*5) Infection Time of Effective Infectors:* In previous experiments, the propagation time has been measured from an overall perspective. To estimate the time needed to protect a system from any infected computer, we calculate the time to infect, $t2i(i)$, taken for each effective infector $i$ to infect another victim by calculating the difference between their own infection times.

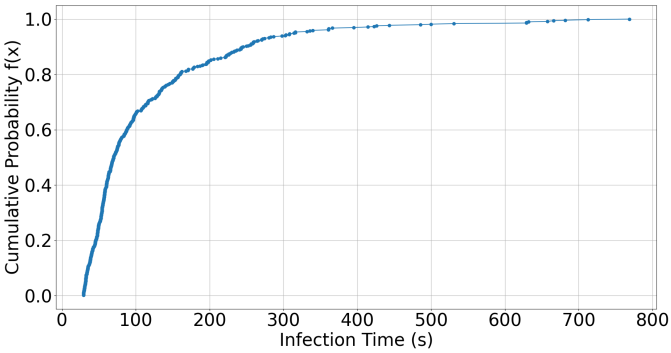


Fig. 7: Infection Time of Effective Infectors

Figure 7 depicts the empirical cumulative distribution function of $t2i$ for 490 reported infections in 10 repetitions (49 infections per repetition). As depicted, nearly 20% of infections are processed in less than 50 seconds. Hence, if a reaction mechanism is able to stop the propagation from an infected host in less than 50 seconds, nearly 80% of the other hosts could be proactively defended, but this also assumes that this time includes all necessary steps from the detection, including alerting, before being able to react appropriately.

### *E. Discussion*

Compared to NotPetya, WannaCry exhibits a different propagation strategy. Our WannaCry sample uses the EternalBlue exploit in parallel with the network scan once a new host is reachable, while the NotPetya sample waits until the network

scan ends before performing the exploit. As a result, $PC1$ in the NotPetya experiments is the only effective infector that infects all victims. For Wannacry, since infectors start the infection process immediately, the propagation speed is much faster (even though some infections caused by WannaCry infectors may take a bit longer as usual due to their competition). The result shown in Figure 4 also confirms that the propagation speed of the WannaCry sample accelerates non-linearly, hence increasing the need for reducing reaction time to mitigate its impact. Even under the ideal assumption that the WannaCry sample can be detected immediately after installation, reacting to it remains challenging.

Besides, the network scanning behaviors of the NotPetya and WannaCry samples significantly affect their propagation speed. In some other experiments on our 50-host network, DHCP was reconfigured to assign IP addresses with a subnet of 255.255.0.0 (*i.e.,* 10.0.0.0/16). The propagation time of the NotPetya sample increased as expected due to the increase in network scanning time. For WannaCry, the network is scanned from 10.0.0.1 to 10.0.254.255 (10.0.0.0/16) with a pattern of incrementing the third byte while keeping the first two bytes constant. For example, the scanning sequence is from 10.0.0.1 to 10.0.255.1, then from 10.0.0.2 to 10.0.255.2, and so on until 10.0.0.254 to 10.0.255.254 is reached (approximately 20-35 seconds for each subrange). Since the range of IP addresses assigned to hosts is only from 10.0.0.1 to 10.0.0.50, this results in a large propagation time (approximately 1613.67 seconds on average) to spread across all 50 hosts (even though scanning and exploit processes are still running in parallel).

In an attack event, networks providing critical services can continue to provide them in degraded mode up to a certain critical threshold. Our approach enables the determination of the time interval during which triggering the appropriate mitigation allows critical functions to continue operating safely.

## IV. Conclusion

A dynamic analysis of WannaCry and NotPetya is presented in this paper to understand their propagation behavior and measure their propagation speed. Experiments have been conducted on a network of 50 Windows VMs to study the behavior of two samples of WannaCry and NotPetya with 10 repetitions. In this 50-host network, the total propagation time of the NotPetya sample is stable compared to experiments in smaller networks since there is only one infector doing the propagation. In the case of WannaCry, the propagation speed of the second infection of the 50-host network is slightly slower than that of the 2-host networks. When multiple victims are targeted by WannaCry simultaneously, the performance of the malware sample may vary. The propagation speed measured from these two malware samples can be challenging for detection and mitigation solutions. Additionally, the experimental results show large confidence intervals, indicating a dynamic propagation strategy that may vary from one situation to another. Further experiments are planned to make the evaluation more realistic by increasing the number of hosts to 1000 and running more than 10 repetitions.

In the future, we aim to propose a fast mitigation approach dedicated to such malware attacks. This approach leverages microservices and Intent-Based Networking (IBN) systems to deploy countermeasures based on reaction intents (*i.e.,* the desired outcome without specifying concrete operations). An opportunistic approach will be proposed to share information between microservices and synchronize their behaviors to autonomously react to malware attacks to cope with the fast propagation time of malware.

## Acknowledgment

This work has been partially supported by the French National Research Agency under the France 2030 label (Superviz ANR-22-PECY-0008). The views reflected herein do not necessarily reflect the opinion of the French government.

## References

[1] S. S. Chakkaravarthy *et al.*, "A survey on malware analysis and mitigation techniques," *Computer Science Review*, vol. 32, pp. 1–23, 2019.
[2] Z. 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.
[3] (2016) Server message block overview. [Online]. Available: https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831795(v=ws.11)
[4] (2023) Microsoft security bulletin ms17-010 - critical. [Online]. Available: https://learn.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010
[5] (2021) Windows kernel-mode hal library. [Online]. Available: https://learn.microsoft.com/en-us/windows-hardware/drivers/kernel/windows-kernel-mode-hal-library
[6] K. Da-Yu, S.-C. Hsiao, and T. Raylin, "Analyzing wannacry ransomware considering the weapons and exploits," in *ICACT*. IEEE, 2019, pp. 1098–1107.
[7] LogRhythm Labs. (2017) Notpetya technical analysis.
[8] K. Sood and S. Hurley, "Notpetya technical analysis part ii: Further findings and potential for mbr recovery," *CrowdStrike Blog*, 2017.
[9] D. Moore, C. Shannon, and K. Claffy, "Code-red: a case study on the spread and victims of an internet worm," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, 2002, pp. 273–284.
[10] C. C. Zou, W. Gong, and D. Towsley, "Code red worm propagation modeling and analysis," in *ACM CCS*, 2002.
[11] Y. Wang, S. Wen, Y. Xiang, and W. Zhou, "Modeling the propagation of worms in networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 942–960, 2014.
[12] C. C. Zou *et al.*, "Modeling and simulation study of the propagation and defense of internet e-mail worms," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 2, pp. 105–118, 2007.
[13] A. Zhaikhan, M. A. Kishk, H. ElSawy, and M.-S. Alouini, "Safeguarding the iot from malware epidemics: A percolation theory approach," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 6039–6052, 2021.
[14] H. K. C. Hong Guo and K. Kelley, "Impact of network structure on malware propagation: A growth curve perspective," *Journal of Management Information Systems*, vol. 33, no. 1, pp. 296–325, 2016.
[15] J. Jones, "Ransomware analysis and defense-wannacry and the win32 environment," *IJISS*, vol. 6, no. 4, pp. 57–69, 2017.
[16] A. Zimba and M. Mulenga, "A dive into the deep: demystifying wannacry crypto ransomware network attacks via digital forensics," *IJITS*, vol. 10, no. 2, pp. 57–68, 2018.
[17] M. Akbanov, V. G. Vassilakis, and M. D. Logothetis, "Wannacry ransomware: Analysis of infection, persistence, recovery prevention and propagation mechanisms," *JTIT*, no. 1, pp. 113–124, 2019.
[18] K. Sood and S. Hurley, "Notpetya technical analysis–a triple threat: File encryption, mft encryption, credential theft," *CrowdStrike Blog*, 2017.
[19] A. Chernikova, N. Gozzi, N. Perra, S. Boboila, T. Eliassi-Rad, and A. Oprea, "Modeling self-propagating malware with epidemiological models," *Appl. Netw. Sci.*, vol. 8, no. 1, p. 52, 2023.