Hindawi
Complexity
Volume 2021, Article ID 5415724, 19 pages
https://doi.org/10.1155/2021/5415724

Research Article

MalSEIRS: Forecasting Malware Spread Based on Compartmental Models in Epidemiology

Isabella Martínez Martínez,[1] **Andrés Florián Quitián**,[1] **Daniel Díaz-López**,[1] **Pantaleone Nespoli**,[2] **and Félix Gómez Mármol**[2]

[1]*School of Engineering, Science and Technology, Universidad del Rosario, Carrera 6#12C-16, Bogotá, 111 711, Colombia*
[2]*Faculty of Computer Science, University of Murcia, Edificio 32, Campus de Espinardo, Murcia, 30 100, Spain*

Correspondence should be addressed to Daniel Díaz-López; danielo.diaz@urosario.edu.co

Received 29 October 2021; Revised 5 December 2021; Accepted 6 December 2021; Published 27 December 2021

Academic Editor: Yafei Li

Over the last few decades, the Internet has brought about a myriad of benefits to almost every aspect of our daily lives. However, malware attacks have also widely proliferated, mainly aiming at legitimate network users, resulting in millions of dollars in damages if proper protection and response measures are not settled and enforced. In this context, the paper at hand proposes MalSEIRS, a novel dynamic model, to predict malware distribution in a network based on the SEIRS epidemiological model. As a result, the time-dependent rates of infection, recovery, and loss of immunity enable us to capture the complex dynamism of malware spreading behavior, which is influenced by a variety of external circumstances. In addition, we describe both offensive and defensive techniques, based on the proposed MalSEIRS model, through extensive experimentation, as well as disclosing real-life malware campaigns that can be better understood by using the suggested model.

1. Introduction

Malware attacks are currently a serious concern on computer networks and IoT devices [1, 2], although they are not a new phenomenon. The term *malware* refers to any software intentionally designed to cause damage to a computer, server, or computer network. Such malicious activity is performed by a threat agent, who may be an individual actor or even an organized crime group. A set of activities carried out by threat agents using specific techniques for some particular purpose is called a *malware campaign*. When conducting a study on how malware impacts a computer network, it is crucial to identify the type of malware that is being dealt with and its behavioral characteristics.

Different types of malware exist on the Internet, such as Trojan, spyware, ransomware, virus, worm, rootkit, among others, each having special characteristics that make them unique in the way they operate [3]. For example, the primary difference between a virus and a worm [4] is that a virus is triggered and spread through the intervention of the victim, whereas a worm is a stand-alone malicious program that can self-replicate and propagate through a network without requiring any human intervention. This situation makes a worm more infectious than a virus, but not necessarily more dangerous.

Some examples of malware campaigns are the *WannaCry* attack, carried out in May 2017 and estimated to have infected more than 200,000 computers in 150 countries [5], and the *ILOVEYOU* attack, which in May 2000 infected about 10% of the global internet-connected computers, causing global economy damages of up to $8 billion [6]. In this context, an analogy can be made between the spread of a virus or a worm in a computer network and the spread of diseases in human beings as part of a pandemic. In addition, the concept of disease applicable to living beings can be seen as a malware infection in devices connected to the Internet, vulnerable to certain attacks. Thus, various terms applicable in epidemiology could be extrapolated to a malware propagation context.

An infection occurs with a certain probability when contact between a susceptible and an infected individual occurs; likewise, this contact can be understood as a connection through the Internet, for example, between a vulnerable and an infected computer. Just as some diseases do not show symptoms immediately, an infected device may not present abnormal behaviors for a period, known as latency, either because the malware stays stealthy for a time by its own decision or because user intervention is required to complete the infection. As in the context of diseases, an infected device can recover through various strategies, such as patches, updates, or antivirus strategies, and obtain a period of immunity in which the malware will not be able to attack it again. However, as malware can also mutate or be obfuscated, the device can be vulnerable again. Therefore, an epidemiological model that rules the dynamics of a disease could also be applicable in the context of malware propagation.

One of the most basic epidemiological models is *SIR* [7], whose letters represent the natural transition between the following states: Susceptible, Infected, and Recovered. This model only considers an infection process caused by contact between a susceptible individual and an infected one, and a recovery process that occurs in an infected individual after an undetermined period. An important characteristic of this model is that the period of immunity is infinite, i.e., a recovered person can never get reinfected. One of its more complex derivations is called *SEIRS* [8], which adds a new state named Exposed, representing the latency period between being infected and being infectious. Besides, this former model does recognize possible reinfection and can add other dynamics such as births, external deaths, and deaths due to the disease.

Thus, this paper proposes MalSEIRS, a new model to analyze, better understand, and even predict the malware propagation in a computer network based on the existing *SEIRS* model, which considers the infection, recovery, and immunity loss rate as time-dependent to provide a more accurate meaning in a real context. The MalSEIRS model described in this article is not only an adaptation of the SEIRS model into a malware context but also a modification of its parameters and its rates. Furthermore, using the predictive capacity of the MalSEIRS model, defensive strategies could be proposed to mitigate highly contagious malware, guaranteeing not only the restoration of the operation of a computer network but also the security of the data. On the other hand, offensive strategies could also be proposed from the MalSEIRS model, which defines desired characteristics of a malware campaign in the context of a cyberwar [9]. In short, the main contributions of this paper are presented next:

(1) The proposal of a new model (MalSEIRS) to analyze and forecast the propagation of malware, which considers the dynamism of infection, recovery, and loss of immunity rate.

(2) The development of specific defensive and offensive strategies through exhaustive experiments, so as to demonstrate the usefulness of the MalSEIRS model.

The remainder of this paper is structured as follows: Section 2 introduces a theoretical background to ensure a good understanding of the key concepts employed in our research. Section 3 analyzes the major works proposed in the field. Next, Section 4 presents in detail the structure of the proposed MalSEIRS model, delving particularly into time-dependent rates. Then, Section 5 shows the experiments carried out to demonstrate the usefulness of the model in determining relevant defensive and offensive strategies. Section 6 then justifies the functions chosen to represent the spread of malware based on different real-world malware campaigns. Finally, Section 7 concludes the work, summarizing the main outcomes and highlighting some future research directions.

2. Background

As mentioned in Section 1, epidemiological models can be extrapolated in the context of malware propagation. Epidemiological models are mainly composed of differential equations as these relate a function with its derivative, allowing to describe the changes in a population exposed to an infectious disease. In order to model malware propagation, we will consider that a "node" belonging to a "population" may be a machine that is connected to the Internet, such as a router, a host, or a server. Thus, this section will explain how both the *SIR* and *SEIRS* epidemiological models may be applied in a computer network context.

2.1. SIR Model. The *SIR* model given by equation (2) only considers the three states of Susceptible, Infected, and Recovered, making it one of the easiest models to work. Each individual of the studied population belongs to only one of these states at a given moment. Also, it considers a static population and does not have a death rate, which means that the population is constant at all the stages of the analyzed pandemic. With this order of ideas, the total population in a SIR model is given by

$$N = S + I + R, \tag{1}$$

where

(1) N is the total population composed by all the nodes considered in the analysis,

(2) S (Susceptible) is the set of nodes that have a certain vulnerability, which means they can be infected by the malware,

(3) I (Infected) is the set of nodes already infected with the malware, and

(4) R (Recovered) is the set of nodes that have the necessary security measures to detect and delete the malware.

In addition, every node in a SIR model may shift between these states allowing two possible changes:

(1) Susceptible to infected ($S \longrightarrow I$): This means that a susceptible node had contact with an infected node,

and the malware has been spread from the latter to the former. In addition, this state change also considers the case when a susceptible node has been infected by a deliberate action from an attacker, not from another node.

(2) Infected to recovered ($I \longrightarrow R$): An infected node that has deleted the malware, e.g., through an antivirus running locally, may change from infected to recovered. This state change may also occur when the malware executes a self-destroy routine once its lifetime has finished.

Furthermore, the SIR model follows the structure shown in Figure 1, where the susceptible population turns into infected at a rate $\beta \in [0, 1]$, and the infected population recovers at a rate $\gamma \in [0, 1]$.

Overall, the system of equations is shown in equation (2), where $S(t) = S_t$, $I(t) = I_t$ and $R(t) = R_t$ represent the number of nodes at time t in each of the sets S, I, and R, respectively.

$$\begin{cases} \dfrac{dS}{dt} = -\beta SI, & S_0 = N - I_0, \\ \dfrac{dI}{dt} = \beta SI - \gamma I, & I_0 > 0, \\ \dfrac{dR}{dt} = \gamma I, & R_0 = 0. \end{cases} \tag{2}$$

2.2. SEIRS Model. In turn, the SEIRS model given by equation (4) is a variation of the SEIR model that in turn is a variation of the SIR model. It introduces the state Exposed (E), to indicate that a node is "incubating" the malware, as we will explain later. Moreover, this model introduces more parameters such as the incubation rate and the death rate caused by the virus or other causes, allowing the population not to be constant throughout the pandemic. Finally, the SEIRS model also takes into account the births from the population. Thus, the total number of nodes N of a population in the SEIRS model is computed as

$$N = S + E + I + R. \tag{3}$$

In addition, the SEIRS model allows the following transitions of states:

(1) Susceptible to exposed ($S \longrightarrow E$): The change from susceptible to exposed in an epidemiological context occurs when a person is in contact with the virus. In a malware context, a susceptible node becomes exposed when a malware has been installed in the node, but it has not executed its full functionality.

(2) Exposed to infected ($E \longrightarrow I$): In this case, the malware installed in a node has been executed, which means the node is infected.

(3) Recovered to susceptible ($R \longrightarrow S$): This means that the node has lost immunity, allowing subsequent reinfections. This may occur, for example, when an antivirus license has expired or when an operating system is not duly updated.

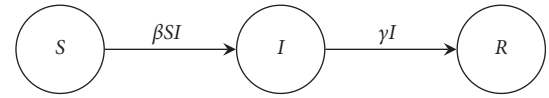

Figure 1: SIR model flow diagram.

(4) Susceptible to recovered ($S \longrightarrow R$): New security measures enabled in a node allow it to change from susceptible to recovered, meaning that the node cannot be infected by the malware. It works like a vaccine, and, therefore, the node is immune to the malicious software.

The *SEIRS* model works as observed in Figure 2, where susceptible nodes turn into exposed nodes at a rate $\beta \in [0, 1]$, and the infected population is sanitized against malware at a rate $\gamma \in [0, 1]$, i.e., the recovery rate. Moreover, the exposed population might be infected with a latency $\sigma \in [0, 1]$, i.e., the malware execution rate. As for those nodes recovered, they could actually lose their immunity with a rate $\omega \in [0, 1]$. Furthermore, in each state, there is a death rate $\mu \in [0, 1]$ due to external causes. Finally, the infected nodes could also "die" because of the virus at a rate $\alpha \in [0, 1]$, and the susceptible population also considers the new "births" with a rate μN. Table 1 contains a summary of all the parameters of the SEIRS model.

Finally, we could represent the previous diagram with the system of equations shown in equation (4).

$$\begin{cases} \dfrac{dS}{dt} = \mu N - \beta IS + \omega R - \mu S, \\ \dfrac{dE}{dt} = \beta IS - \sigma E - \mu E, \\ \dfrac{dI}{dt} = \sigma E - \gamma I - (\mu + \alpha) I, \\ \dfrac{dR}{dt} = \gamma I - \omega R - \mu R. \end{cases} \tag{4}$$

3. State of the Art

As previously stated, malware is still one of the main threats to computer networks. In particular, its propagation within system assets poses several challenges that demand robust solutions to model and, possibly, prevent its dangerous diffusion.

In Ref. [10], the authors proposed a dynamical propagation model to study the spread of malware within a virtual environment. Specifically, the authors explored different malware propagation factors, such as installing antiviruses in the virtual machines composing the cloud ecosystem. Similar to Ref. [16], the presented model is based on three states, namely, Protected (P), Susceptible (S), and Infected

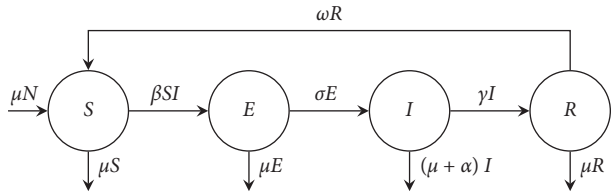

Figure 2: *SEIRS* model flow diagram.

Table 1: Parameters of the *SEIRS* model.

Parameter	Definition
β	Malware propagation rate
γ	Malware removal rate
σ	Malware execution rate
ω	Rate of loss of antiviruses and security measures (loss of immunity rate)
α	Machine unavailability rate caused by malware
μ	Machine unavailability rate caused by other reasons

(I). The malware propagation is modeled with a set of differential equations, in which the states are depicted with their time dependence. Then, equilibrium points and their stability (i.e., local and global) are computed, and the system of equations is simulated, showing that the malware cannot be eliminated from the environment if the parameters of the model are not correctly tuned.

In addition, authors in Ref. [11] applied the SIR model to capture the features of malware propagation. After a thorough discussion on the similarity of pathogens and malware spreading, the SIR is generalized to analyze malware diffusion in susceptible populations with two infection methods, i.e., Peer-to-Peer (P2P) and Central Source (CS). Interestingly, the authors tested the presented model on real-world data, a dataset containing malware traces from April 2017 to April 2018. Results on the prediction demonstrate that it is possible to derive the malware diffusion methodology from its temporal profile of the model fit to it. Besides, the model suggests that vaccination (i.e., malware signatures) is more effective in containing the spread of CS malware.

Furthermore, a mathematical model that contemplates quarantine and vaccination to defend against worm attacks in Wireless Sensor Network (WSN) is proposed in Ref. [12]. Concretely, the authors leverage the capabilities of the SEIQRV model, which adds Quarantined (Q) and Vaccinated (V) to the abovementioned states. After computing the equilibrium and stability of the system, various simulations are executed to calculate: (i) the effects of communication radius on the worm diffusion, (ii) the impact of the node density, (iii) the evaluation of the model under different conditions (i.e., changing the parameters), and (iv) the comparison between SEIRV and SEIQRV models. To this extent, the authors claim that the experimental results can help in the development of antivirus software specific for WSNs.

Besides, authors in Ref. [13] analyzed a novel theoretical model to simulate the expansion of malware in WSNs. Particularly, they studied a SCIRS model that considers population dynamics, carrier (C) compartment, vaccinations, and possible reinfections. As an extension of the proposal in Ref. [17], the authors presented the model and calculated the steady states. Then, numerical simulations are introduced together with the proposal of high-level security countermeasures against the threat.

Moreover, another malware propagation model for mobile Internet is presented in Ref. [14]. Specifically, the authors proposed an SLBQR model, which incorporates the Latent (L) and Breaking out (B) states to detail the malware diffusion. Remarkably, they consider vaccination with temporary immunity and quarantined strategies. After discussing quite realistic assumptions, the authors calculated the equilibrium and numerically simulated the model. The main conclusions of the work are two: (i) the endemic equilibrium will not be worm-free because of the potential worm variations, and (ii) it is possible to enforce a quarantine strategy to stop the expansion of the threat.

Also, authors in Ref. [15] introduced a time-delayed worm propagation model with variable infection rate. Assuming that the worm infection is a nonlinear process and that temporary patches can lose efficiency, they presented a SIQVD model, which adds the Delay (D) state to consider the time delay in states' transitions. Then, the stability of equilibrium is determined. Simulations are run, selecting the Slammer worm for the experiments. The outcomes underlined the dependence of the virus spread with the introduced time delay, i.e., the propagation can be predicted with the control of such a variable.

Overall, the analyzed works represent an advance of the state-of-the-art regarding the modeling of worm propagation in several scenarios. Nevertheless, some interesting features are still missing or partially addressed by the literature. To this extent, Table 2 compares those works based on the main characteristics proposed in this article. Precisely, none of them provides the source code of the implemented experiments. In this direction, we believe that the dissemination of the simulation code is fundamental to share the outcomes and let other researchers compare models with more equity. In addition, very few works consider the intrinsic dynamism of the equation parameters

Table 2: Comparison of the related works highlighting the main features.

Related work	Model	Simulation code	Dynamic parameters	Defense vs offense	Real-world threat
Gan et al. [10]	SIP	×	×	×	×
Levy et al. [11]	Generalized SIR	×	×	×	✓
Ojha et al. [12]	SEIQRV	×	×	×	×
Hernandez Guillen et al. [13]	SCIRS	×	×	≈	×
Zheng et al. [14]	SLBQR	×	≈	×	×
Yao et al. [15]	SIQVD	×	≈	×	✓
Our proposal	Dynamic SEIRS	✓	✓	✓	✓

Legend: ✓ Yes – × No – ≈ Partially.

for the propagation model. Indeed, bearing in mind the dynamism of modern computer networks, the MalSEIRS model proposed in the paper at hand contains equations dependent on the time to have a correct characterization. Besides, the offensive perspective of the threat spread is generally neglected in the current state-of-the-art-related works. Undoubtedly, the main goal of those models is to encapsulate the features of the threat in order to contain/block it from a defensive viewpoint. However, our paper studies the spread also from an attacker angle, aiming at potentially acquiring additional knowledge. Last but not least, discussion on possible applications of the proposed models on real-world threats is also lacking in related works. By doing so, as we do in our paper, the mathematical model and simulations would possess a closer connection with existing worms, thus proving their robustness.

4. Proposed MalSEIRS Model

This section describes the key aspects of the proposed MalSEIRS malware propagation model, explaining the reasons that support each of its main components. The proposed model is an innovative modification of the SEIRS model to allow: (i) adaptation of the infection, recovery, and loss of immunity rates to make them time-dependent, (ii) inclusion of births in the context of computer networks, and (iii) addition of the concept of vaccine to shield recovered nodes. Thus, the system of differential equations proposed for modeling the distribution of malware in a network is shown in equation (5):

$$\begin{cases} \dfrac{dS}{dt} &= p\Lambda - \beta(t)IS + \omega(t)R - (\mu + \phi)S, \\ \dfrac{dE}{dt} &= \beta(t)IS - \sigma E - \mu E, \\ \dfrac{dI}{dt} &= \sigma E - \gamma(t)I - (\mu + \alpha)I, \\ \dfrac{dR}{dt} &= (1-p)\Lambda + \gamma(t)I - \omega(t)R - \mu R + \phi S, \end{cases} \tag{5}$$

where $\beta(t), \gamma(t), \omega(t) \in [0, 1]$ are the infection or malware propagation rate, recovery or malware removal rate, and loss of immunity rate, which will be described in Sections 4.1–4.3, and are represented by equations (4)–(6), respectively. In addition, $\phi, p \in [0, 1], \Lambda \geq 0$ are the rate of immunization, birth's susceptibility rate, and the new nodes that connect to the network, as specified in Sections 4.4 and 4.5, respectively. Finally, $\mu, \sigma, \alpha \in [0, 1]$ are the machine unavailability rate caused by other causes, malware execution rate, and machine unavailability rate caused by malware as described previously in Table 1.

4.1. Infection Rate $\beta(t)$. The infection or malware propagation rate is generally considered a constant in typical SEIRS models. However, we believe that such a rate variate over time depending on the circumstances of the infection and the environment [18], including the different reactive countermeasures deployed to contain the attack [19]. Particularly, when the malware outbreaks, a high number of infected hosts might be reached in a short time, which is called a peak, leading to a high infection rate $\beta_0 \in [0, 1]$. However, such an infection rate β_0 will eventually decrease due to different reaction actions [20] against the attacks such as (i) network congestion as an effect of the propagation of the malware campaign, (ii) update of host-based anti-malware solutions currently deployed on susceptible nodes to include indicators of compromise that detect the malware, (iii) patching or upgrading of operative systems of susceptible nodes to mitigate related vulnerabilities, (iv) setting up of network security solutions like firewalls or IPS (Intrusion Prevention System) that contain the attack before it reaches the susceptible nodes, or (v) simply disconnecting nodes to make them unreachable.

$$\beta(t) = \frac{\beta_0}{1 + \xi I(t)}. \tag{6}$$

Thus, the infection rate $\beta(t)$ will be computed as observed in equation (6), which includes the initial infection rate β_0 and $\xi > 0$ as a positive constant used to adjust the speed of the decrease of the infection rate. This function was selected since it represents the reduction in infection rate simply. It also allows to adjust its decay speed depending on the context, and it approaches zero when the number of infected goes to infinity.

4.2. Recovery Rate $\gamma(t)$. The recovery or malware removal rate could also variate over time depending on the availability of treatment to be applied to infected nodes. At the beginning of a new malware campaign, it is expected that the recovery rate may be low, depending on the grade of the

novelty of the attack, e.g., a malware could be considered with more novelty if it exploits a zero-day vulnerability [21]. However, as the malware campaign evolves, it is expected that new corrective mechanisms to clean and sanitize infected nodes become available, driving a possible increase in the recovery rate. Such an increase in the recovery rate depends not only on the availability of corrective mechanisms but also on the timely application of them to the infected nodes.

$$\gamma(t) = \tanh\left(\frac{I(t)}{c}\right). \tag{7}$$

Thus, the recovery rate $\gamma(t)$ (see equation (7)) may be represented by a hyperbolic tangent, as it offers rates between 0 and 1 for positive time values, and it tends to 1 as the number of recovered nodes grows. Constant $c > 0$ is a positive value, and it is used to determine how fast the recovery rate will approach 1, i.e., how fast the adequate update and protection measures will be applied by the victim organization.

4.3. Loss of Immunity Rate $\omega(t)$. The initial stage of a malware attack can be chaotic in terms of the stability and efficiency of the security measures adopted by the susceptible and infected nodes. The process of tuning security measures to make them stable and efficient may last a time depending on: (i) the available information about a malware campaign, where new indicators of compromise may show up, (ii) the discovery of some new functionalities in the malware that were sleeping in the initial stage of the malware campaign, or (iii) the appearance of mutated or obfuscated versions of the malware, among others [22, 23]. This situation may provoke that the loss of immunity rate $\omega(t)$ oscillated for a time, even if it is also expected that this rate will eventually be 0, representing a situation where all nodes in the network become immune.

$$\omega(t) = \left|e^{-at}\cos\left(\frac{2\pi t}{m}\right)\right|. \tag{8}$$

The loss of immunity rate $\omega(t)$ may be computed as stated in equation (8), which is a damped cosine to represent the oscillations mentioned previously. The absolute value is defined to guarantee that the rate values are between 0 and 1. The parameter $a > 0$ determines the strength of the damping, i.e., the higher its value, the faster the loss of immunity rate will stabilize at 0. On the other hand, $m > 0$ determines the length of the cosine period and allows to extend the time of the oscillations.

4.4. Rate of Immunization ϕ. The rate of immunization ϕ will depend on the existence of a vaccine, which in this context is represented as a preventive countermeasure applied on susceptible nodes to make them reach the status recovered without being infected previously. In the context of a computer network, the vaccine may be seen as the installation of an anti-malware solution, the update of the signatures of an existing antivirus solution, or another protection strategy that immunizes a susceptible device. For instance, a value of $\phi = 0.5$ indicates that a matching security solution, such as an update, is deployed on half of the devices that are susceptible to being attacked by a threat at each time step, e.g. each day.

4.5. Birth's Susceptibility Rate p. Parameter Λ represents the new nodes that connect to the network, which may be seen as the birth of individuals. New nodes will enter the group of those susceptible or recovered based on birth's susceptibility rate p.

For instance, a Λ value of 5 means that at, each time step, 5 new devices enter the network. These may or may not be protected against the threat. If $p = 0.6$, it means that 3 of these devices are susceptible to malware, while the remaining 2 are protected against it and enter directly into the group of recovered.

5. Analysis of the MalSEIRS Model

Several experiments were conducted on the proposed model to assess its performance when changing its parameters to improve offensive and defensive strategies. The initial conditions are reported in Section 5.1, while Section 5.2 analyses a parameter-based examination of the proposal. Then, Section 5.3 compares the presented model with the exposed state-of-the-art proposals. Finally, in Section 5.4, certain specific scenarios are explored in order to establish some optimal values.

5.1. Initial Conditions. We assumed a network initially composed of $N = 33$ nodes with the initial conditions defined as below. These initial conditions may represent a small network composed of PCs or IoT devices, i.e., a quite common scenario in smart homes and SMB (Small and Medium businesses). In such a context, one single device may be infected easily by downloading and installing a malicious application. As the infected device has connectivity with the other devices of the network, the spreading of the infection will be a matter of time.

(1) $S(0) = 30$
(2) $E(0) = 2$
(3) $I(0) = 1$
(4) $R(0) = 0$

Table 3 shows the initial parameters considered for the experiments, where we picked specific values for the parameters related to the malware. We considered that the malware would be infectious at the first stage of the attack and that it would execute itself right after its installation, which is represented by a β_0 and σ close to the upper limit. Also, we select a reduced value of α to represent a malware that infects but does not cause unavailability. In addition, chosen values for a and m indicate that nodes are more vulnerable to lose their immunity over time, which means loss of security measures caused by the changes in the

Table 3: Initial parameters of the model for the experiments.

Parameter	Description	Value
β_0	Malware initial propagation rate	0.8
ξ	Parameter for infection rate (Equation (6))	1
μ	Machine unavailability rate provoked by other causes	0.1
α	Machine unavailability rate provoked by malware	0.2
c	Parameter for recovered rate (Equation (7))	5
σ	Malware execution rate	0.9
ϕ	Immunization rate	0.46
p	Birth susceptibility rate	0.1
Λ	Number of new nodes	1
a	Parameter to control strength of damping in loss of immunity rate	0.1
m	Parameter to control time of oscillations in loss of immunity rate	1

malware campaign (new obfuscation techniques, new vulnerabilities being exploited, among others).

Also, to avoid increasing the considered factors over the previous scenario, we consider that the death rate μ by external causes is not considerable. Value of parameter ξ (1) represents network congestion and the ability to implement containment measures easily. At last, we contemplated that the nodes will be immunized pretty fast by setting p with a minimal value and ϕ near to 0.5.

The simulation results of the proposed model with the previously defined parameters are depicted in Figure 3, where the simulation time is 30 time units. Time units refer to the time scale (seconds, hours, etc.), and it depends on the attack context. Figure 3 is consistent with the typical observed behavior of a controlled attack, where the number of susceptible, exposed, and infected nodes eventually go to zero, while all devices in the network converge to the group of recovered. The greatest change between groups occurs in the first 5 units of time: malware attacks a certain number of devices quickly (the infected peak is reached with 5 compromised devices), but its effectiveness decreases significantly once the security team is able to respond to the attack. Small initial oscillations in groups S and R show the effect of loss of immunity. However, such effect is not significant enough and quickly drops to zero, so these groups stabilize quickly.

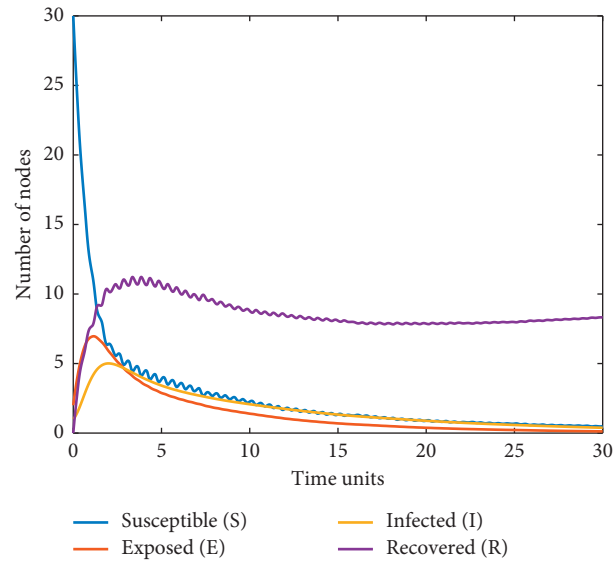

Figure 3: Simulation of the proposed model when applying the initial parameters.

5.2. Analysis of MalSEIRS in terms of Its Parameters. Besides the initial experiments conducted in the previous section, several simulations were performed by varying each parameter given in Table 3 while keeping constant the values of the rest of the parameters. These experiments will allow us to identify the trend of the simulations of the MalSEIRS model when the parameters change. Variation of the parameters during the simulation aims to include at least three meaningful values in the scale of each parameter, e.g., a low, a medium, and a high value for each case, as seen in Figures 4 and 5. All code and data used in the simulations conducted in the present paper are available for open consult in the project repository (https://github.com/BelisaDi/Malware-Propagation-Model).

When the initial infection rate β_0 takes a value as small such as 0.2 (see Figure 4(a)), the peaks of the exposed and infected nodes decrease in comparison with the ones obtained in Figure 3 where β_0 was 0.8, meaning that the scale of propagation of the malware is small, reducing the impact of the attack. For $\beta_0 = 0.5$, we observe that the propagation behavior stays generally between the values obtained when β_0 is 0.8 and 0.2. We also see that the values of the recovered nodes are similar during the first 2 units of time for all values of β_0, but gain difference between them when the peak is reached and beyond. It is important to notice that results obtained when β_0 is 0.5 and 0.8 are closer between them, in comparison with the results obtained when β_0 is 0.2. This kind of situation represents for the attacker the importance to manage a considerable initial infection rate β_0, as it would help to take possession of the network, and therefore make difficult the recovery from the attack.

Moreover, a low malware execution rate σ, as the one shown in Figure 4(b) ($\sigma = 0.2$), could indicate that the malware is a virus, i.e., it needs the interaction of a person for the infection, increasing the latency in the propagation of the malware. For a malware execution rate of $\sigma = 0.2$, the exposed nodes surpass the number of infected nodes, in contrast with Figure 3 ($\sigma = 0.8$). A high malware execution rate, e.g., $\sigma = 0.9$, indicates that the malware is a worm, and

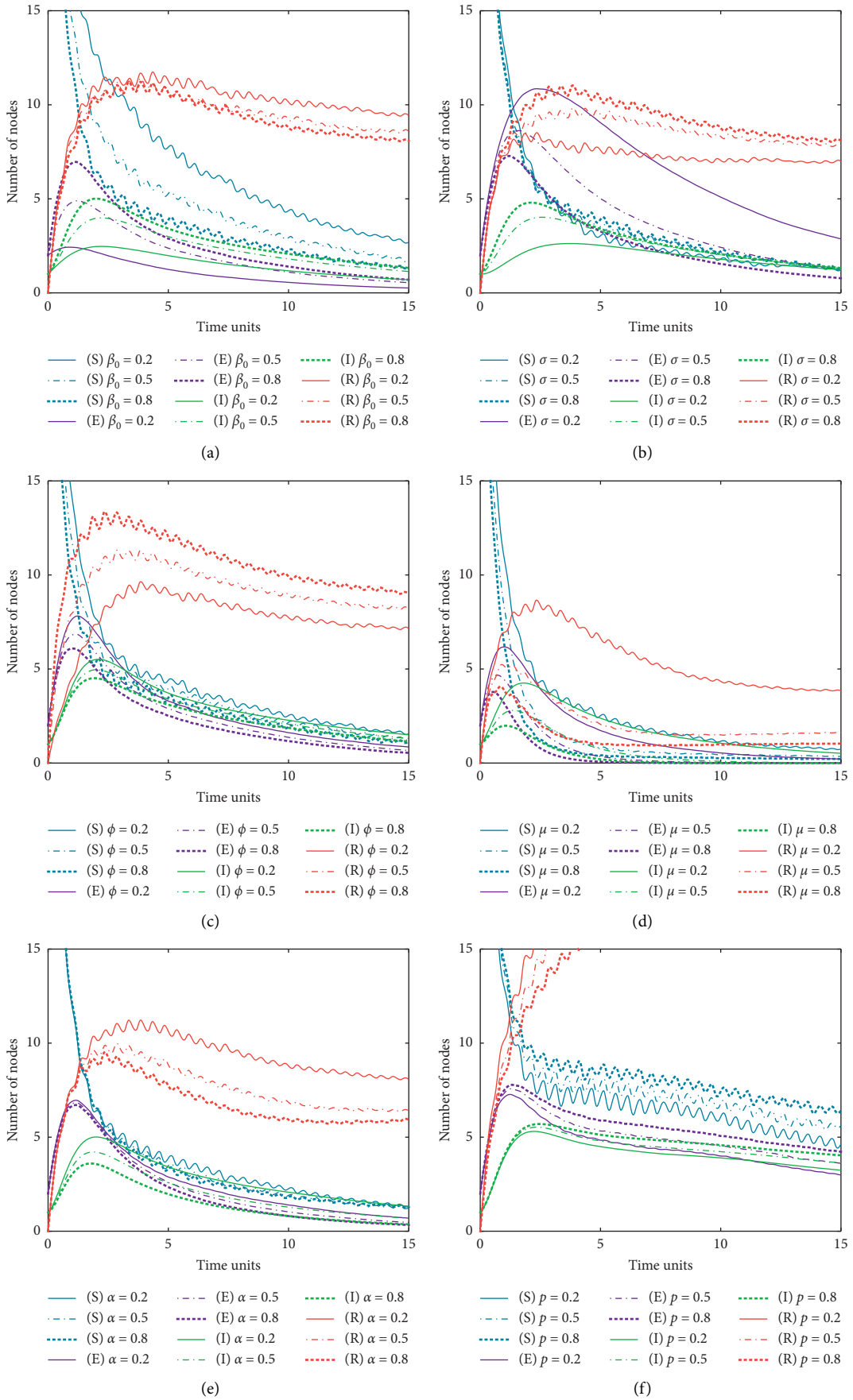

FIGURE 4: Simulation of the MalSEIRS model varying at different constants and rates. (a) Variation of the constant β_0 used in the infection rate β_t. (b) Variation of the malware execution rate σ. (c) Variation of the rate of immunization ϕ. (d) Variation of the machine unavailability rate provoked by other causes μ. (e) Variation of the machine unavailability rate provoked by malware α. (f) Variation of the birth's susceptibility rate p when the number of new nodes $\Lambda = 5$.

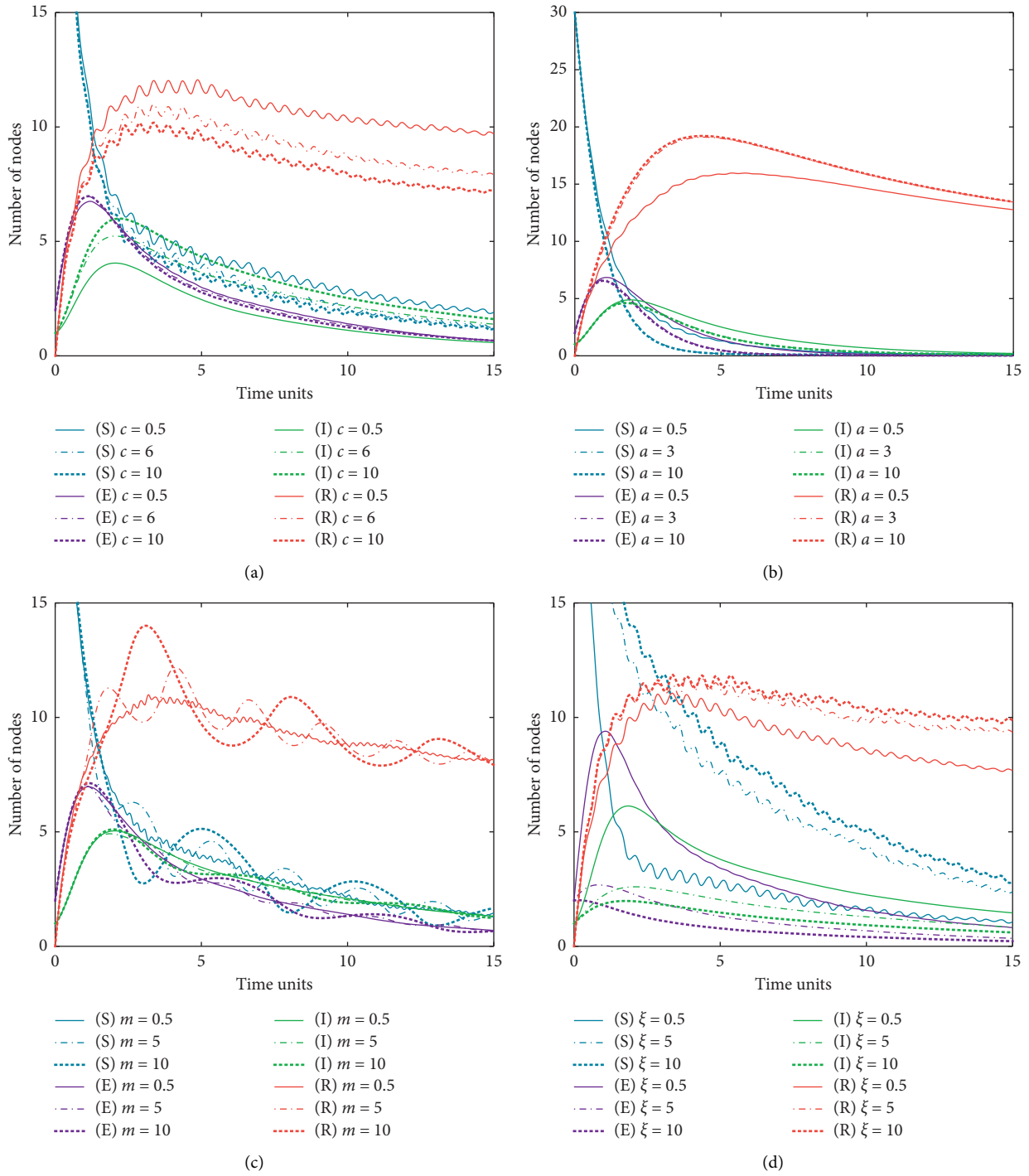

Figure 5: Simulation of the MalSEIRS model varying at different constants and rates (Cont'd). (a) Variation of the constant c used in the recovery rate $\gamma(t)$. (b) Variation of the constant a used in the loss of immunity rate $\omega(t)$. (c) Variation of the constant m used in the loss of immunity rate $\omega(t)$. (d) Variation of the constant ξ used in the infection rate $\beta(t)$.

so it would propagate faster throughout the network. Malware could also have some execution characteristics such as *Scheduled Task/Job* [24], allowing the malware to be executed at a certain time, which may be represented by a $\sigma = 0.5$. This last situation means that the malware has the same probability of being executed or not in each time, representing an average malware.

In addition, when increasing the immunization rate ϕ, as shown in Figure 4(c) ($\phi = 0.8$), the number of recovered nodes increases while the susceptible nodes decrease a little faster in comparison to $\phi = 0.5$, which means that some security measures probably have been adopted (e.g., host-based firewall, anti-malware solutions or others) to overcome the attack and minimize the damages, and therefore

the malware could not easily propagate through the network (actually, it would be a suitable way to finish the infection). Nevertheless, the network could always have a susceptible node. Moreover, a small value of ϕ like 0.2 may represent a vulnerable network with no containment measures. Actually, we can see in Figure 4(c) that when $\phi = 0.2$ the number of susceptible increases and the recovered decreases since there is no active immunization of devices. In addition, it could also indicate that the malware has defense evasion [24] to avoid detection, which would make it difficult to detect.

The machine unavailability rates provoked by other causes (μ) and provoked by the malware (α), as indicated in Table 3, were also submitted to variation. When μ takes a high value such as $\mu = 0.8$ (see Figure 4(d)), each group stabilizes near to zero. Something similar happens with $\mu = 0.5$, where the different groups of nodes tend to zero, indicating that around 50% of the network nodes have been disconnected or that the network nodes are damaged due to external causes, not by the malware. Those situations imply the end of the malware propagation due to the lack of nodes to infect, i.e., a total disconnection of the network since there are no nodes alive. A high μ represents an easy way to terminate the infection even if it implies to leaving the network inoperative. On the other hand, when α increases up to a value of $\alpha = 0.8$ (see Figure 4(e)) the number of exposed and infected nodes decrease since the malware is so much destructive. Indeed, if α is so high, it means that the malware did not have time to propagate throughout the network, which is more typical in an attack only focused on a specific part of the network or in a specific kind of malware like a rootkit.

In addition, when α increases from 0.2 to 0.8, we observe that the group of recovered nodes reduce significantly since there are no nodes to recover due to mortality. Also, in such a situation, the infected peak decreases considerably, indicating that not many machines were infected as the mortality was so high.

When the number of new nodes is $\Lambda = 5$ and the birth's susceptibility rate $p = 0.2$ (see Figure 4(f)), we observe that the recovered nodes increment quickly, and it would even be relatively safe to allow new nodes to enter the network. Furthermore, this means that the new nodes on the network are posing the necessary security measures to overcome the infection. On the other hand, when $p = 0.9$ (see Figure 4(f)), we observe an increment in the susceptible nodes, which is not an ideal situation because it means that the pandemic might take much longer to be fully extinguished. Moreover, we can see that the number of recovered nodes when $p = 0.2$ or $p = 0.5$ decreases when compared with $p = 0.8$, which means that the malware could last through time since the network is allowing the access of vulnerable nodes.

As mentioned in Section 4.2, a high value of c in the recovery rate $\gamma(t)$ (see equation (7)) means that the recovery will be slow. In this regard, Figure 5(a) shows the results of testing our model with $c = 10$ and $c = 6$. Such a scenario means that the malware is using defense evasion to avoid uninstalling [24], as we mentioned before with the immunization rate. If we compare these outcomes with the ones obtained in Figure 3 and results obtained in Figure 5(a) with $c = 0.5$, we observe that the peak of the recovered nodes decreased while the peak of the infected nodes increased, meaning that the recovery of nodes is more difficult. Such a circumstance most likely occurs due to situations like anti-sandbox properties of the malware or other mechanisms that the malware poses to identify that an anti-malware solution is running in the node and then avoiding to show malicious behavior.

The parameters for the loss of immunity rate $\omega(t)$ were also varied as part of our experiments. If the parameter a is incremented up to 3 or 10 (see Figure 5(b)), i.e., a value bigger than the considered in the initial parameters ($a = 0.1$), $\omega(t)$ would approach to zero pretty fast, meaning that it would be remarkably hard for a node to lose immunity as seen in Figure 5(b)). Indeed, such a situation would cause the end of the malware propagation. On the other hand, as the parameter m causes oscillations on the susceptible and recovered nodes, increasing m would prolong the oscillation period, as shown in Figure 5(c).

Finally, the parameter ξ used in the infection rate $\beta(t)$ determines how fast the propagation would approach to zero (see equation (6)). For a great value of ξ, $\beta(t)$ will be near to zero pretty fast, and on the other hand, for a small value of ξ, the infection rate will decrease slower. For that reason, from a defensive perspective, it is better to have a great value of ξ, as shown in Figure 5(d), where the exposed and infected nodes peaks get reduced when $\xi = 10$. This situation contrasts with the case when $\xi = 0.5$, which shows higher peaks. Thus, a high ξ means that the malware is not too infectious or that the attacker centralized the attack, while a low ξ means that the malware is too infectious.

The abovementioned experiments helped us to better understand the behavior and dynamics of the proposed MalSEIRS model in order to predict malware propagation under several realistic circumstances and scenarios.

5.3. Comparative Evaluation of MalSEIRS. Additional simulations were run to verify the usefulness of the proposed model, comparing its behavior to that of a background model, the SEIRS [8], and two related state-of-the-art models, the SLBQR [14] and the SIQVD [15]. The parameters used for all models are described in Table 4, and the initial conditions utilized were proposed in subsection 5.1.

Starting with the simplest model, let us review the SEIRS model, given by equation (4). Note the clear similarity with the parameters used in the proposed model, only that in this case, all rates are constant.

Although the recovery rate is high, with more than half of those infected in each temporal step recovering, and the rate of loss of immunity being minimal, observe that the number of infected in SEIRS never achieves zero (see Figure 6(a)), suggesting an endemic equilibrium state. Furthermore, because the infection rate never diminishes, the peaks of both infected and exposed devices are substantially greater. In this instance, the proposed model implies that, in the event of an attack, it will finally be feasible to implement suitable and relevant security measures to

Table 4: Values selected for different parameters used in the comparative evaluation of MalSEIRS.

Parameter	Value according to the evaluated model			
	MalSEIRS	SEIRS [8]	SIQVD [15]	SLBQR [14]
β_0 or β	0.8	0.8	0.8	0.8
ξ	1	—	1	—
γ	—	0.6	0.5	—
μ	0	0	—	0
α	0	0	—	—
c	5	—	—	—
σ	0.9	0.9	—	0.9
ϕ	0.46	—	0.46	—
ε	—	—	—	0.46
Λ	0	—	—	—
p	0	—	—	—
ω	—	0.05	0.05	—
a	0.1	—	—	—
m	1	—	—	—
θ	—	—	0.6	—
δ	—	—	0.2	0.6
λ	—	—	—	0.6
τ	—	—	5	5

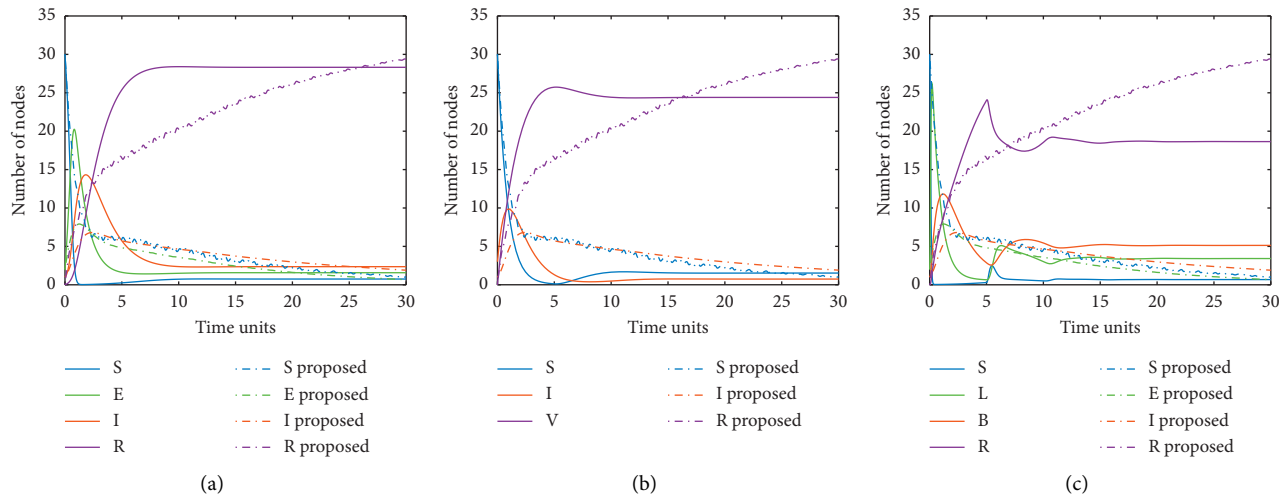

Figure 6: Comparisons between MalSEIRS, SEIRS, SIQDV, and SLBQR models. (a) MalSEIRS vs SEIRS [8]. (b) MalSEIRS vs SIQVD [15]. (c) MalSEIRS vs SLBQR [14].

eradicate the malware, which is a reasonable assumption. Furthermore, since SEIRS stabilizes fast, it does not account for counterattack scenarios such as the deployment of mutated variants of malware via a command and control center, e.g., using the techniques described in Ref. [24].

Moving on to more interesting models, the SIQVD presented by Yao et al. [15] should be analyzed. Susceptible (S), infected (I), quarantined (Q), vaccinated (V), and delay hosts (D) are the statuses included in the model. The most relevant thing is that, like the proposed model, the infection rate $\beta(t)$ parameter is time-dependent and adopts the same form as described in Section 4.1. The model is given by equation (9). Although the original article has more details on the model's behavior and construction, considerable effort was made to ensure that the values of the parameters utilized were as similar as possible to those suggested in the proposed model.

$$\begin{cases} \dfrac{dS}{dt} = -\beta(t)I(t)S(t) - \phi S(t) + \omega V(t-\tau), \\ \dfrac{dI}{dt} = \beta(t)I(t)S(t) - (\gamma+\delta)I(t), \\ \dfrac{dQ}{dt} = \delta I(t) - \theta Q(t), \\ \dfrac{dV}{dt} = \phi S(t) + \gamma I(t) + \theta Q(t) - \omega V(t), \\ \dfrac{dD}{dt} = \omega V(t) - \omega V(t-\tau). \end{cases} \tag{9}$$

Therefore, as can be seen in Table 4, the values for the beta function (β_0, ξ) and the vaccination rate (ϕ) remain the same as those proposed. Relatively high values of recovery rates for infected (γ) and quarantined (θ) are considered to compensate for the dynamism of the function proposed in this article, as well as a low rate of loss of immunity (ω). The quarantine rate (δ) remains at a low value since it is impossible to compare for that group.

Results can be seen in Figure 6(b). One problem with not considering a latency state is the number of scenarios and/or solutions that are left out. As the transition from susceptible to infected is assumed to be immediate, this suggests that the malware executes and infects the device as soon as it is on the machine, implying that it is suitable for a specific malware family such as worms [4]. Along the same line, an overlooked option is warning victims not to execute the malicious file, which will prevent infection even if the executable is already on the machine. This is described as an execution technique in the MITRE matrix [24], where techniques such as social engineering can be relied on to get the malware to execute. However, mitigations such as M1017, which is based on training users to learn to recognize malicious files and phishing, can be modeled with our model and not with the SIQVD one.

Furthermore, a constant loss of immunity rate means that for a certain percentage of devices, an appropriate protective state will never be attained. This may be sufficient for a large-scale situation such as the whole Internet, but it is critical for smaller scales such as businesses to obtain this status. It makes more sense in these sorts of situations for the loss of immunity to reach zero once the attack is under control. In other words, SIQVD does not consider the presence of a trained incident response team [25] or an environment in which digital security is promoted, maintaining, for example, antivirus licenses, important updates, etc.

In the same way as the previous case, the proposed model controls the attack while the simulation for the SIQVD model reaches an endemic equilibrium. In addition, the peak for those infected is more prominent in the latter case, although it handles a variable infection rate over time in the same way as that presented in this article.

Finally, the SLBQR model proposed by Zheng et al. [14] can be reviewed. It considers the susceptible (S), latent (L), breaking-out (B), quarantined (Q), and recovered (R) states. The analogy between the latent and the exposed state (E) and between the breaking out and the infected (I) in the case of SEIRS is clear. Such a model is given by equation (10), and in the same way as in the previous case, an important effort was made to simulate it under equal conditions for a fair comparison.

$$\begin{cases} \dfrac{dS(t)}{dt} = -\beta S(t)\left(L(t) + B(t)\right) - \mu S(t) + \mu N + \varepsilon L(t-\tau)e^{-\mu\tau} + \lambda Q(t-\tau)e^{-\mu\tau}, \\ \dfrac{dL(t)}{dt} = \beta S(t)\left(L(t) + B(t)\right) - \mu L(t) - \sigma L(t) - \varepsilon L(t), \\ \dfrac{dB(t)}{dt} = \sigma L(t) - \delta B(t) - \mu B(t), \\ \dfrac{dQ(t)}{dt} = \delta B(t) - \mu Q(t) - \lambda Q(t), \\ \dfrac{dR(t)}{dt} = \varepsilon L(t) + \lambda Q(t) - \mu R(t) - \varepsilon L(t-\tau)e^{-\mu\tau} - \lambda Q(t-\tau)e^{-\mu\tau}. \end{cases} \tag{10}$$

The chosen values are found in Table 4. Here all rates are constant. This is unrealistic, as discussed later in Section 6. For example, it is notorious that the infection rate of the Code Red virus peaked in its attack and then fell rapidly [26]. The parameters retain their meaning, and N is the total population; however, the inclusion of the ε and λ rates is worth noting.

Although ε can be interpreted as the vaccination rate and λ as the recovery rate for quarantined devices, these also influence the loss of immunity of recovered devices. Interestingly, this loss of immunity does not occur based on the number of recovery hosts present but rather on the number of latent and quarantined devices. This is dangerous because if the number of devices in Q and L is much larger than that of R, it is possible to reach negative values for the latter, which does not fall within the feasibility region. This is not to mention that it does not make sense that the rate of loss of immunity is the same as the rate of vaccination, for example, since they are two completely different processes.

The simulation results can be seen in Figure 6(c). The behavior of the proposed model is essentially smoother and more predictable than that of the SLBQR, which, due to the delay, presents two large peaks in each of the groups: one at the beginning of the simulation and another 5 units of time later. This may also be due to the high rate of loss of immunity.

The peak of latents (or equivalently, of exposed) is too high for the SLBQR model and occurs very fast compared to other models previously analyzed. This presents a rather dangerous scenario in which the malware spreads to almost all hosts instantly. This behavior is not applicable to scenarios where best security practices are applied or considering that network congestion can greatly reduce the infection rate. In addition, this also happens because a contact with a latent can also induce a contagion, which again is only applicable in certain cases, since once more, the malware may require user execution. Besides, once more, the SLBQR reaches an endemic equilibria in which the malware is not successfully eradicated.

5.4. Applying MalSEIRS to Different Cybersecurity Scenarios. This section contains two analyses aimed to highlight the feasibility of our proposed MalSEIRS model for the corresponding adoption of defensive and offensive methods. The first analysis is focused on the application of our model to compare a loud vs a subtle attack, and the second analysis aims to describe how our model describes a network under attack. The impact of each parameter of our model has already been evaluated independently, but the following tests allow us to further justify Section 6.4 of recommendations, as we will see later. It is also worth noting that, unless otherwise stated, the parameter values are still determined by Table 3, and the simulation time is 30 time units. Furthermore, zero devices are exposed and recovered in the tests, with only one host infected and the rest of the population susceptible.

5.4.1. A Loud vs Subtle Attack. In this first analysis, we consider the behavior of an infection in computer networks, observing how it evolves depending on the initial susceptible population and some other factors such as the number of new nodes (Λ) in the network and its probability of being susceptible (p). Moreover, we also analyzed how an attack may be impacted by the parameters β_0, i.e., the initial infection rate, and ξ, i.e., positive value that adjusts the speed of decrease of the infection rate.

In this order of ideas, simulations over 3 different scenarios were performed in order to analyze the behavior of infected nodes in two different situations: (i) an infectious malware (loud attack) that spreads itself pretty fast and (ii) a malware that makes damages in a prolonged time but is not too infectious (subtle attack).

The settings and results for scenario 1 are shown in Table 5. This scenario assumes that ξ depends on β_0, where ξ takes values between 0 and 1. This scenario illustrates a computer network with infectious malware, such as a worm or ransomware like WannaCry, spreading pretty fast from the beginning of the attack.

In Table 5, we found that as the number of susceptible nodes increases, the initial infection rate β_0 should also increase (when $\beta_0 = \xi$) in order to get a maximum number of accumulated infected nodes. Moreover, in order to get a maximum infected peak, the malware should be more infectious in the early stages of the attack. A similar situation occurs when $\xi = \beta_0/2$, where the maximum accumulated infected nodes or the maximum infected peak is achieved when β_0 gets close or equal to 1.

The situation represented in scenario 1 would suggest that the attacker wants to get "possession" of the network in the shortest amount of time. Nevertheless, after the infection peak is achieved, the number of infected nodes will probably decrease as fast as it increases, which may mean that the malware will not last a long time due to some factors such as network congestion or implementation of security countermeasures. Two more scenarios were evaluated: Scenario 2 represents malware that spreads swiftly over the network, and Scenario 3 represents malware that is more subtle, as shown in Tables 6 and 7, respectively. In these two scenarios, the following conditions apply: accumulated reinfections refer to the sum of the infected vector, immunized nodes entering the network since $p = 0.1$.

As seen in Tables 6 and 7, the number of accumulated infected nodes and maximum infected peaks at scenario 3 decreased in comparison to scenario 2 for all values of $S(0)$, which is due to the differences in the initial infection rate for both scenarios. Thus, we can affirm, from an offensive perspective, that a high infection rate would allow the malware to replicate quicker across the network, resulting in the largest number of accumulated infected nodes and maximum infected peak. It is important to note in Table 6 that scenario 2 with $\xi = 1$ may represent the availability of efficient containment countermeasures, which would decrease the infection rate β faster. This contrasts with the results of infected nodes obtained for the same scenario 2 $\xi = 0.5$, where there is a decrease in the number of accumulated reinfections and the infected peak.

Moreover, as shown in Table 7, when ξ takes small values like 0.15, it is evident that it allows an increase in the number of accumulated reinfection, even having a $\beta_0 = 0.3$. Now, since small values of ξ may represent a vulnerable network with inaccurate containment countermeasures and $p = 0.1$ in scenarios 2 and 3, we can say, from a defensive perspective, that it is desirable to have the lowest number of new nonimmunized devices in order to stop the malware spread and overcome the attack.

In addition, Figure 7(a) shows the simulation results for scenarios 2 and 3, allowing to note that the combination of $\xi = 0.5$ and $\beta_0 = 1$ gets the biggest values of infected nodes, in comparison with other options. It is also important to observe that a model with $\beta_0 = 1$ exposes a substantial reduction in the number of the infected nodes after reaching the infection peak. Also, Figure 7(b) depicts the influence of optimal disconnection and vaccination tactics on the rate of drop in the number of infected hosts. Last but not least, Figure 7(c) shows the situation where only the combined disconnection and vaccine techniques are able to contain the attack within the specified time frame, as opposed to other strategies, which require values outside the range examined. We can also observe the influence of μ, which despite ϕ probably being big, gets the lowest number of infected nodes.

Table 5: Results for scenario 1 representing malware spreading in a network with $\beta_0 = \xi$ and $\beta_0/2 = \xi$.

$S(0)$	$\beta_0 = \xi$		$\beta_0/2 = \xi$	
	Max accumulated infected	Max infected peak	Max accumulated infected	Max infected peak
34	0.98	1	0.993	1
54	0.981	0.999	0.997	1
199	0.984	0.978	1	1
499	0.987	1	0.993	1

Table 6: Results for scenario 2 representing malware spreading in a network with with $\beta_0 = 1$, $\Lambda = 10$, and $p = 0.1$.

$S(0)$	$\beta_0 = 1$ with a simulation time = 90			
	$\xi = 0.5$		$\xi = 1$	
	Accumulated re/infections	Infected peak	Accumulated re/infections	Infected peak
34	25 198	8	22 149	6
54	27 135	11	23 786	9
199	42 924	40	36 833	31
499	76 577	102	65 361	79

Table 7: Results for scenario 3 representing malware spreading in a network with $\beta_0 = 0.3$, $\Lambda = 10$, and $p = 0.1$.

$S(0)$	$\beta_0 = 0.3$ with a simulation time = 90			
	$\xi = 0.15$		$\xi = 0.5$	
	Accumulated re/infections	Infected peak	Accumulated re/infections	Infected peak
34	19 881	7	15 589	5
54	21 778	9	17 011	6
199	37 046	36	27 189	22
499	69 885	95	49 986	56

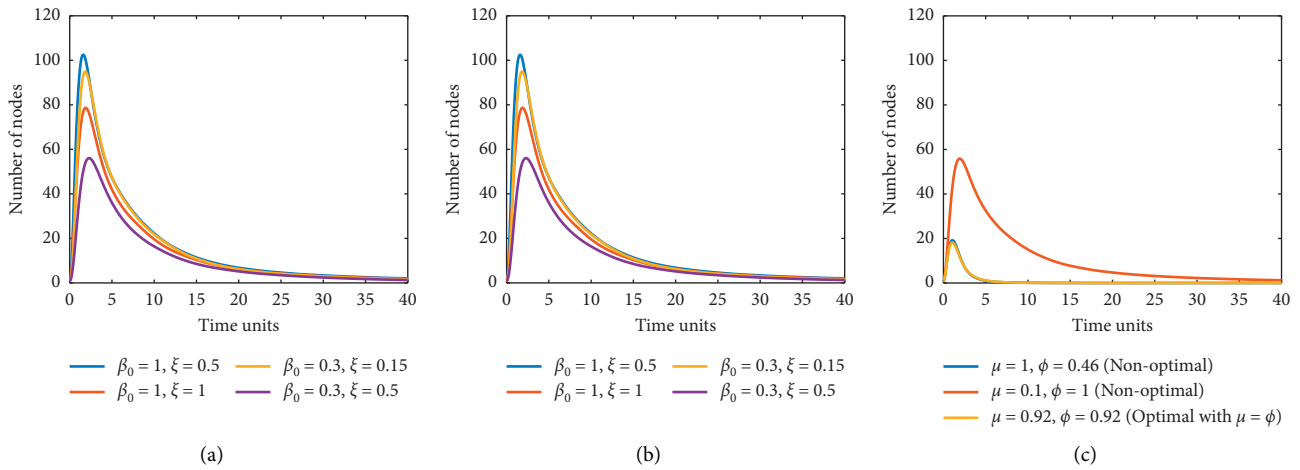

Figure 7: Results for scenarios 2 and 3 with $S(0) = 499$. (a) Infected nodes for different combination of parameters β_0 and ξ. (b) Infected nodes for different combination of parameters β_0 and ξ and with $\Lambda = 10$, $p = 0.1$. (c) Infected nodes for different combination of parameters μ and ϕ with $\Lambda = 30$, $p = 0.5$.

5.4.2. Operation under Attack. In this second analysis, the capacity of a network of computers to continue operating in the event of an attack is evaluated, especially when a certain number of new devices are required to be incorporated into the network with the aim of having connectivity. Here, the death of a node from other causes (μ) can represent a simple network disconnection. Both the appropriate values of μ and the vaccination rate (ϕ) could be determined through our proposed model in order to allow new devices access to the network while ensuring that the attack can be controlled, i.e., $I(t)$ tends to zero. In circumstances when these calculations are not possible, values will be specified as Out of Range (OoR).

This analysis is done through scenarios 4, 5, and 6, with the settings and obtained results depicted in Tables 8–10. All these scenarios have simulation times of 30 and 90 time

units, with $p = [0.1, 0.5]$, and $\Lambda = [10, 30, 60]$. If the devices belonging to the network are susceptible to the malware under analysis, vaccination and device disconnection methods are required to control the attack, as one might assume. However, under these conditions, as shown in Tables 8–10, when the number of new devices is large enough ($\Lambda = 60$), no measure is able to keep the onslaught under control. Thus, from a defensive perspective, this situation may be convenient to regulate the addition of new nodes to the network through some techniques such as Network Access Control (NAC) server.

In turn, from Table 8, we can observe that in order to make $I(t) \longrightarrow 0$, a minimum of $p = 1.05$ is required, which is out of range but very close to the maximum of 1. Also, combining vaccine and disconnection strategies yields the best outcomes, as seen in Table 9. This may be an optimal strategy if currently connected devices may be disconnected for a long period. Table 10 also demonstrates that relying solely on vaccination tactics to confine malware does not yield good enough results, as this can only be done over a lengthy period of time and with a small number of susceptible devices.

6. Use Case: Analysis of Malware Propagation

This section shows how different real cases of malware campaigns have evidenced behaviors that prove the suitability of an appropriate selection of infection, recovery, and loss of immunity rate variables along time, as is proposed by MalSEIRS.

6.1. Analysis of the Infection Rate. It is relatively common that several malware campaigns share a similar behavior in their infection process: attacks quickly infect a large number of unsuspecting hosts, and then the contagion rate gets dramatically reduced when a peak is reached. This is due to the fact that the accumulated number of infected nodes generally follows a logistic growth [27–29]. Section 4.1 indicated that network congestion may be one of the reasons for the reduction of the contagion rate, which was effectively seen in Slammer, one of the fastest computer worms in the history of the Internet, which in 2003 spread so rapidly that the countermeasures taken became ineffective [27].

As explained in Ref. [27], although the payload of Slammer was benign, it infected more than 90% of vulnerable hosts worldwide within 10 minutes, causing a significant disruption to financial, transportation, and government institutions. Its speed was unlike anything the Internet had ever witnessed before: in just 3 minutes, the worm performed more than 55 million scans per second. In addition, the Slammer scanning technique was so aggressive that it quickly interfered with its growth. When Slammer had reached its maximum scan rate, it was forced to decrease it because significant portions of the network had insufficient bandwidth to accommodate more growth. In this case, the behavior of Slammer itself regarding the network use caused a change in the infection rate of the malware through the different stages of the propagation.

Table 8: Results for scenario 4 aimed to get minimum μ required to $I(t) \longrightarrow 0$.

Λ	Simulation time = 30		Simulation time = 90	
	$p = 0.1$	$p = 0.5$	$p = 0.1$	$p = 0.5$
10	0.27	0.56	0.04	0.52
30	0.52	1.07 (OoR)	0.33	1.05 (OoR)
60	0.77	1.52 (OoR)	0.59	1.5 (OoR)

Table 9: Results for scenario 5 aimed to get minimum μ and ϕ required to $I(t) \longrightarrow 0$, assuming $\mu = \phi$.

Λ	Simulation time = 30		Simulation time = 90	
	$p = 0.1$	$p = 0.5$	$p = 0.1$	$p = 0.5$
10	0.29	0.53	0.13	0.5
30	0.51	0.94	0.36	0.92
60	0.71	1.32 (OoR)	0.56	1.29 (OoR)

Table 10: Results for scenario 6 aimed to get minimum ϕ required to $I(t) \longrightarrow 0$.

Λ	Simulation time = 30		Simulation time = 90	
	$p = 0.1$	$p = 0.5$	$p = 0.1$	$p = 0.5$
10	4.16 (OoR)	7.22 (OoR)	0.23	3.14 (OoR)
30	11.82	20.74	1.71	10.4
60	23.21 (OoR)	40.96 (OoR)	3.91 (OoR)	21.28 (OoR)

Other possible causes of the decrease in the infection rate have their origin in different defensive measures that are applied as a reaction to an attack. Possible use cases can be studied by analyzing its campaigns in 2020 and its modus operandi, as can be seen in Ref. [30]. Emotet was first spotted in 2014, and over the past decade, it became one of the most dangerous pieces of malware on the Internet until an operation in early 2021 by Europol and Eurojust dismantled its infrastructure [31].

Emotet spreads mostly through e-mail and employs social engineering tactics to get victims to open or download a malicious file [32]. Furthermore, Emotet has the ability to harvest e-mail address books, message headers, and body material, as well as send phishing attacks from infected systems. In the case of the news portal Heise, containment measures were not implemented until the number of infected nodes had risen to the point that communications to Emotet servers were being intercepted by the firewall. The most crucial step, which Heise applied, was to remove compromised devices from the network. This prevented the spread of the malware from infected hosts to other devices. Thus, if the malicious e-mail was sent by a Trusted Third Party (TTP), the best course of action is to quarantine all communications with them until such TTP can clean up the infection on their end, as recommended in Redscan Whitepaper about Emotet [31]. Notifying users of the ongoing attack and advising them to be mindful of opening attachments is another strategy to minimize this rate as time goes on. Then, as time allows for a response by the security team, defensive isolation measures, as well as educating people who are vulnerable to be cautious with the files they run, allow Emotet campaigns to vary their propagation effectiveness, as supported in Section 4.1.

6.2. Analysis of the Recovery Rate. The recovery process has a similar behavior between different malware campaigns composed of elements like antivirus signatures updates, security patches, or machine disconnections. The security measures grow as the number of infected nodes increases, i.e., generally, after a notable outbreak is present, actions to fix the infected machines appear and become relevant [33, 34].

The ransomware WannaCry was one of the most famous attacks that emerged in 2017. Its worm-capabilities made WannaCry really contagious and, in a single day, it was able to infect over more than 200,000 computers in more than 150 countries [5]. WannaCry took advantage of a Windows vulnerability known as MS17-010 to propagate by itself. Also, the attackers behind WannaCry were able to exploit such vulnerability thanks to the EternalBlue code [35]. Paying the ransom that hackers were asking to get the encrypted files back was one of the ways to get out of this problem, and effectively around 0.06% of the victims paid such an amount to get over the attack [36].

Microsoft presented a patch for the vulnerability MS17-010 two months before the infection, in March 2017. However, it was not installed on most of the computers, making the scale of the attack rather big. As WannaCry was infecting Windows versions from Windows XP to Windows 8.1 [37], Microsoft decided to publish a patch for those versions [36], becoming the first official response from Microsoft to help prevent the attack. Later, two British computer security researchers (Marcus Hutchins and Jamie Hankins) found a kill switch for WannaCry, consisting of buying a domain that was being requested by the malware in order to avoid encrypting the files at the victim host, and it was registered on May 12, 2017 [38]. Nowadays, there exists some kind of tools such as Norton or Quick heal that help to delete WannaCry even if they do not help to get the encrypted files back. All these responses to the attack support our consideration that the recovery rate will be variable along time since the number of infected nodes increased while antivirus, network, and host-based signatures of the malware were unknown. Then, as signatures were discovered and shared with the community, new containment measures were deployed, increasing the recovery rate.

Nonetheless, a kill switch is not always present with the aim of stopping an attack. It is the case of Emotet, introduced in Section 6.1, which is also able to propagate another kind of malware [39], such as Qakbot, a banking trojan with worm capabilities.

Emotet also avoids the detection based on signatures [39] and has anti-sandbox and anti-VM properties [40], which hinders its detection and also the recovery. Emotet attacks are known for their magnitude, as witnessed in the case of Heise or Fabrikam [41] (Fabrikam is the pseudonym given by the DART Microsoft Detection and Response Team). The security measures for recovery were taken after the infection outbreaks, i.e., the infection in Fabrikam shut down the entire business network, and after asking DART to support, Fabrikam decided to deploy solutions of Defender Advanced Threat Protection, Azure Security Scan, and other Microsoft malware detection tools. Moreover, after they identified the virus, DDoS (Distributed Denial of Service), and phishing emails, they found out that it was needed to make changes in the network infrastructure to stop Emotet. DART entered the network and implemented asset controls, creating buffer zones separating the assets with administrative privileges in the environment, and they also uploaded the antivirus signatures. In addition, the buffer zones helped to contain the virus and to delete it with antivirus [41]. Nevertheless, it is always recommendable to disconnect the infected machines from the network to prevent propagation and reduce damages. Indeed, the recovery came after the infection outbreaks, i.e., after a considerable amount of nodes was infected. However, the security measures would have been more effective to detain the malware campaign if they had been applied in the early stages of the infection.

6.3. Analysis of the Immunity Loss Rate. The main reason, but not the only one, why an oscillating immunity loss rate is considered is the threat posed by polymorphic malware. This is a type of malware that constantly changes its identifiable features in order to evade detection. Thus, signature-based detection solutions will not be able to recognize a sample as malicious because of the malware evolving characteristics.

This is the case of Emotet, which was previously described in Section 6.1. The polymorphic packer used by Emotet produces packed samples that vary in size and structure. At runtime, the encrypted loader is unpacked, and the unpacking code then executes the freshly unpacked code. Packers aid malware developers to avoid being discovered by making static binary analysis more complex. These packer changes make profiling a fresh sample solely based on the footprint of the packer extremely unfeasible, posing a significant challenge for anti-virus software that attempts automatic unpacking as part of its analysis. Thus, malware developers are repackaging their program as a unique executable for each potential victim in order to circumvent detection by signatures. As a result, standard antivirus software may fail to detect the most recent and complex attacks, putting enterprises in danger. This is a nontrivial problem since these viruses are frequently associated with ransomware attacks soon after infection.

However, while the problems with some signature-based strategies in maintaining immunity against Emotet have been highlighted, this rate is expected to tend to zero. The AV-Test lab says that the first detection rate is only 25%, but within seven days, it rises to about 90%2. Hence, as time goes by, these solutions become better at recognizing Emotet attacks and effectively immunizing devices.

Another feature of Emotet that makes it difficult for not only analysts but also antivirus tools is its heavily obfuscated code. Emotet possesses encrypted imports and function names that are deobfuscated and resolved dynamically at runtime. Emotet encrypted Command and Control (C2) data have also been saved in the data portion of HTTP POST requests delivered to the malware C2 servers since March 2019. Because most web proxies do not log the data section of HTTP requests by default, tracing Emotet C2 connections becomes more difficult [32].

Finally, Emotet receives updates via C2 servers. This works in the same way that a PC operating system upgrades do, and it can happen seamlessly and without warning. This allows the attackers to install new versions of the malware, as well as other infections, such as banking Trojans, and operate as a dumping ground for stolen data, such as financial credentials, sensitive credentials, such as usernames and passwords, and emails. Thus, it is possible that, even though a device is protected against a variant of Emotet, another infected device may receive an update of the said malware and try to transmit it to the other nodes, rendering said immunity useless. As a result of the Emotet capabilities (polymorphism, obfuscation, and constant updates from C2 server), Emotet is one of the clearest examples of a variable immunity loss rate.

6.4. Offensive and Defensive Strategies. After validating the proposed MalSEIRS model in Section 5 and analyzing real case scenarios in Sections 6.1–6.3, some defensive recommendations are contributed next:

(1) It is always better to try to increase preventive measures against a possible attack as previously discussed in Section 6.3, i.e., increase ϕ (real-time immunization rate) and reduce p (birth susceptibility rate).

(2) In the event of an attack, and if it is not possible to force an increase in the recovery rate $\gamma(t)$, it might be more feasible to disconnect the infected machines from the network and carry out a reinstallation directly, which would imply to increase the machine unavailability rate μ by other causes.

(3) To prevent damage in a network with infected nodes, it is better not to allow new machines to enter the network ($\Lambda = 0$), centralizing the defensive efforts in the current attacked nodes and avoiding the damage that might spread.

Likewise, a number of offensive strategies derived from our work are also presented as follows:

(1) From an attacker point of view, it is important to note that malware should not be too lethal: if it makes infected machines inaccessible too quickly, the malware would not be able to spread. This implies managing a small α, as it happened with Emotet: it manages to infect a whole network, as we saw in Section 6.2. However, the malware could be too destructive if the attacker wants to centralize an attack, i.e., cause a denial of service in a specific server.

(2) In the case of a virus, the execution rate σ may be decreased by promoting cybersecurity awareness between the users, as it is highly dependent on the users. Nevertheless, for another kind of malware, it would be difficult to get the execution rate σ under control, e.g., in the case of a Trojan with worm capabilities, an attacker could control trigger actions such as tampering of DLL's libraries, registry keys or exploitation of OS vulnerabilities that allow the autonomous execution of the malware, impacting in this way the execution rate.

(3) Offensively, a high initial rate of infection (β_0) may be convenient if the attacker wants to impact many devices in a short time, e.g., a DDoS campaign that aims to leave the network inoperable. In practice, this infection rate will depend on the countermeasures that the victim deploys as response to the attack, such as the isolation of infected devices, which are reflected in the parameter ξ that controls the decrease of the infection rate, as indicated in Section 6.1. A victim not having countermeasures to deploy may be represented by a parameter ξ close to zero.

(4) As previously discussed, in order to carry out an effective attack as in the case of Emotet (see Section 6.3), malware must be altered each time, since it becomes widely identified and removed by security tools. Such behavior would imply mathematically that the constant a in equation (8) approaches zero, as it means that the loss of immunity rate is kept a long time, i.e., the malware attack may persist.

7. Conclusions and Future Work

The emergence of the Internet, along with its massive amount of wide-ranging applications, has brought innumerable benefits, not only in the personal sphere but also in the field of business productivity. However, as everything is migrating to the cloud, the Internet has become the target of numerous attacks carried out by black hat hackers. Thus, the need arises to develop models that allow anticipating a possible attack in order to conduct and enforce pertinent defensive strategies, so as to mitigate its disruptive and harmful actions. Also, in the cyberwar arena, it can be beneficial to determine how a computer virus/worm should behave for a successful attack.

In this paper, we developed MalSEIRS, a new dynamic model to predict the propagation of malware within a network based on the SEIRS epidemiological model. Our novel model identifies the limitations of the SIR and SEIRS models regarding their constant rates that do not allow us to express the change in these parameters caused by external factors, and therefore we defined new equations dependent on the time to represent the infection rate, recovery rate, and loss of immunity rate, in the context of a dynamic computer network. Besides, offensive and defensive strategies were analyzed in terms of the proposed MalSEIRS model, so actionable information may be used as part of playbooks of Computer Security Incident Response Teams (CSIRT).

A profound analysis of our model, including conducting comprehensive experiments to compare it against other relevant models (such as SEIRS, SLBQR, and SIQVD), helped us to shed light on both successful defensive and offensive strategies. From a defensive perspective, it is determined that it is always better to increase preventive measurements against a possible attack, such as increasing antiviruses instances in real time, or in the case of new

devices entering the network, guaranteeing that they are as less susceptible to malware as possible and restricting the number of them. In the worst cases, it may even be worthy to carry out a complete disconnection of all devices on the network. Likewise, in the offensive plane, the most important outcome for a lasting attack is that the malware should not be lethal since an infected and malfunctioning device will not be able to further spread a malicious file. In addition, from an offensive perspective, it is advantageous that the malware "outbreaks" very quickly and suddenly, i.e., that it manages to infect as many devices as it can initially, limiting the time for a proper defensive reaction.

Finally, even though our proposed MalSEIRS model may exhibit oscillations in the number of susceptible and recovered nodes, particularly at the beginning of the attack, it manages to stabilize quickly. Furthermore, the model is robust in the sense that small changes in the parameters do not cause substantial changes in its behavior. As future works, we plan to develop specific versions of the proposed model for different malware families, considering its different capabilities in terms of propagation, persistence, reinfection, among others.

Data Availability

The data and code used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Universidad del Rosario (Bogotá) through the project "IV-TFA043 - Developing Cyber Intelligence Capacities for the Prevention of Crime." This work was also supported by an FPU predoctoral contract granted by the University of Murcia and by a Ramón y Cajal research contract (RYC-2015-18 210) granted by the MINECO (Spain) and co-funded by the European Social Fund.

References

[1] P. Nespoli, D. Useche Pelaez, D. Díaz-López, and F. Gómez Mármol, "Cosmos: collaborative, seamless and adaptive sentinel for the internet of things," *Sensors*, vol. 19, no. 7, 2019.

[2] D. Díaz López, M. Blanco Uribe, C. Santiago Cely et al., "Developing secure iot services: a security-oriented review of iot platforms," *Symmetry*, vol. 10, no. 12, 2018.

[3] C. Sánchez, C. Aguado, D. Díaz-López, and J. García, "Using reverse engineering to face malware," *Revista Ingenieria Solidaria*, vol. 15, no. 2, pp. 1–26, 2019.

[4] E. d. O. Andrade, J. Viterbo, C. N. Vasconcelos, J. Guérin, and F. C. Bernardini, "A model based on lstm neural networks to identify five different types of malware," *Procedia Computer Science*, vol. 159, pp. 182–191, 2019.

[5] A. Maxat and V. Vassilios, "Wannacry ransomware: analysis of infection, persistence, recovery prevention and propagation mechanisms," *Journal of Telecommunications and Information Technology*, vol. 1, pp. 113–124, 2019.

[6] S. Joseph, *Cybersecurity Legislation and Ransomware Attacks in the united states, 2015-2019*, 2021, https://digitalcommons.odu.edu/cgi/viewcontent.cgi?article=1134&context=gpis_etds.

[7] H. H. Weiss, *The Sir Model and the Foundations of Public Health*MAT, Barcelona, 2013, https://api.semanticscholar.org/CorpusID:4931923.

[8] M. Krzywinski, N. Altman, O. N. Bjørnstad, and K. Shea, "The seirs model for infectious disease dynamics," *Nature Methods*, vol. 17, no. 46, pp. 557–559, 2020.

[9] M. Eilstrup-Sangiovanni, "Why the world needs an international cyberwar convention," *Philosophy & Technology*, vol. 31, no. 3, pp. 379–407, 2018.

[10] C. Gan, Q. Feng, X. Zhang, Z. Zhang, and Q. Zhu, "Dynamical propagation model of malware for cloud computing security," *IEEE Access*, vol. 8, pp. 20325–20333, 2020.

[11] N. Levy, A. Rubin, and E. Yom-Tov, "Modeling infection methods of computer malware in the presence of vaccinations using epidemiological models: an analysis of real-world data," *International Journal of Data Science and Analytics*, vol. 10, no. 4, pp. 349–358, 2020.

[12] R. P. Ojha, P. K. Srivastava, G. Sanyal, and N. Gupta, "Improved model for the stability analysis of wireless sensor network against malware attacks," *Wireless Personal Communications*, vol. 116, no. 3, pp. 2525–2548, 2021.

[13] J. D. Hernández Guillén, A. Martín del Rey, and R. Casado-Vara, "Security countermeasures of a sciras model for advanced malware propagation," *IEEE Access*, vol. 7, pp. 135472–135478, 2019.

[14] Y. Zheng, J. Zhu, and C. Lai, "A seiqr model considering the effects of different quarantined rates on worm propagation in mobile internet," *Mathematical Problems in Engineering*, vol. 2020, Article ID 8161595, 16 pages, 2020.

[15] Y. Yao, Q. Fu, W. Yang, Y. Wang, and C. Sheng, "An epidemic model of computer worms with time delay and variable infection rate," *Security and Communication Networks*, vol. 2018, Article ID 9756982, 11 pages, 2018.

[16] F. Abazari, M. Analoui, and H. Takabi, "Effect of anti-malware software on infectious nodes in cloud environment," *Computers & Security*, vol. 58, pp. 139–148, 2016.

[17] J. D. Hernández Guillén and A. Martín del Rey, "Modeling malware propagation using a carrier compartment," *Communications in Nonlinear Science and Numerical Simulation*, vol. 56, pp. 217–226, 2018.

[18] L. Feng, X. Liao, H. Qi, and H. Li, "Dynamical analysis and control strategies on malware propagation model," *Applied Mathematical Modelling*, vol. 37, no. 16, pp. 8225–8236, 2013.

[19] D. E. Useche-Peláez, D. Sepúlveda-Alzate, D. Díaz-López, and D. E. Cabuya-Padilla, "Building malware classificators usable by state security agencies," *Iteckne*, vol. 15, no. 2, pp. 107–121, 2018.

[20] P. Nespoli, F. Gómez Mármol, and J. Maestre Vidal, "A bio-inspired reaction against cyberattacks: ais-powered optimal countermeasures selection," *IEEE Access*, vol. 9, pp. 60971–60996, 2021.

[21] A. Fagioli, "Zero-day recovery: the key to mitigating the ransomware threat," *Computer Fraud & Security*, vol. 31, no. 1, pp. 6–9, 2019.

[22] P. Reddy Ganganagari, *Defining Best Practices to Prevent Zero-Day and Polymorphic Attacks*, Master's thesis, University of Alberta, Edmonton, Canada, 2021.

[23] J. Alrzini and D. Pennington, "A review of polymorphic malware detection techniques," *International Journal of*

Advanced Research in Engineering & Technology, vol. 11, no. 12, pp. 1238–1247, 2020.
[24] The Mitre Corporation, *MITRE ATT&CK®*, 2020, accessed, https://attack.mitre.org/.
[25] R. Hranický, B. Frank, O. Ryšavý et al., "What do incident response practitioners need to know? a skillmap for the years ahead," *Forensic Science International: Digital Investigation*, vol. 37, Article ID 301184, 2021.
[26] D. Moore, *The Spread of the Code Red Worm (Crv2)*, 2001.
[27] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the slammer worm," *IEEE Security & Privacy*, vol. 1, no. 4, pp. 33–39, 2003.
[28] S. Hosseini and M. A. Azgomi, "The dynamics of an seirs-qv malware propagation model in heterogeneous networks," *Physica A: Statistical Mechanics and its Applications*, vol. 512, pp. 803–817, 2018.
[29] D. Chumachenko, K. Chumachenko, and S. Yakovlev, "Intelligent simulation of network worm propagation using the code red as an example," *Telecommunications and Radio Engineering*, vol. 78, no. 5, pp. 443–464, 2019.
[30] C. Patsakis and A. Chrysanthou, *Analysing the Fall 2020 Emotet Campaign*, 2020, https://arxiv.org/abs/2011.06479.
[31] Emotet, *Emotet Is Down but its Legacy Remains: Lessons Learned*, Technical report, Redscan, London, U.K, 2021.
[32] Emotet, *A Technical Analysis of the Destructive Polymorphic Malware*, Technical report, Bromium, California, U.S, 2019.
[33] D. Díaz-López, G. Dólera-Tormo, F. Gómez Mármol, and G. Martínez Pérez, "Dynamic counter-measures for risk-based access control systems: an evolutive approach," *Future Generation Computer Systems*, vol. 55, pp. 321–335, 2016.
[34] P. Nespoli, D. Díaz-López, and F. Gómez Mármol, "Cyber-protection in iot environments: a dynamic rule-based solution to defend smart devices," *Journal of Information Security and Applications*, vol. 60, Article ID 102878, 2021.
[35] Tobias Stier and J. Greve, *An Analysis of Wannacry and Eternalblue*, University of Copenhage, Copenhage, Denmark, 2019.
[36] N. Kshetri and J. Voas, "Do crypto-currencies fuel ransomware," *IT professional*, vol. 19, no. 5, pp. 11–15, 2017.
[37] M. Akbanov, V. G. Vassilakis, and M. D. Logothetis, "Wannacry ransomware: analysis of infection, persistence, recovery prevention and propagation mechanisms," *Journal of Telecommunications and Information Technology*, vol. 1, pp. 113–124, 2019.
[38] P. Mackenzie, "Wannacry aftershock," vol. 1, Sophos, Abingdon, U.K, 2019.
[39] S. Kuraku and D. Kalla, *Emotet Malware -a Banking Credentials Stealer*, 2020, https://www.researchgate.net/publication/343681889_Emotet_Malware_-A_Banking_Credentials_Stealer.
[40] SophosLabs Research Team, "Emotet exposed: looking inside highly destructive malware," *Network Security*, vol. 2019, no. 6, pp. 6–11, 2019.
[41] in*Full Operational Shutdown"*, Technical report, 2020, https://www.microsoft.com/security/blog/wp-content/uploads/2020/04/Case-study_Full-Operational-Shutdown.pdf.