Cyber Network Resilience Against Self-Propagating Malware Attacks

Alesia Chernikova[1](✉), Nicolò Gozzi[2], Simona Boboila[1], Priyanka Angadi[4], John Loughner[4], Matthew Wilden[4], Nicola Perra[3], Tina Eliassi-Rad[1], and Alina Oprea[1]

[1] Northeastern University, Boston, MA, USA
chernikova.a@northeastern.edu
[2] University of Greenwich, London, UK
[3] School of Mathematical Sciences, Queen Mary University of London, London, UK
[4] PricewaterhouseCoopers LLP, Washington, DC, USA

Abstract. Self-propagating malware (SPM) has led to huge financial losses, major data breaches, and widespread service disruptions in recent years. In this paper, we explore the problem of developing cyber resilient systems capable of mitigating the spread of SPM attacks. We begin with an in-depth study of a well-known self-propagating malware, WannaCry, and present a compartmental model called SIIDR that accurately captures the behavior observed in real-world attack traces. Next, we investigate ten cyber defense techniques, including existing edge and node hardening strategies, as well as newly developed methods based on reconfiguring network communication (NodeSplit) and isolating communities. We evaluate all defense strategies in detail using six real-world communication graphs collected from a large retail network and compare their performance across a wide range of attacks and network topologies. We show that several of these defenses are able to efficiently reduce the spread of SPM attacks modeled with SIIDR. For instance, given a strong attack that infects 97% of nodes when no defense is employed, strategically securing a small number of nodes (0.08%) reduces the infection footprint in one of the networks down to 1%.

1 Introduction

Self-propagating malware (SPM) has become one of the top cyber threats in recent years. In 2016, Mirai [13] malware infected more than 600K consumer devices and launched a widespread DDoS attack targeting over 175K websites. The WannaCry [26] ransomware attack of 2017 affected more than 300K vulnerable devices in 150 countries in a few days, from entire healthcare systems to banks and national telecommunications companies. Worryingly, there have been reports of its re-appearance during the COVID-19 pandemic. Recently, ransomware attacks have increased significantly, with the emergence of new threats like Ryuk (2019) [14], PureLocker (2020) [39], and many others. SPM campaigns attempt to exploit vulnerabilities on specific ports by blending in with

V. Atluri et al. (Eds.): ESORICS 2022, LNCS 13554, pp. 531–550, 2022.
https://doi.org/10.1007/978-3-031-17140-6_26

legitimate traffic. Since blocking ports entirely is often not feasible, defending against SPM is particularly challenging. Machine learning (ML) techniques have been employed [33], with the goal of detecting the attack and taking reactive measures after the data breach has occurred. However, the performance of ML methods often degrades when there is a high number of false positives, which are hard to triage by human experts.

In this paper, we take a graph robustness perspective for *proactively protecting cyber networks against self-propagating malware attacks.* We study the problem of how to build cyber resilient systems and how to configure communication in cyber networks to prevent the spread of SPM attacks. Towards this ambitious goal, our first task is understanding and modeling the behavior of a well-known SPM malware, WannaCry, by using compartmental models that stem from epidemiology. We then turn our attention to methods for enhancing network resilience against these attacks. We model the topology of the network via the communication flows collected from real network traces, which we obtained from an industry partner. Our main insight is to analyze communication networks through the lens of graph robustness, an area that has been studied extensively in other applications (e.g., social-, information-, transportation-, and mobility-networks), but much less in cyber security.

On the SPM attack modeling side, we show that prototypical models for virus propagation (such as SIS and SIR [6]) do not fit the behavior of WannaCry well. We thus introduce a new model (SIIDR) that captures the behavior of self-propagating malware more accurately. In particular, our new model introduces a dormant state, in which the malware is installed in the system, but is not active for some interval (this is a common behavior observed in SPM, as well as advanced persistent threat attacks, which have been documented to be "slow-and-low" for months and sometimes years). We use real traffic logs generated by multiple variants of WannaCry to select the best fit model and estimate the parameters (i.e., transition rates) that best characterize WannaCry attacks.

On the defense analysis side, we perform an in-depth evaluation on several complementary topological-based defenses. We investigate a large number of defense techniques (10) based on various cybersecurity strategies such as node hardening, edge hardening, isolation, and reconfiguring communication. We evaluate their ability to increase network resilience to cyber attacks using two robustness metrics that have been shown to be accurate indicators of graph connectivity, and, implicitly, of the network's resilience to attack propagation [16]: (a) the spectral radius of a graph and (b) the relative size of the largest connected component in the graph. We also propose two new defensive methods: *NodeSplit*, which reconfigures the nodes with the largest number of incident edges by migrating half of their communication links to new nodes; and *Community Isolation*, which constructs communities in the graph and then strategically hardens edges that connect the communities to thwart the attack propagation. Hybrid strategies that combine NodeSplit with edge hardening (e.g., setting up firewall rules) are particularly successful at minimizing the spectral radius of a graph (our first robustness metric), achieving over 60% reduction on most of our graphs, with a small budget of only 50 split nodes and a fraction of 0.1 secured edges. Hybrid NodeSplit + Community Isolation strategies perform very well on the second robustness metric, being able to break down the largest connected component to less than 20% of its original size on most of the studied graphs.

To evaluate our defenses in realistic conditions, we use six real-world communication flow graphs collected from a large retail network. These graphs model application communication on well-known ports (22, 80, 139, 383, 443, 445) and are up to 620K nodes and 6.8 million edges. We thoroughly evaluate and compare the defense strategies in terms of their ability to reduce the spread of malware modeled with SIIDR and the budget required for slowing down the attack. Node hardening techniques are the most effective defenses over a wide range of attack scenarios and network topologies studied here, leading to a 20× decrease in the effective attack strength. This reduction results in substantial infection footprint minimization. For instance, given a strong attack that infects 97% of nodes when no defense is employed, strategically securing a small number of nodes (i.e., 50 nodes, which account for 0.08% of the nodes on port 22) reduces the infection footprint on port 22 down to only 1%.

We summarize our contributions below:

- We propose and evaluate SIIDR, a compartmental attack model that captures the behavior of SPM accurately. We use real SPM traffic logs from WannaCry to estimate the attack parameters.
- We perform an in-depth evaluation of 10 defense techniques and compare them using two graph robustness metrics.
- We introduce two novel defenses: NodeSplit (to reconfigure communication of top-degree nodes) and Community Isolation (to harden edges between communities), and show their effectiveness particularly in hybrid strategies.
- We evaluate the effectiveness of various defense strategies against SPM attacks, using six large real-world communication graphs.
- We provide recommendations on the effectiveness and cost of multiple defenses to inform network operators on various proactive defense options against SPM attacks. Our open-source code is available on GitHub [18].

2 Problem Statement and Background

In a recent survey, Freitas et al. [16] note that the study of graph vulnerability and robustness is still nascent in cybersecurity. Existing research includes modeling lateral attack movement between computers and analytical studies of interdependent spatial networks [9,10,15]. However, they point out the need for additional work in several directions, including comprehensive evaluations of various attack and defense scenarios. Our work directly addresses this need, and is, in that sense, particularly timely. We study both facets of building cyber resilient systems, attacks and defenses, in an integrated and complementary way.

In cybersecurity, eliminating or mitigating vulnerabilities is achieved through "system hardening" [32] and depending on where or how protection is applied, it can refer to network hardening, server hardening, operating system hardening, application hardening, etc. Reducing the "attack surface" consists in addressing known vulnerabilities via changing passwords, removing unused services, applying security patches, closing open network ports, configuring firewalls and setting intrusion-detection systems. On the theoretical side, most of the previous techniques for increasing network robustness are based on classic mathematical epidemiology results, which link the spreading of a virus of a graph with its spectral radius [7,36]. The epidemics dies out quickly if the spectral radius is less than a threshold, which depends on the virus propagation method. Hence, topological changes are employed to bring the spectral radius below this threshold.

Threat Model: We aim to design efficient defense strategies that increase network robustness against SPM attacks. We consider cyber networks such as enterprise networks, data center networks, or cloud systems, in which communication flows between nodes can be modeled as a graph. We assume that the malware first compromises one victim machine on the network ("patient zero"), after which it spreads to other vulnerable machines within the network over a specific protocol, e.g., HTTP (port 80), SSH (port 22), SMB(port 445), etc.

We derive a realistic attack model (namely, SIIDR) and its parameters by running an actual WannaCry attack, under homogeneous mixing assumptions [46]. Homogeneous mixing models imply that all hosts have identical rates of infection-spreading contacts. Our attack experiments were carried out within a local subnet, where such an assumption is valid. For WannaCry modeling, we analyze Zeek logs collected at the border of the monitored subnet. We assume that these logs are not compromised by the attacker.

To design and evaluate defense strategies, we use communication data of a large retail network. We assume that the network traffic has not been compromised, and, thus, the logged connections can be used to derive an accurate graph representation of the network communication topology.

Challenges: Building cyber-resilient systems is challenging for multiple reasons. First, realistic modeling of actual attacks is difficult, due to the limited availability of attack traces, and the ethical considerations that prevent us from recreating known attacks in real-world networks. Second, the continual evolution of attacks that attempt to avoid detection requires innovative proactive measures that are able to counter a wide range of potential threats. Third, building resilient infrastructures is budget-dependent both in terms of infrastructure and software updates, as well as human effort. Careful assessments of complete eradication strategies versus mitigation (containment) methods are necessary to establish real-world feasibility of the defenses.

3 SPM Modeling

Infectious disease research has inspired the study of mathematical models for malware propagation, in part due to the similarities in the behavior of human viruses and computer viruses. Many of these epidemiological models are compartmental. That is, the population is divided into states (a.k.a. classes), based on the health status of individuals and the type of disease [6]. Examples include susceptible (S), exposed (E), infectious (I), quarantined (Q), and recovered (R), to name a few. In this study, we are investigating the specific case of self-propagating malware. All types of self-propagating malware have one defining characteristic: once a host becomes infected, it starts probing other computers on the Internet randomly, with the goal of spreading the infection widely. This type of behavior guides the mathematical modeling of SPM.

We use real-world attack traces to model self propagating malware, and derive compartmental models and parameters that closely fit actual attack propagation data. Model fitting for deriving best models and their parameters from data has been widely used in the study of infectious diseases [22,40], but less so in the modeling of computer viruses. To the best of our knowledge, we are the first to model self-propagating malware based on real attack traces to find the model and parameters that most accurately describe a real attack. In the remainder of this section we describe the data, selection methodology and results that identify the model that captures SPM behavior most precisely.

WannaCry Data: We select WannaCry malware as a representative self-propagating malware attack, which can be configured with multiple parameters to generate a range of propagation behaviors. As shown by Ongun et al. [33], other SPM malware such as Mirai, Hajime, and Kenjiro follow similar propagation patterns, and our attack modeling will likely generalize to other attacks. We set up a virtual environment featuring the EternalBlue Windows exploit that was used in the 2017 WannaCry attack. External traffic is blocked in order to ensure isolation of the virtual environment. Initially, one of the virtual machines is infected with WannaCry, and then the attack starts spreading, as infected IPs begin to scan other IPs on the same network. We identified two characteristics that WannaCry uses to control its spread: 1) the number of threads used for scanning, and 2) the time interval between scans. We conducted multiple experiments by running WannaCry with different characteristics and collecting log traces with Zeek network monitoring tool.

Epidemics Reconstruction: We use the WannaCry traces to study the malware behaviour and reconstruct the epidemics. The start and end time of the epidemics for each WannaCry variant is given by the first malicious attempt, and the last communication event, respectively. Hence, the first IP trying to establish a malicious connection represents “patient zero”, and an IP trying to establish at time t a malicious connection for the first time is considered infected at time t. Based on the WannaCry traces, we make the following observations:

- Distribution of Δt intervals between attacks: We observe that the Δt between two consecutive attacks from the same infected IP is not fixed. This heterogeneous distribution of Δt intervals between subsequent attacks from the same infected IP introduces the idea of a $I \leftrightarrow I_D$ dynamics, where I represents the infectious state and I_D represents the dormant state.
- Distribution of Δt intervals between last attack and end of trace: The Δt between the last attack from an infected IP and the end of observations is quite large. The non-zero, high-valued and heterogeneous distribution of Δt between last attack from an infected IP and the end of observation time supports the idea of a $I \to R$ dynamics, with R being the recovered (previously infected, but not infectious anymore) state.

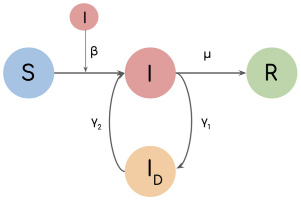


Fig. 1. Compartmental structure and transitions of the SIIDR model. Susceptible nodes (S), exposed to infected IPs, acquire the infection with rate β. Infected nodes transition with rate γ_1 or recover with rate μ. From infected dormant state (I_D) nodes transition back to I with rate γ_2.

Model Description: In accordance with the behavior observed from WannaCry traces, we propose SIIDR, an extension to the SIR model [6] that includes the infected dormant state I_D. While other models like SEIR or SEIRS have been used to model the spread of malware [38], SIIDR more closely explains the behavior observed of WannaCry. Furthermore, a model like SEIRS would not make sense, given that Recovery means patching of the EternalBlue vulnerability. Once the operating system is updated (patched), it will not become susceptible again to the same vulnerability. Figure 1 presents the transition diagram corresponding to the SIIDR model. A node that is infected may either recover with rate μ, or move to the dormant state with rate γ_1. From the dormant state, it may become actively infectious again with rate γ_2. We assume a homogeneous mixing model [46], that is, every node is potentially in contact with all the others. This is a good approximation because the WannaCry attack experiments were run within a subnet, where every node was able to scan every other internal IP within the same subnet. The system of differential equations that describe the dynamics of SIIDR, the derivation of the basic reproduction number of the model and discussion of stability are presented in a companion paper [11].

Model Selection and Parameter Estimation: To determine which model is the best fit, we compare common epidemiological models SI, SIS, SIR with SIIDR using the Akaike information criterion (AIC) [1]. The best fit is the model with the minimum AIC score. The AIC scores for all candidate models are presented in Table 1. We observe that the SIIDR model has the lowest AIC scores overall, except for wc_1_20s (on which all models have high AIC). For instance, the AIC score for the SIIDR model for variant wc_4_500ms is as low as -166, while the AIC scores for SI, SIS, and SIR models are 277, 76, and -45, respectively. Hence, SIIDR fits the WannaCry dynamics better than the other candidate models.

We use Sequential Monte Carlo (SMC) [24,27] to approximate the posterior distribution of rates ($\beta, \mu, \gamma_1, \gamma_2$) that fit the actual WannaCry data. The WannaCry variants with less than 20% infected IPs (i.e., 7 infections) were excluded due to insufficient samples to generate accurate models. Table 2 lists the mean values of SIIDR parameters posterior distributions. `dt` is the simulation time step. We have one contact per `dt`, thus, the transmission probability over a contact-link

Table 1. AIC scores for each of the SPM models for different WannaCry variants. Each WannaCry variant is identified by two parameters: the number of threads used for scanning and the time interval between scans (i.e., wc_1_500ms uses 1 thread to scan every 500 ms).

WannaCry variant	**SI**	**SIS**	**SIR**	**SIIDR**
wc_1_500ms	583	143	114	**-126**
wc_1_1s	431	188	145	**-127**
wc_1_5s	683	163	143	**72**
wc_1_10s	462	197	53	**-92**
wc_1_20s	704	**559**	696	700
wc_4_500ms	277	76	-45	**-166**
wc_4_1s	222	160	107	**-55**
wc_4_5s	412	186	158	**-46**
wc_4_10s	513	94	-36	**-145**
wc_4_20s	606	76	11	**-117**
wc_8_500ms	375	101	18	**-147**
wc_8_1s	178	91	51	**-116**
wc_8_5s	149	104	-35	**-121**
wc_8_10s	253	74	-90	**-118**
wc_8_20s	387	164	173	**-89**

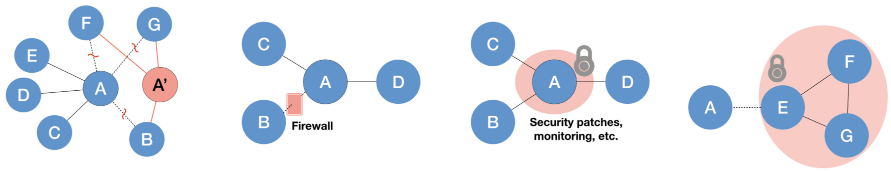
Table 2. Mean values from posterior distribution of SIIDR parameters, estimated with sequential Monte Carlo. For details, see our companion paper [11].

WannaCry variant	β	μ	γ_1	γ_2	dt
wc_1_500ms	0.10	0.06	0.76	0.04	0.09
wc_1_1s	0.11	0.07	0.71	0.07	0.06
wc_1_5s	0.37	0.52	0.27	0.44	0.16
wc_1_10s	0.12	0.06	0.75	0.05	0.09
wc_4_1s	0.14	0.07	0.75	0.08	0.05
wc_4_5s	0.12	0.07	0.76	0.07	0.07
wc_8_20s	0.13	0.09	0.74	0.08	0.07

equals β. We use these parameters to evaluate defenses in Sect. 7. For more details on model selection and parameter estimation, see our companion paper [11].

4 Defense Methodology

We model the network as a host-to-host communication graph. The nodes represent systems like computers, mainframe, peripherals, load balancer devices, etc. that communicate over TCP/UDP. An edge exists between two systems if they exchange network communication. We create these communication graphs from NetFlow data collected inside the organization. Putting defensive control on some of the nodes and edges in the network makes them inaccessible to the attacker. We define the `Attacker's Reachability Graph` (ARG) as the nodes and edges that an attacker has access to after defenses have been applied. We devise and study defense strategies from four different perspectives, as illustrated in Fig. 2 and summarized in Table 3. We describe these defenses next:



(a) Node Splitting (b) Edge Hardening (c) Node Hardening (d) Isolation

Fig. 2. Cybersecurity defense strategies: a) Node splitting – access to nodes B, F, G from node A is disabled, and a new node A' is created, with links to nodes B, F, G. b) Edge hardening via firewall rules. c) Node hardening via security patches and monitoring techniques. d) Isolation – the denser cluster comprised of nodes E, F, G is separated from the rest of the network through enhanced security measured at the boundary.

- **Node Splitting:** This newly proposed method reconfigures network communication by splitting and balancing top-degree nodes. It requires node addition and rewiring of edges, which can be implemented by reconfiguring communication via traffic management solutions available in data centers.
- **Edge Hardening:** This implies monitoring certain edges in the communication graph and corresponds to edge removal from ARG. It can be implemented via firewall rules for blocking certain flows, closing of some network ports, or adding rules to network intrusion detection systems (NIDS).
- **Node Hardening:** This implies monitoring certain nodes in the communication graph and corresponds to node removal from ARG. It can be implemented via system updates, security patch installation, and monitoring through endpoint agents.
- **Isolation:** This newly introduced method requires hardening of edges or nodes on community boundaries via separation of densely connected communities from the rest of the network.

Node Splitting: We introduce NodeSplit, a new algorithm that reconfigures network communication in order to increase resilience to SPM attacks. In our proposed approach, node splitting is targeted at top-degree nodes. If compromised,

Table 3. Defense strategies and associated topological changes.

Defense strategy	Algorithms	Topological changes in ARG	Cybersecurity measures
Node splitting	NodeSplit	Node addition and edge rewiring	Reconfiguring access rules
Edge hardening	MET [20], RandE	Edge removal	Firewalls, closing open ports
Node hardening	Degree, ENS, NB [43], RandN	Node removal	System updates, Security patches
Isolation	CI-Edge, CI-Node	Edge/node removal	Hardening at community boundary

these highly connected nodes become super-spreaders and potentially propagate the infection within a large portion of the network. By decreasing the size of their one-hop neighborhoods we can reduce the infection footprint. Breaking up the super-spreaders takes advantage of the highly heterogeneous topology of real networks [35].

NodeSplit is an iterative method, which progressively selects the current top-degree node and splits it in two: half of the links remain with the existing node, and half of the links are moved to a newly created node. The total number of peer nodes added equals the number of splits and depends on the available budget. In practice, the new node can be physical or virtual. A new physical node requires additional hardware, while a virtual node requires computational and memory resources on an existing server. Virtual machines are usually a cheaper and more flexible alternative to adding physical servers to the network.

Node splitting can be implemented in real networks by using existing traffic management frameworks. For instance, Microsoft Azure offers the Traffic Manager service [25], which allows routing of an application's traffic to different endpoints. Its randomized traffic splitting procedure distributes an application's traffic at random to multiple nodes and can be used directly to implement NodeSplit. Network traffic can also be reconfigured to different nodes using SDN-enabled policies [4].

Edge Hardening: Choosing the optimal edges to secure has been previously investigated in the context of spectral analysis, which refers to the study of the eigenvalues and eigenvectors of the adjacency matrix A of a graph. One particularly important metric in spectral analysis is the largest eigenvalue of the adjacency matrix, λ_1, because it captures communicability (i.e., path capacity) in a graph. Minimizing λ_1 was proven to effectively stifle the spread of a virus [7,36]. We investigate the following edge hardening methods:

1. RandE: Baseline that randomly chooses b edges from the graph to harden.
2. MET (short for Multiple Eigenvalues Tracking) [20], a well-known algorithm that was shown to successfully minimize λ_1, which utilizes *eigen-scores*[1] to estimate the effect of removing edges. MET iteratively chooses edges with highest eigen-score to remove until the budget b of edges is reached.
3. NodeSplit + MET hybrid strategy: This combined strategy consists of reconfiguring a small number of nodes with NodeSplit in order to make the edge distribution more homogeneous, followed by edge hardening with MET. We show in this study that this hybrid strategy is able to minimize the leading eigenvalue more than each of the two methods used separately.

Node Hardening: Targeted defense methods in epidemic spreading contain the virus by immunizing a small number of nodes. Which nodes to prioritize for immunization is a very relevant question both in social and computer networks, and has been the objective of numerous studies [5,35,43]. From a cybersecurity perspective, immunization corresponds to node hardening methods such as security

[1] The *eigen-score* of an edge $e = (i, j)$ equals to the product of the i-th and j-th elements of the left and right eigenvectors corresponding to λ_1, i.e. $u(i) \times v(j)$.

patches, system updates, and node monitoring via endpoint agents. The secured nodes become very hard to compromise and are removed from the `Attacker's Reachability Graph`. We analyze four methods for node immunization:

1. RandN: This is a baseline method, in which the nodes are chosen randomly throughout the network.
2. Degree: We progressively immunize the most connected nodes, as they contribute highly to the spread of infection.
3. ENS (Effective Network Size): Nodes with the highest effective network size are more likely to act like "bridges" between dense clusters [47] and monitoring them is likely to prevent attack spreading.
4. NB (Nonbacktracking Centralities): We explore a recent method [43], which uses the behavior of the spectrum of the nonbacktracking matrix. This method identifies nodes whose removal from the network has the largest impact on the leading eigenvalue of the nonbacktracking matrix.

Isolation: Communities are topological groups of nodes with dense internal connections. We design and explore community isolation strategies that work in two steps: First, community detection algorithms are used to identify communities. Many such algorithms are readily available – we use the well-known Infomap, Leading Eigenvector and the newer Leiden algorithm [45]. Second, community borders are secured, by hardening either nodes or edges. This translates in securing candidate bridge connections from the `Attacker's Reachability Graph`, in order to effectively detach the communities of a network and limit the spread of the attack. We study three isolation methods:

1. CI-Edge, in which all the edges on the borders are secured.
2. CI-Node, in which boundary nodes with highest degree are secured, in decreasing order of their degree.
3. NodeSplit + CI-Edge hybrid strategy. Reconfiguring a small number of nodes before performing community isolation improves the division of a network into modules. This is shown by an increase in modularity[2].

5 Experimental Setup

Datasets: We use an anonymized network flow dataset from an industry partner consisting of 3.4 million nodes and 50 million links. The Critical Watch Report of 2019[3] found out that 65% of vulnerabilities on TCP/UDP ports are linked to only 3 ports: 22, 443 and 80. Therefore, we extract and investigate the communication corresponding to a few representative ports: 22 (SSH), 80 (HTTP), 139 (SMB

[2] Modularity is defined as the number of edges falling within groups minus the expected number in the null model of the network (i.e., an equivalent network with edges placed at random) [29].

[3] https://www.newnettechnologies.com/study-finds-majority-of-port-vulnerabilities-are-found-in-three-ports.html.

Table 4. Topological data for the six port-based graphs studied.

Port	Number of Nodes	Number of Edges	Mean Degree	Density	Diameter	Avg Dist	Transitivity (global)	Transitivity (avg local)
22	60,825	333,797	11	0.0002	11	2.81	0.0001	0.181
80	287,156	1,833,568	13	0.00004	9	3.01	0.00003	0.058
139	1,912	9,532	10	0.005	9	2.93	0.000006	0.001
383	7,101	22,910	6	0.0009	7	3.45	0.001	0.236
443	620,096	4,437,255	14	0.00002	12	2.80	0.000002	0.023
445	317,031	6,832,418	43	0.0001	10	2.79	0.00001	0.058

over NetBIOS), 383 (HP data alarm manager), 443 (HTTPS), and 445 (SMB), described in Table 4. For each port, We construct undirected, unweighted, 3-core graphs (i.e., the maximal subgraph where all the vertices have degree at least 3). We note the wide range of graph sizes we are investigating, and also the small 'Avg Dist' (i.e., the mean vertex-to-vertex distance), which implies that any attack will spread fast within the network. For other properties illustrated here (Density, Diameter, Transitivity) we refer the reader to [30].

We discover that our graphs have power-law degree distribution: while the bulk of the nodes have small degree, there is a smaller number of nodes with degrees much higher than the mean value. This property plays a crucial role in devising best defense strategies and it has been shown to be key to explain the success of targeted immunization strategies.

Evaluation Measures: Let G be the original graph and G' the perturbed graph after applying the defense methods. We use the following evaluation measures, whose importance in quantifying a network's resilience to attacks has been pointed out in previous research [43]:

- **EigenDrop $\Delta\lambda$ – drop in the leading eigenvalue:** This metric captures the path capacity reduction within the graph. The leading eigenvalue characterizes the epidemic threshold [36] – i.e., the regime required for an epidemic to occur. Decreasing the leading eigenvalue of the graph essentially increases the epidemic threshold and enables stronger attacks to die out fast. The percentage drop in the leading eigenvalue λ is: $\Delta\lambda\% = 100 \times \frac{\lambda-\lambda'}{\lambda}$, where λ' is the leading eigenvalue of the perturbed graph G'.
- **Fragmentation σ – size of the largest connected component relative to the total graph size:** Let N be the size of the graph, and N_L the size of its largest connected component. We define $\sigma = N_L/N$ as the fraction of nodes contained in the largest connected component. The larger the number of nodes that can be reached by the attack, the more damage it can cause. Reducing σ enables attack surface reduction by containing the attack within smaller connected components and thus reduces the infection footprint.

6 Evaluation of Network Resilience

In this section we evaluate how successful are the four types of defenses at increasing the network resilience to SPM.

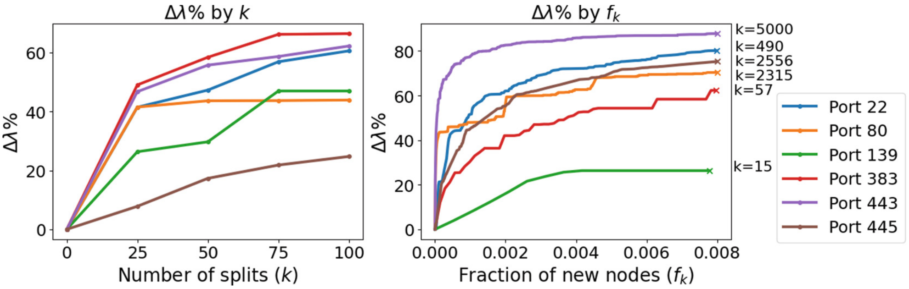


Fig. 3. NodeSplit: EigenDrop after node splitting, depending on the number of split nodes, k (left), and the fraction of split nodes, f_k (right) (higher is better). A 40–50% increase is observed on four of the graphs after only 50 splits (left). Steady increase for all ports as the fraction of split nodes increases (right).

Node Splitting: The NodeSplit method progressively selects the node with highest degree and transfers half of its edges to a newly created node to balance the number of connections. Thus, the graph becomes more homogeneous in terms of the degree distribution. We analyzed this transition using Alstott et al. [3]'s mathematical package and report the following findings. For the smaller graphs (ports 139, 383 and 22), a critical point has been reached before a fraction of 0.2 new nodes were added to the graph; after that, the distribution is closer to exponential, rather than power-law (indicated by the loglikelihood ratio). The trend towards a more homogeneous degree distribution also occurs for the larger graphs, however reaching the critical point requires significantly more splits.

Intuitively, the "communicability" in the graph is also decreased, as the fastest spreaders of information, the hubs, have reduced their number of connections. This is captured by a decrease in the leading eigenvalue. Figure 3 illustrates the EigenDrop, $\Delta\lambda\%$, depending on the number of new nodes (which equals the number of splits). On most ports, even a small number of splits leads to a substantial decrease in the leading eigenvalue. For example, just 50 node splitting operations are needed to achieve a 40–50% lambda drop for graphs whose mean degree is in teens. The long-term trend of a slow but steady EigenDrop increase is visible in Fig. 3 (right) for larger graphs, as we keep splitting more nodes. On the other hand, the largest connected component (i.e., our second evaluation metric) is generally preserved.

Edge Hardening: We compare RandE, MET [20], and a hybrid method that combines NodeSplit with MET in Fig. 4 in terms of EigenDrop. While MET reduces the leading eigenvalue by itself, the hybrid strategy provides a significant additional drop. For example, at 10% edges removed (x-axis), the hybrid strategy with 100 split nodes (NodeSplit-100 + MET) almost triples MET's EigenDrop on all graphs. MET is designed to work within a connected graph, and fragmentation (our second resilience metric) is generally negligible.

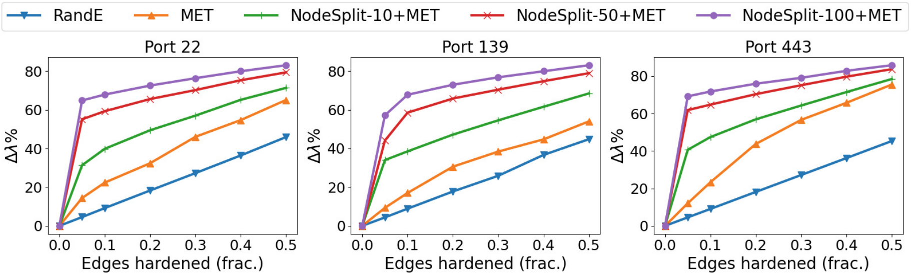


Fig. 4. Edge Hardening: EigenDrop depending on the percentage of removed edges. NodeSplit-k denotes a number of k splits (higher is better). Hybrid strategies provide additional increase in EigenDrop over MET on all graphs.

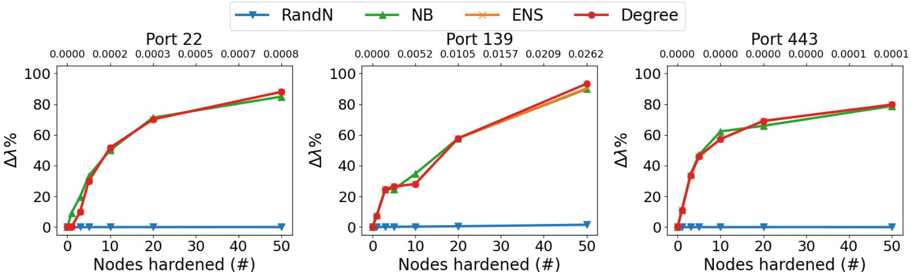


Fig. 5. Node Hardening: EigenDrop depending on the number of immunized nodes (higher is better). Degree, ENS and NB methods significantly outperform RandN. Top x-axis shows the *fraction* of immunized nodes, which is very small. Similar results on the other port-based graphs.

Node Hardening: Next, we evaluate the benefit of "immunizing" a small number of nodes to minimize the spread of malware using the following methods: RandN (baseline random node removal), Degree (top-degree nodes), ENS (top nodes according to the effective network size metric), and NB (non-backtracking centralities). Figure 5 illustrates the decrease in the leading eigenvalue after immunizing a number of nodes given by the budget. Interestingly, Degree, ENS and NB deliver very similar performance. This is due to the presence of a highly skewed degree distribution with a few heavily connected star-like nodes which also act as bridges: information needs to go through them to reach other nodes, resulting in large ENS and high path capacity (NB). As expected, uniform immunization strategies (RandN) are not effective, as they give the same weight to nodes with high and low connectivity.

How does the graph structure change when the immunized nodes are removed? Fig. 6 shows that Degree, ENS and NB exhibit a sharp drop in the size of the largest connected component. In contrast, RandN tends to select low-degree nodes that represent the vast majority of nodes, and whose removal has a low impact on connectivity. On ports 22 and 139, immunizing just 50 nodes with Degree/ENS is enough to break down the largest connected component.

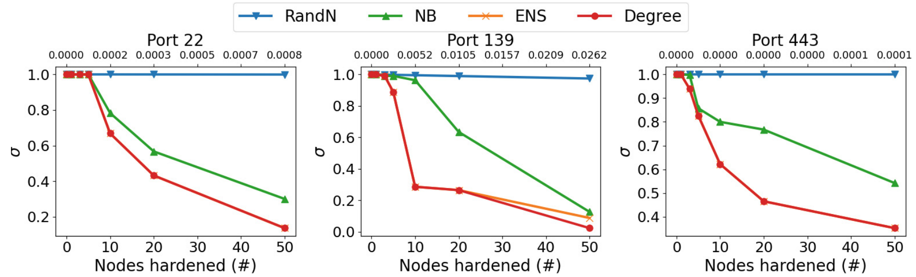


Fig. 6. Node Hardening: The size of the largest connected component in terms of the original graph size (σ) (lower is better). Immunizing just 50 nodes on all ports significantly breaks down the largest connected component with all of the methods. Top x-axis shows the *fraction* of immunized nodes.

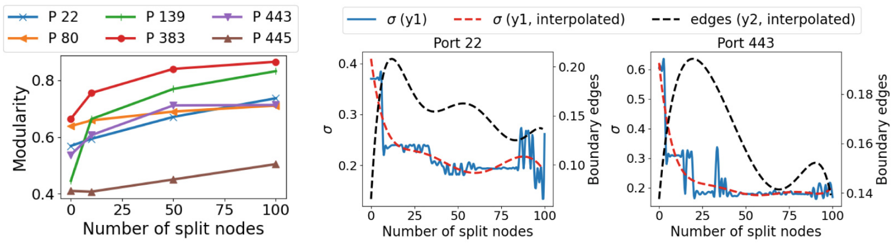
Such a rapid disintegration happens due to the highly heterogeneous degree distribution, and was shown to be characteristic of scale-free networks [2].

Isolation: Isolating communities breaks the `Attacker's Reachability Graph` into smaller connected components and thus decreases the attack footprint. This method uses a community extraction algorithm to identify communities, and then secures the borders. We compared three methods for community extraction: Leiden, Infomap and Leading Eigenvector. Leiden performed better, both in terms of fragmentation achieved and run time, therefore, the experiments in this section use the Leiden method.

Our experiments revealed that node hardening methods perform similarly at graph level and community level, because many of the most connected nodes are also inter-community "bridge nodes" (ports 22, 80, 443, 445). However, community isolation offers a viable alternative to immunization when it is preferred to secure edges instead of nodes. Hybrid strategies consisting of NodeSplit + CI-Edge are particularly promising, given that the modularity property of isolated communities increases with the number of splits on all graphs (Fig. 7a). As communities become more modular, we can achieve a better partitioning of the network. This hybrid method is able to reduce the size of the largest connected component significantly. The trade-off between the level of fragmentation obtained and the fraction of boundary edges removed is illustrated in Fig. 7b: σ decreases, and, eventually, after an initial peak, the communities become more modular, with fewer inter-community edges.

7 Evaluation of Infection Spreading

In this section, we investigate which defenses are able to mitigate or even completely eradicate attacks. To this end, we run stochastic simulations of SIIDR on the communication graphs, using the parameters estimated from modeling WannaCry from Table 2. "Patient zero" $P0$ is a single randomly chosen initially infected node. This is a good approximation, because targeted attacks would be



(a) Modularity

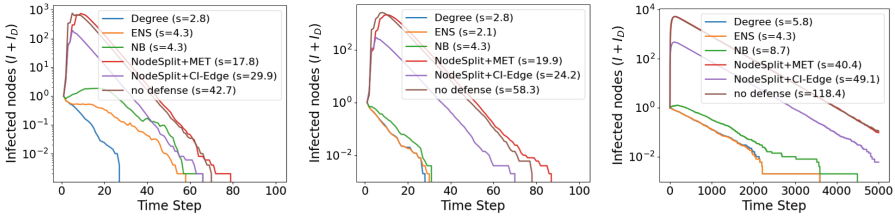
(b) Largest connected component, frac. (σ)

Fig. 7. NodeSplit+CI-Edge hybrid strategy depending on the number of splits. (a): Modularity on all graphs increases. P stands for Port. (b): The size of the largest connected component decreases (y1-axis). At the same time, the boundaries become sparser, with fewer edges (y2-axis) and better isolated communities.

even more promptly stopped by some of the defense methods (i.e., those targeting the super-spreaders, such as node hardening and NodeSplit). The results are averaged over 500 stochastic instances of $(P0, \beta, \mu, \gamma_1, \gamma_2)$ and use the following budgets: 50 nodes for NodeSplit, 50 nodes immunized with Degree, ENS, NB, and a fraction of 0.1 edges removed with MET.

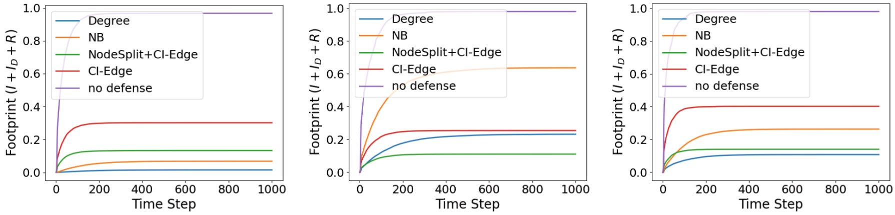
Attack Eradication: Prakash et al. [36] derived the effective strength $s = \lambda_1 \times (\beta/\mu)$ for a generic epidemics model that covers SIR and SIS in addition to several other models. If $s \leq 1$, then the infection dies out *exponentially* fast, which translates into a linear decay on a log-linear plot [7]. Our experimental results on ports 139 and 383 presented in Fig. 8 show that s is able to capture the "tipping point" for SIIDR as well, with the infection exhibiting a nearly linear decay when s is close to 1. We observe that, compared to 'no defense', the node hardening strategies lead to a 20× decrease in the effective attack strength s, while the hybrid NodeSplit-based strategies achieve a $2-3\times$ decrease. For the attack variants studied here, only the node hardening techniques are getting close to a linear infection decay, and, thus, are able to prevent a major outbreak. Hybrid NodeSplit-based methods would be successful in eradicating stealthier attacks (with slower propagation speeds than our attack variants). Attack eradication becomes even more challenging on larger scale-free graphs like ports 22, 80, 443, 445, which are comprised of millions of edges and up to 600K nodes. The difficulty is due to very high λ_1 values that have been shown to grow with the size of scale-free graphs [12,34]. Implicitly, large λ_1 results in high effective attack strength s, which requires higher budgets of hardened nodes and edges than the ones used here to quickly stop the attack.

Attack Mitigation: However, even at low budgets, our defenses can still achieve major improvements in terms of minimizing the infection footprint, defined as the total fraction of nodes in compartments I, I_D and R. We will demonstrate these results next, using the larger port-based graphs. Defense methods that reduce the largest connected component, such as node hardening and community



(a) Port 139, wc_1_5s (b) Port 383, wc_1_5s (c) Port 383, wc_8_20s

Fig. 8. Number of infected nodes ($I + I_D$) in log scale, averaged over 500 simulations, depending on the time step (lower is better). Methods that lead to linear infection decay (i.e., node hardening) are able to prevent outbreaks.



(a) Port 22, wc_1_1s (b) Port 80, wc_1_10s (c) Port 443, wc_4_5s

Fig. 9. Infection footprint ($I + I_D + R$) depending on the time step (lower is better). Degree and ENS overlap, hence, ENS was omitted. Significant reduction of the infection footprint was obtained on all ports across all attack variants after applying the defenses (other port/variant scenarios omitted due to space limit).

isolation strategies, are the most successful at attack mitigation, because they contain the attack within smaller segments of the network. Figure 9 shows a reduction of the infection footprint from about 97% with 'no defense' down to 1% on port 22 (with Degree), 11% on port 80 (with hybrid NodeSplit+CI-Edge), and 11% on port 443 (with Degree) for the attack variants illustrated here. Note that these are strong attacks, that lead to almost all nodes being infected when no defenses are employed. We obtained substantial attack mitigation across the board, for all ports and variants.

Defense Generality: In this section, we presented defense results against self-propagating malware implemented with the SIIDR model, using parameters estimated for the 7 WannaCry malware variants from Table 2. We have additionally run SPM attacks with a range of parameters for infection and recovery rates in SIIDR, varying the ratio $\beta/\mu \in (0, 1]$, and we observed similar findings. Typically, node hardening strategies, such as Degree and ENS, achieve the lowest infection footprint. Node splitting and community isolation defenses are also effective, particularly in hybrid strategies. While we selected the SIIDR model for our experiments here (given its lowest AIC scores on our malware variants), we believe that our defenses will be similarly effective against other epidemiological models for malware propagation, including SIR and SIS.

8 Related Work

Network robustness had been studied across multiple domains like infrastructure, communication and social networks, in order to understand the functioning and vulnerabilities of complex interconnected systems. Key metrics for measuring robustness (including the spectral radius and largest connected component used in this study) have been proposed and used in several papers [2,5,10,16,42]. A large body of work has looked at modeling the spread of epidemiological processes in networks [7,34–36]. Building up on this, several strategies that propose to stop the infection by manipulating the adjacency matrix of the graph have been developed, including edge removal algorithms [20,41] and node removal techniques that use centrality measures such as degree centrality, betweenness centrality, PageRank, eigenvector centrality, Katz centrality, X-centrality, etc. [16,42,43].

Rey [38] provides a comprehensive review of compartmental models for malware modeling. Mathematical proposals for the modeling and evaluation of malware dynamics include [19,28,44,48]. Other works focus on malware propagation in specific settings, like wireless sensor networks [31] or VMs under the infrastructure as a service (IaaS) architecture [17]. Closer to our work on malware fitting, Levy et al. [21] use the classical SIR model to identify the rate of infection and other parameters from real traces.

Reconfiguring communication generally has tackled the reverse problem compared to our NodeSplit method: rather than preventing attack propagation, the goal of previous research was to rewire edges in order to have alternative paths if a hub fails [8,23]. Community isolation has been studied from the point of view of the attacker, using module-based attacks to fragment social, infrastructure and biological networks [37].

9 Discussion and Conclusions

Recent large-scale cyber attacks such as WannaCry and Mirai have demonstrated how pervasive the risk of self-propagating malware has become. With cyber threats looming, it is important to proactively address vulnerabilities in networks to minimize the impact of an attack. From our extensive experiments with real-world graphs and realistic modeling of WannaCry attacks, we have gained several cybersecurity insights. We summarize these insights in a set of recommendations for security teams to increase network resilience against SPM attacks.

First, we note that SPM malware relies on network connections to spread. Therefore, the best defenses must create topological structures that prevent SPM from spreading without impeding the transmission of legitimate traffic. Closing unused ports, enforcing firewall blocking rules, and creating access control policies for communication flows are some of the possible edge hardening techniques that help reduce the attack surface. Second, the best defenses exploit the inherent hierarchy in networks. Attacks that target hubs or bridges have the potential to be the most devastating. The most effective defense is to identify key superspreaders and allocate security budgets to protect them. Third, reconfiguring communications via access control policies can greatly increase the robustness of the network against attacks. We have shown in our experiments that it is efficient to split the most connected nodes in half to balance their connections; we expect high gains even with smaller budgets.

Finally, focusing on communities within the network makes it possible to prioritize security measures under limited budgets. A modular approach can also leverage densely connected partitions by isolating them from the rest of the network to prevent a large-scale infection spread. Moreover, it addresses the issue of vanishingly small epidemic thresholds of large scale-free networks, making it easier to stop a self-propagating attack before it becomes an epidemic.

In our experiments, we analyzed in detail the impact of these recommendations as a function of their cost in terms of topological changes. From a cybersecurity perspective, hardening nodes through security updates and patches is the most straightforward method and can help prevent major SPM attacks, including the WannaCry malware studied here. Monitoring nodes through intrusion-detection systems is more costly, especially when traffic analysis is performed over a large number of connections, but it can prevent other cyber attacks. Node splitting is based on managing network traffic in a cyber network, which can be implemented with existing traffic management solutions (available in cloud data centers) or SDN-enabled policies. These solutions provide performance improvements in addition to security, but they induce additional implementation costs.

Acknowledgements. This research was sponsored by PricewaterhouseCoopers LLP and the U.S. Army Combat Capabilities Development Command Army Research Laboratory (DEVCOM ARL) under Cooperative Agreement Number W911NF-13-2-0045. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DEVCOM ARL or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

1. Akaike, H.: Information theory and an extension of the maximum likelihood principle. In: Parzen, E., Tanabe, K., Kitagawa, G. (eds.) Selected Papers of Hirotugu Akaike. Springer Series in Statistics. Springer, New York (1998). https://doi.org/10.1007/978-1-4612-1694-0_15
2. Albert, R., Jeong, H., Barabási, A.L.: Error and attack tolerance of complex networks. Nature **406**, 376–382 (2000)
3. Alstott, J., Bullmore, E., Plenz, D.: Powerlaw: a python package for analysis of heavy-tailed distributions (2013). http://arxiv.org/abs/1305.0215
4. Anjum, I., et al.: Removing the reliance on perimeters for security using network views. In: ACM SACMAT, pp. 151–162 (2022)

5. Baig, M.B., Akoglu, L.: Correlation of node importance measures: an empirical study through graph robustness. In: WWW, pp. 275–281 (2015)
6. Brauer, F.: Compartmental models in epidemiology. In: Mathematical Epidemiology. LNM, vol. 1945, pp. 19–79. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78911-6_2
7. Chakrabarti, D., Wang, Y., Wang, C., Leskovec, J., Faloutsos, C.: Epidemic thresholds in real networks. ACM Trans. Inf. Syst. Secur. **10**(4), 1:1-1:26 (2008)
8. Chan, H., Akoglu, L.: Optimizing network robustness by edge rewiring: a general framework. Data Min. Knowl. Discov. **30**(5), 1395–1425 (2016). https://doi.org/10.1007/s10618-015-0447-5
9. Chen, P., Choudhury, S., Rodriguez, L., III, A.O.H., Ray, I.: Enterprise cyber resiliency against lateral movement: a graph theoretic approach. CoRR (2019)
10. Chen, Z., Tong, H., Ying, L.: Realtime robustification of interdependent networks under cascading attacks. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 1347–1356 (2018)
11. Chernikova, A., Gozzi, N., Boboila, S., Perra, N., Eliassi-Rad, T., Oprea, A.: Modeling self-propagating malware with epidemiological models (2022)
12. Chung, F., Lu, L., Vu, V.: Eigenvalues of random power law graphs. Ann. Comb. **7**, 21–33 (2003)
13. Cloudflare: Inside the infamous Mirai IoT Botnet. https://blog.cloudflare.com/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis/. Accessed Apr 2022
14. Coresecurity: What is Ryuk Ransomware? (2022). https://www.coresecurity.com/core-labs/articles/what-is-ryuk-ransomware
15. Freitas, S., Wicker, A., Chau, D.H.P., Neil, J.: D^{2}M: dynamic defense and modeling of adversarial movement in networks. In: SDM, pp. 541–549. SIAM (2020)
16. Freitas, S., Yang, D., Kumar, S., Tong, H., Horng (Polo) Chau, D.: Graph vulnerability and robustness: a survey. TKDE (2022)
17. Gan, C., Feng, Q., Zhang, X., Zhang, Z., Zhu, Q.: Dynamical propagation model of malware for cloud computing security. IEEE Access **8**, 20325–20333 (2020)
18. GitHub: Open-source code for investigating cyber network resilience (2022). https://github.com/neu-nds2/cyber-resilience
19. Guillen, J.H., Del Rey, A.M., Casado-Vara, R.: Security countermeasures of a SCIRAS model for advanced malware propagation. IEEE Access **7**, 135472–135478 (2019)
20. Le, L.T., Eliassi-Rad, T., Tong, H.: MET: A fast algorithm for minimizing propagation in large graphs with small eigen-gaps. In: SDM, pp. 694–702 (2015)
21. Levy, N., Rubin, A., Yom-Tov, E.: Modeling infection methods of computer malware in the presence of vaccinations using epidemiological models: an analysis of real-world data. Int. J. Data Sci. Anal. **10**(4), 349–358 (2020). https://doi.org/10.1007/s41060-020-00225-1
22. Liao, C.M., Liu, W., Tang, S., Xiao, Y.: Model selection and evaluation based on emerging infectious disease data sets including A/H1N1 and Ebola. Comput. Math. Methods Med. (2015)
23. Louzada, V.H.P., Daolio, F., Herrmann, H.J., Tomassini, M.: Smart rewiring for network robustness. J. Complex Networks **1**(2), 150–159 (2013)
24. McKinley, T.J., et al.: Approximate Bayesian computation and simulation-based inference for complex stochastic epidemic models. Stat. Sci. **33**(1), 4–18 (2018)
25. Microsoft Azure: Traffic Manager documentation (2022). https://docs.microsoft.com/en-us/azure/traffic-manager/

26. Mike Azzara: What is wannacry ransomware and how does it work? https://www.mimecast.com/blog/all-you-need-to-know-about-wannacry-ransomware/
27. Minter, A., Retkute, R.: Approximate Bayesian computation for infectious disease modelling. Epidemics **29**, 100368 (2019)
28. Mishra, B.K., Pandey, S.K.: Dynamic model of worm propagation in computer network. Appl. Math. Model. **38**(7–8), 2173–2179 (2014)
29. Newman, M.E.J.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. **103**(23), 8577–8582 (2006)
30. Newman, M.: The structure and function of complex networks. SIAM Rev. **45**(2), 167–256 (2003)
31. Ojha, R.P., Srivastava, P.K., Sanyal, G., Gupta, N.: Improved model for the stability analysis of wireless sensor network against malware attacks. Wireless Pers. Commun. **116**(3), 2525–2548 (2021)
32. Zlotnik, O.: System hardening guidelines for 2022 (2021). https://www.hysolate.com/blog/system-hardening-guidelines-best-practices/
33. Ongun, T., et al.: PORTFILER: port-level network profiling for self-propagating malware detection. In: IEEE CNS, pp. 182–190 (2021)
34. Pastor-Satorras, R., Vespignani, A.: Epidemics and immunization in scale-free networks. In: Handbook of Graphs and Networks. Wiley-VCH (2003)
35. Pastor-Satorras, R., Castellano, C., Van Mieghem, P., Vespignani, A.: Epidemic processes in complex networks. Rev. Mod. Phys. **87**, 925–979 (2015)
36. Prakash, B., Chakrabarti, D., Faloutsos, M., Valler, N., Faloutsos, C.: Threshold conditions for arbitrary cascade models on arbitrary networks. Knowl. Inf. Syst. **33**, 537–546 (2011)
37. Requiao Da Cunha, B., González-Avella, J., Gonçalves, S.: Fast fragmentation of networks using module-based attacks. PLoS ONE (2015)
38. Rey, A.: Mathematical modeling of the propagation of malware: a review. Secur. Commun. Networks 8 (2015)
39. Seals, Tara: Innovative PureLocker ransomware emerges in targeted attacks (2019). https://threatpost.com/purelocker-ransomware-targeted-attacks/150229/
40. Stocks, T., Britton, T., Hohle, M.: Model selection and parameter estimation for dynamic epidemic models via iterated filtering: application to rotavirus in Germany. Biostatistics **21**(3), 400–416 (2018)
41. Tong, H., Prakash, B.A., Eliassi-Rad, T., Faloutsos, M., Faloutsos, C.: Gelling, and melting, large graphs by edge manipulation. In: CIKM, pp. 245–254 (2012)
42. Tong, H., Prakash, B.A., Tsourakakis, C., Eliassi-Rad, T., Faloutsos, C., Chau, D.H.: On the vulnerability of large graphs. In: IEEE ICDM, pp. 1091–1096 (2010)
43. Torres, L., Chan, K., Tong, H., Eliassi-Rad, T.: Nonbacktracking eigenvalues under node removal: X-centrality and targeted immunization. SIAM J. Math. Data Sci. **3**, 656–675 (2021)
44. Toutonji, O.A., Yoo, S.M., Park, M.: Stability analysis of VEISV propagation modeling for network worm attack. Appl. Math. Model. **36**(6), 2751–2761 (2012)
45. Traag, V.A., Waltman, L., van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. Sci. Rep. **9**(1), 5233 (2019)
46. Vespignani, A.: Modelling dynamical processes in complex socio-technical systems. Nat. Phys. **8**(1), 32–39 (2012)
47. Wikipedia: Structural holes. https://en.wikipedia.org/wiki/Structural_holes. Accessed Apr 2022
48. Zhu, Q., Yang, X., Ren, J.: Modeling and analysis of the spread of computer virus. Commun. Nonlinear Sci. Numer. Simul. **17**(12), 5117–5124 (2012)