SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2011)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.298

SPECIAL ISSUE PAPER

Agent-based modeling of malware dynamics in heterogeneous environments

Abhijit Bose[1]* and Kang G. Shin[2]

[1] IBM TJ Watson Research, 19 Skyline Drive, Hawthorne, NY 10532, U.S.A.
[2] Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48105, U.S.A.

ABSTRACT

The increasing convergence of power-law networks such as social networking and peer-to-peer applications, web-delivered applications, and mobile platforms makes today's users highly vulnerable to entirely new generations of malware that exploit vulnerabilities in web applications and mobile platforms for new infections, while using the power-law connectivity for finding new victims. The traditional epidemic models based on assumptions of homogeneity, average-degree distributions, and perfect-mixing are inadequate to model this type of malware propagation. In this paper, we study four aspects crucial to modeling malware propagation: *application-level interactions among users of such networks*, *local network structure*, *user mobility*, and *network coordination of malware such as botnets*. Since closed-form solutions of malware propagation considering these aspects are difficult to obtain, we describe an open-source, flexible agent-based emulation framework that can be used by malware researchers for studying today's complex malware. The framework, called Agent-Based Malware Modeling (AMM), allows different applications, network structure, network coordination, and user mobility in either a geographic or a logical domain to study various infection and propagation scenarios. In addition to traditional worms and viruses, the framework also allows modeling network coordination of malware such as botnets. The majority of the parameters used in the framework can be derived from real-life network traces collected from a network, and therefore, represent realistic malware propagation and infection scenarios. As representative examples, we examine two well-known malware spreading mechanisms: (i) a malicious virus such as Cabir spreading among the subscribers of a cellular network using Bluetooth and (ii) a hybrid worm that exploit email and file-sharing to infect users of a social network. In both cases, we identify the parameters most important to the spread of the epidemic based upon our extensive simulation results. Copyright © 2011 John Wiley & Sons, Ltd.

KEYWORDS
malware models; mobile viruses; worms; agent-based modeling; hybrid propagation; infection strategies

***Correspondence**
Abhijit Bose,
E-mail: abose@google.com
Present Address: Abhijit Bose's present address is Google Inc., Mountain View, CA 94043, U.S.A.

1. INTRODUCTION

In recent years, the landscape of malware attacks has changed considerably from fast-spreading Internet worm and virus incidents to more directed and precise attack targeting. While the primary damage from past worms and viruses such as Code Red [1], Nimda [2], and Slammer [3] has been clogged networks and expensive clean-up operations, the new generation of malware is designed to steal confidential information, control remote systems for malicious purposes, install backdoors, and shut down targeted websites by launching coordinated denial of service attacks. Examples include bot networks (botnets) [4,5], topological worms [6–9] that spread via file-sharing, instant messaging (IM), Internet Relay Chat (IRC) and email networks, 'drive-by-downloads' [10], and emerging mobile viruses [11,12] capable of spreading via Bluetooth and SMS/MMS messages. The increasing convergence of social networks and heterogenous computing devices has led to a new generation of malware that can exploit vulnerabilities of web applications and mobile platforms to infect users while using the power-law topology of these networks to propagate very fast. While there have been many studies to capture propagation dynamics of Internet-scale malware such as Code Red or Nimda, very few studies have considered the heterogeneity and complexity of mobile and social networking environments at a sufficient detail when modeling the emerging malware families. Due to the rapidly growing popularity of power-law networks and heterogeneous computing devices, studying malware propagation is an important area of research.

In particular, four key aspects of malware propagation have not received adequate attention in existing models: (i) most power-law networks are overlaid on a combination of wireless LANs, wired segments and mobile networks, with different bandwidth and latency distributions, (ii) interactions among users can be diverse and (iii) mobility of users can affect the rate at which an epidemic can either grow or subside in the network, and (iv) malware such as botnets use network coordination techniques to launch coordinated attacks. In this paper, we develop realistic propagation models at the time-scale and network structure of mobile and power-law network environments, addressing the above aspects. Most of the parameters in our epidemic models can be derived explicitly from traces collected from such networks, similar to what we collected from a large enterprise network. Although our simulations show the epidemic spreading within a target enterprise environment, our modeling approach is general.

This paper makes three primary contributions. First, we demonstrate that the current epidemic models fail to capture applications, network connectivity structure, and user mobility. Second, we present a general-purpose simulation framework for understanding malware epidemics in such *integrated* environments. Our framework, called *Agent-Based Malware Modeling* (AMM), explicitly incorporates non-homogeneous user interactions, user connectivities, malware coordination (i.e. botnets), network bandwidth, channel models of short-range radio devices, and user mobility within a domain. An AMM model is built upon autonomous agents that incorporate realistic models of services and mobility. The agents can be arranged hierarchically in much the same way an enterprise network is designed. For example, in our implementation of AMM, 'base station agents' can monitor and collect aggregated statistics of activities of 'mobile device agents' in their respective wireless local area networks (WLANs). In case of botnets, network coordination can be achieved by configuring either a centralized command-and-control (C&C) agent or via distributed agents in the model. Third, AMM can be used by network designers and IT security staff to investigate propagation dynamics and the final size of an epidemic for a variety of scenarios. This will then help them to plan for proactive defense strategies. Our goal is to publish AMM as an open-source software for the malware research and network security community.

The paper is organized as follows: We discuss the primary challenges in modeling malware in power-law and mobile environments in Section 2. Section 3 describes the AMM framework in detail, including infection models for applications commonly targeted by malware, and user mobility models as implemented in the framework. Section 4 discusses modeling botnets in AMM by setting up a time-evolving C&C coordination network embedded within the primary power-law network. In Section 5, we simulate two well-known attack scenarios to understand the factors affecting the spreading rate of an epidemic. Section 6 briefly reviews the existing literature on malware modeling. We conclude in Section 7 with a discussion of our future work.

2. MALWARE MODELING CHALLENGES

There are four major challenges in accurate modeling of an epidemic in power-law and mobile environments. To the best of our knowledge, there does not exist any fine-grained model that addresses all of these challenges.

2.1. Application diversity

The diversity of power-law network applications (e.g. email, social networks, and P2P) and mobile platforms means that even when a set of hosts are running similar services, not all of them are equally vulnerable to the same exploits due to different versions of client software. The epidemic models [1,3] developed for wide-area networks, such as the Internet, do not consider such heterogeneity at the level of individual users and hosts. However, diversity is important when we consider propagation dynamics at microscopic levels (i.e. considering individual users and hosts) where the homogeneity assumption is no longer valid. A naive application of the Kephart–White epidemic model [13] is often not valid in this case. To incorporate heterogeneity in epidemic models, many previous studies (e.g., Ref. [6]) have assumed a vulnerability ratio for the population. However, it is not clear how one can come up with a vulnerability ratio for hybrid malware that exploit multiple power-law networks as in the case of Nimda and Fizzer. The epidemic modeling framework presented in this paper explicitly captures message-level interactions among the users, and captures the diversity of applications and the host environment (OS, applications, transport protocols, etc.). The service interactions from malicious agents are superimposed on the normal background traffic calculated from collected traces, and therefore, represent a more realistic environment.

2.2. Local network structure

The shortcomings of the uniform-mixing assumption have led to development of epidemic models that capture the effects of contact patterns between individuals, instead of the mean-field theory. In uniform-mixing models, an infected host has the same probability of infecting any vulnerable host in the population—this assumption is clearly not valid for malware targeting power-law or mobile networks that exploit the local network structure. Therefore, several recent studies have investigated the effects of local network connectivity on epidemic spreading. We refer to Ref. [14]

and the references therein as the relevant literature, particularly for complex networks, small-world effects, models of network growth, and power-law degree distribution. The various models can be divided into two broad categories based on whether the contact network structure is either 'small-world' [15] or 'scale-free' [16]. The scale-free nature of technological networks and epidemic spreading in such networks have also been studied, for example, in Refs. [17] and [18], respectively. However, these studies derive only the steady-state outcome of the epidemic in the limit of long times, and do not provide the time evolution of the infection process which is crucial in understanding how an epidemic spreads in its initial stages and where to deploy containment strategies. The propagation dynamics of malicious codes in various models of scale-free networks have been studied by a number of researchers [19,20]. In majority of these studies, topologies are generated with power-law degree distributions via either the Barabasi and Albert [15] or the Klemm and Eguiluz [16] algorithm for a specified number of nodes and a given power-law index. For example, Figure 1 plots a typical distribution of the computed 'betweenness centrality' (BC) of P2P, email and overlapping (hosts having email and P2P services) topologies from our traces collected from a large class-B IP network. The BC at a vertex k is computed as follows. Let $C_k(i, j)$ denote the set of the shortest pathways between a pair of vertices i and j through the vertex k. The fraction $g_k(i, j) = (C_k(i, j))/(\sum C_k(i, j))$ indicates the importance of the vertex k between two vertices i and j. The BC of vertex k is then defined as $g_k = \sum_{i \neq j} g_k(i, j)$. Figure 1 confirms the scale-free nature of the service topologies in a real-life enterprise environment. However, replacing an enterprise service topology with a corresponding power-law network model still does not account for the true propagation dynamics. Since almost all malware exploit specific vulnerabilities in sequence of messages and in specific OS stack or applications, the local interactions constitute an important criteria for infection. This is best captured when interaction topologies are constructed explicitly from traces collected from the target network.

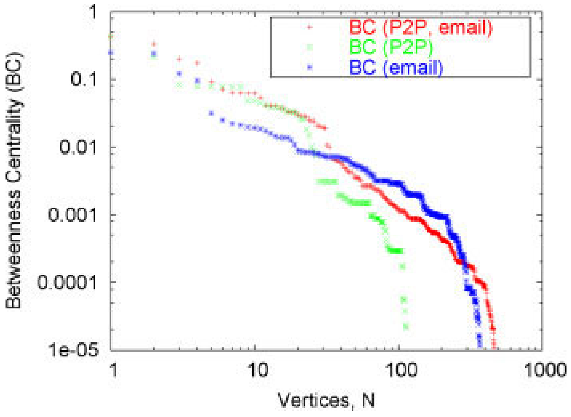

Figure 1. BC of email and P2P topologies.

2.3. Mobile users

Today's mobile platforms introduce new propagation vectors such as Bluetooth, SMS/MMS messaging, and situational applications that malware writers increasingly target. User mobility changes the interaction topology as devices move around the physical environment of the enterprise. The problems with the Kephart–White infection model for malware that spread via short-range RF such as Bluetooth have been identified in a recent study [21]. The standard epidemic models fail because they ignore node velocity and the non-homogeneous connectivity distributions among the nodes. In addition, the location-specific density distribution of mobile devices can potentially affect the spread of an epidemic. The spread of an epidemic has not been studied in environments that consist of overlapping mobile wireless and wired segments with users switching to different network resources in different locations of an enterprise. Our framework addresses this by incorporating user mobility models as part of each agent in the domain.

2.4. Network coordination among infected hosts

Network coordination among the infected hosts is a special consideration for modeling botnets. While traditional worms and viruses can be modeled as independent infections, botnets use a number of mechanisms to achieve coordination among the infected hosts. We enable network coordination among the connected nodes of a malware network via an embedding of a secondary network structure in AMM. Since botnets have modeling requirements that are different from traditional malware, we describe incorporation of botnets in AMM in a separate section (see Section 4).

3. AGENT-BASED MALWARE MODELING

3.1. Motivation

Deterministic methods [1,3] developed for modeling the previous generations of worms, such as Code Red, Sapphire, and variants thereof, are well-suited to characterize the spread of an epidemic in large populations such as the Internet. However, they are not accurate for modeling small populations such as a social network. As we already argued, these models unrealistically assume perfect-mixing and homogeneity within the population. Malware that propagates by exploiting local node connectivity can hardly be modeled by the homogeneity assumption. The homogeneity assumption fails to hold on both host attributes (i.e., diversity of OSs, services, and mobility) as well as the network structure among the hosts. The homogeneity assumption also does not hold when interactions are highly correlated with network structure. Topological worms [6,20] spread by targeting specific services such as IM, P2P, and email—the topology of these service-interaction networks[†] may lead to significant deviations from the results of the differential equation-based models. Similarly, in case of mobile nodes, the connectivity patterns change depending on how users roam around the network as well as their speed and pause times.

An agent-based modeling approach can relax the homogeneity and perfect-mixing assumptions by (i) incorporating heterogeneity in agent attributes, (ii) modeling the state transitions of an agent as an explicit stochastic process, and (iii) allowing highly structured topologies of service interactions among the agents. The interaction topologies can be generated from traces collected from a network, and input to the agent-based model.

Note that AMM provides a more realistic description of the world. It can easily incorporate changes in individual user mobility patterns and messaging patterns among the agents (i.e., service interactions). This makes the model closer to reality than averaged equation-based methods. As discussed in Ref. [22], agent-based modeling makes it possible to realize the full potential of the data one may have to describe the dynamics of a physical phenomenon. The complexity of differential equation-based approaches will have to increase exponentially to account for hosts running multiple services or for mobile agents. On the other hand, AMM models activities at the agent level, and sources of randomness are applied to these activities and the underlying service queues, as opposed to arbitrarily adding noise terms to an aggregate epidemic model.

While AMM can readily incorporate agent diversity and interaction topologies, the computations required to perform a full sensitivity analysis can be expensive due to the large number of parameters in a typical agent-based model. Clearly, such an approach is not feasible for modeling malware propagation over the entire Internet. However, our studies show that network-level modeling with AMM is feasible and offers a much richer simulation approach to mobile malware modeling. Due to the stochastic nature of AMM, it is possible that in some cases, either no epidemic is observed or the epidemic ends early even when the basic reproduction number indicates otherwise. We investigate this further in our evaluation (Section 5), and show that the local network structure greatly influences the probability of an infection spreading through an enterprise.

3.2. The AMM framework

We now describe the AMM prototype developed for studying malware propagation. We model a given domain (either geographic or logical) as a collection of networked and autonomous decision-making entities called *agents*. The agents represent networked devices within an enterprise, such as desktops, servers, laptops, access points, PDAs, and cell phones, and their users. The connectivity among the agents depends on the interaction topology. In case of agents representing mobile devices, the connectivity changes as users roam about the physical space of the domain. The behaviors of the agents are specified by a set of services (or, applications) running on them. For example, an agent may consist of client programs for email and IM, whereas another agent may consist of an email (SMTP) server only. Thus, there are two types of topologies in our simulation environment. The *physical* connectivity is determined by the physical network infrastructure, movement of the agents, location of access points and base stations, whereas the *logical* connectivity is determined by the messages exchanged among the agents. An agent may participate in multiple logical topologies corresponding to different services like email, IM, P2P, social network, etc. We also group the agents in a hierarchical manner. For example, agents representing wireless access points can keep track of mobile devices in their respective WLANs. Accordingly, access point agents are able to collect information aggregated over the individual devices in their WLANs. This capability of higher-level agents to aggregate observations collected from lower-level agents reflects real-life processing of information in such environments. We note that the information processed at these different levels can also be used to activate different response mechanisms against a spreading malware.

Figure 2 shows a flowchart of our prototype simulator. The first step is to prepare the following input parameters for AMM: (i) *infection-model parameters* for the target services, (ii) *topology* of service interactions among the hosts, (iii) *location* of access points and base stations, (iv) *mobility models* for hosts that are mobile, (v) *attack vector* of malware, (vi) *detection model* of malware, and (vii) *an attack response model* (containment, rate-limiting, anti-virus, etc.), if any. At the beginning of a simulation run, agents are instantiated with appropriate attributes (i.e. agent-level objects). At each timestep, the coordinates of mobile agents are updated based on their mobility patterns, resulting in new connectivity graphs. Each agent exchanges messages with other agents according to the service model––the probability of any of these messages being infected is calculated from the service-infection model. The time steps are repeated over a user-specified number of trials so that the results can be averaged over these trials. The simulator is general enough to experiment with different algorithms for malware detection and containment. The detection algorithm can be implemented at various levels of hierarchy, e.g., at individual devices, access points, or networks segments, depending on the granularity of the detection algorithm. If a domain has an established set of containment policies, whenever an infection is detected, containment steps can be activated at various levels of the logical and physical network topologies.

The infection-model parameters and service topologies can be extracted from traces collected from the service provider's network. The infection-model of a service consists of parameters that affect the propagation of a malware

[†] The term "contact networks" is also used in the epidemiology literature.

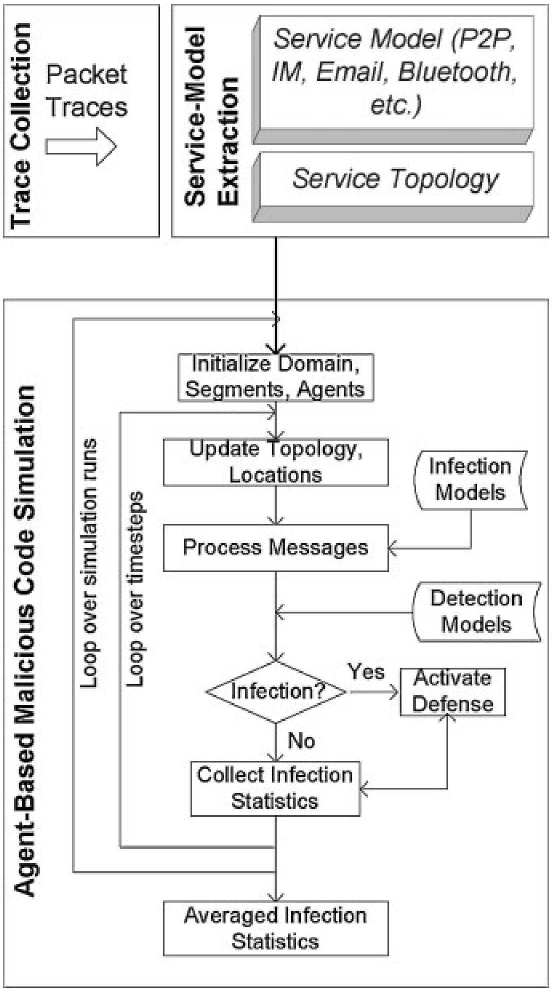
Trace Collection
Packet Traces
Service-Model Extraction
Service Model (P2P, IM, Email, Bluetooth, etc.)
Service Topology
Agent-Based Malicious Code Simulation
Loop over simulation runs
Loop over timesteps
Initialize Domain, Segments, Agents
Update Topology, Locations
Infection Models
Process Messages
Detection Models
Infection?
Yes
Activate Defense
No
Collect Infection Statistics
Averaged Infection Statistics

Figure 2. Flowchart of AMM prototype.

targeting that specific service. We describe infection-model parameters for email, P2P, IM, and Bluetooth in Section 5. The model parameters are ideally fitted to data from a set of network traces collected from the target enterprise environment as shown in Figure 2. However, data from the existing literature are often sufficient for incorporating an infection model exploiting a given service. Examples of possible services that may be targeted by emerging malware are SMS/MMS, web services, VoIP, etc. Reliable infection models for these services are not yet available. However, using our simulator, various possible infection models can be studied—this is where the trace-driven AMM can be very attractive and promising. The inclusion of a service-infection model results in a more realistic epidemic spreading in AMM. An alternative is to simply input the topology of a particular service like email, and consider all nodes in the graph equally vulnerable to an email worm. The spread of the epidemic in this case solely depends on a constant infection probability and the topology of the email network. While most modeling literature on malware spreading that exploits services follows this methodology, this simulates only the worst-case attack scenario and may not represent the true spreading of the epidemic within an enterprise.

3.2.1. Agent attributes.

Figure 3 shows an UML representation of our AMM prototype. The framework has three major classes of agents: *domain agents* maintain a list of all global parameters, perform averaging over multiple trials and contain a number of *network segment agents*. The network segment agents can be of three types: *wired, cellular*, and *wireless* (e.g., 802.11b WLANs). Each segment agent contains a number of *device agents* (i.e., hosts and users). In case of wireless and cellular segments, the device agents are built with embedded mobility models (described later), whereas for a wired network segment, the device agents are stationary. Note that the segments can be overlapping, i.e., a wireless segment can be built on top of a wired segment with access points being the communication links between the two segments. As mentioned earlier, we also explicitly construct agents to model access points, base stations and service gateways for popular services such as SMS or IM. This allows us to investigate not only malware propagation but also containment and mitigation strategies.

3.2.1.1. Agent mobility Mobile and wireless devices are a rapidly increasing constituent of any enterprise environment. To simulate such an environment, AMM employs mobile agents. There are a variety of mobility models available to simulate different cellular and ad hoc wireless environments. We refer to Refs. [23,24] for a discussion of these models. We have implemented two commonly used models proposed for ad hoc wireless and cellular environments, namely, Random Waypoint (RWP) and Gauss-Markov (GM) mobility models, respectively.

In the RWP model, a node randomly chooses a destination in the simulation area and moves at a speed v chosen randomly from the uniform distribution $[v_{\min}, v_{\max}]$ along a straight path towards the destination. Then, the node pauses for a constant time t_{pause} before it chooses a new destination randomly. A node in the RWP model is, therefore, characterized by its current coordinates, current speed, current destination point, and pause time. Following Ref. [25], we avoid the initial high variability in average neighbor numbers by discarding the results of the initial 1000 s of simulation time and then saving the mobile positions as the initial starting locations of our simulation.

The GM mobility model uses a Markov process in updating both speed and direction of a mobile node. Originally proposed for simulation of PCS networks [26], GM allows adaptation to different levels of randomness via a tunable parameter. In GM, each node updates its speed (s_t) and direction (d_t) at time t based on their values at time $t-1$ as:

$$s_t = \alpha s_{t-1} + (1-\alpha)\bar{s} + \sqrt{(1-\alpha^2)}s_{x_{t-1}} \quad (1)$$

$$d_t = \alpha d_{t-1} + (1-\alpha)\bar{d} + \sqrt{(1-\alpha^2)}d_{x_{t-1}} \quad (2)$$

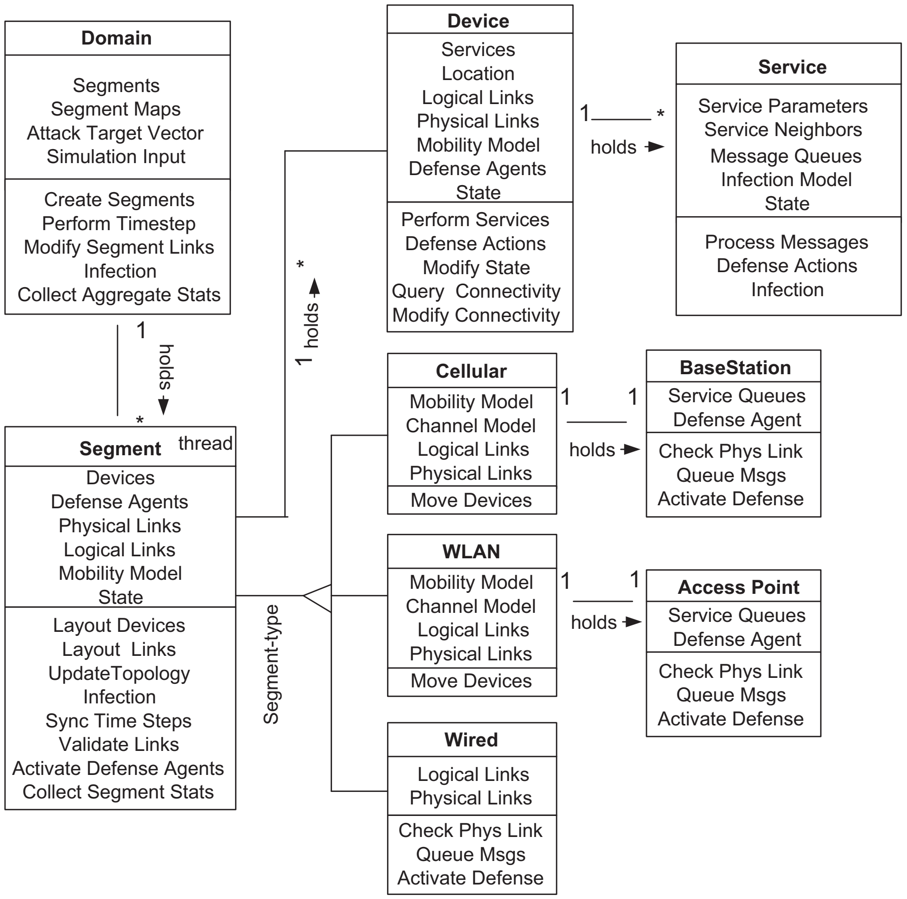

Figure 3. Agent attributes and functions.

where $0 \le \alpha \le 1$ is the tuning parameter, $\bar{s}$ and $\bar{d}$ are the mean value of speed and direction as $n \to \infty$, respectively. $s_{x_{t-1}}$ and $d_{x_{t-1}}$ are random variables drawn from a Gaussian distribution.

3.2.2. Service-infection models.

In AMM, a device agent can be set up to run a set of services. Following the worm taxonomy model of Ellis [27], we denote the service availability as a mapping of services to ports and write it as a set of tuples $\{(s_1, \text{port}_1), (s_1, \text{port}_1), \cdots, (s_n, \text{port}_n)\}$. Some of these services constitute the set of exploits for a spreading malware. Figure 4 shows the state machine of a generic service running on an agent. The set of states of a service are {Immune, Vulnerable, Infected, {Quarantined, Throttled}}. {Quarantined, Throttled} represents fine-grained states denoting the defensive action taken when an infection is detected. This allows one to simulate different defensive measures and compare their effectiveness. For known attacks, an anti-virus patch can also be applied to a service, thereby transitioning the state of the service from *Infected* to *Immune*. In a contained network, a device can

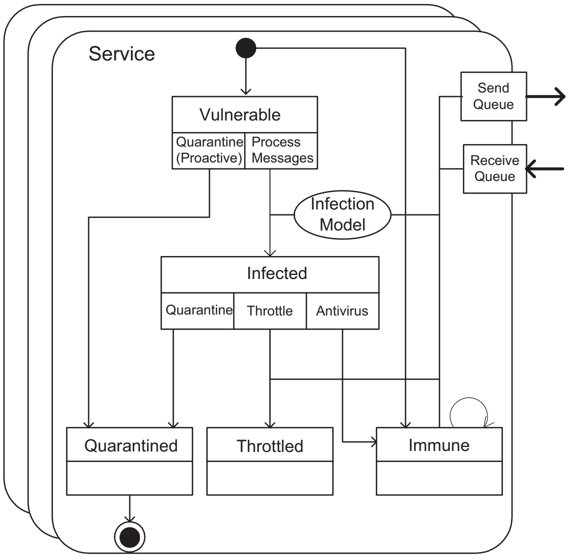

Figure 4. Model of an infection targeting a service.

Infection Model	Model Parameters	Source
SMS	message sending rate, $n_s(N_{cs})$ message receiving rate, $n_r(N_{cs})$ cdf of user-to-user message size, B SMS user topology, $G(N_{cs}, E_{cs})$ cdf of message service time, T_s^{sms} malicious agent messaging rate, $m_s(I_{cs})$ message reading probability, P_r^{sms}	T T T T T M M
Bluetooth	path loss component, standard deviation for fading model, threshold radius, r_0 transmit power, p_t threshold receive power, $p_{r,th}$	E E M E E
IM	message sending rate, $n_s(N)$ file transfer rate per user, $n_f(N)$ cdf of message service time, T_s^{im} malicious agent messaging rate, $m_s(I_{cs})$ message reading probability, P_r^{im}	T T T M M
P2P	file query rate, $n_q(N_p)$ cdf of session duration, S cdf of peer uptime, T_{up} file opening probability, P_f^p	T T T M
Email	email checking time interval, T(~N($_{T}$, ${}^{2}{}_{T}$)) email opening probability, P_m	M M

T: Trace, M: Emperical Model, E: Calibration Experiement

Figure 5. Infection models and their parameters.

attain any of the three final states {Quarantined, Throttled, Immune}.

A device agent sends and receives messages from other agents corresponding to each service tuple (s, port). The service class data structure achieves this via separate send and receive message queues for each service. Each service also has an infection model of a malware exploiting the specific vulnerability. The state transition from *Vulnerable* to *Infected* is determined by this infection model. The infection model is service-specific and consists of a set of parameters with their values given either as data ranges or probability density functions. Figure 5 lists the service-infection model parameters for SMS, Bluetooth, IM, P2P, and Email, that we have implemented in AMM. The sources of these parameters are traces collected from an enterprise (T), empirical models of user behaviors (M) and calibration experiments (E). When the state of any service is *Infected*, the outgoing messages from an agent are tagged as *Infected* based on runtime values of these parameters. Similarly, when an infected message is received from another host or user in the network, the infection model determines whether the service state should be changed from *Vulnerable* to *Infected*. Next, we detail service-infection models for SMS, Bluetooth, IM, P2P, and Email.

3.2.2.1. SMS model A recent study [28] presented SMS user behavior based on call data records and SS7 traces collected over a 3-week period from a large cellular carrier with 10 million mobile users. The data allowed us to construct a realistic SMS messaging network with the following parameters: message sending rates ($n_s(N_{cs})$), message receiving rates ($n_r(N_{cs})$), cumulative density functions (cdf) of user-to-user message size (B) and message service time (T_s^{sms}), and the SMS user topology ($G(N_{cs}, E_{cs})$), where N_{cs} and E_{cs} represent the total number of cellular subscribers in the network and the number of service-interaction edges, respectively. We also introduce two parameters describing the spread of the malicious agent: malicious agent messaging rate ($m_s(I_{cs})$) for the set of infected mobile users (I_{cs}) and a probability of reading an infected message (P_r^{sms}). In the absence of traces collected during an actual occurrence of a mobile worm or virus, one has to estimate these two parameters.

3.2.2.2. Bluetooth RF model The connectivity of an ad hoc wireless network such as those formed by Bluetooth and other short-range RF devices strongly influences the effectiveness of malware spreading via proximity scanning. To determine if two Bluetooth-enabled devices are neighbors, one can simply use a threshold distance (r_0). For example, in case of class-2 Bluetooth devices, $r_0 = 10$ m. However, one should consider a more realistic wireless channel model by considering shadowing effects that are induced by the presence of obstacles. This means that the connectivity between two devices is now a stochastic parameter. Following Bettstetter and Hartmann [23], we adopt a log-normal shadow-fading model to determine if an infected device can send a message to a nearby device using an existing vulnerability in the Bluetooth stack. In a shadow-fading environment, the signal attenuation, $\beta(u, v)$ between a pair of nodes u and v is expressed as the sum of (i) a deterministic geometric component β_1 based on the relative distance $r(u, v)$ and (ii) a stochastic component β_2 where

$$\beta_1(u, v) = \alpha\, 10\log_{10}\left(\frac{r(u, v)}{1m}\right)\mathrm{d}B \tag{3}$$

and β_2 is chosen from a log-normal probability density function:

$$f_{\beta_2}(\beta_2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\beta_2^2}{2\sigma^2}\right) \tag{4}$$

where α is the path-loss component ($2 \leq \alpha \leq 5$) and σ is the standard deviation ($|\sigma| \leq 10\,\mathrm{d}B$). For a given transmit power p_t and a threshold receive power $p_{r,th}$, two devices u and v are neighbors if the attenuation between them satisfies: $\beta(u, v) \leq \beta_{th}$ where the threshold attenuation β_{th} is given by:

$$\beta_{th} = 10\log_{10}\left(\frac{p_t}{p_{r,th}}\right)\mathrm{d}B \tag{5}$$

Equations (3) and (5) along with the mobility models give us a propagation model of a malware that exploits Bluetooth vulnerability and spreads to different areas of an enterprise as the users move about the physical space. In Section 5, we study Cabir-like malware that exploit vulnerabilities in the Bluetooth stack to propagate.

3.2.2.3. IM model We refer to Ref. [7] for a discussion of IM worms, client vulnerabilities, and proposed defensive measures. Our model for IM worm propagation consists of: message sending rate ($n_{\mathrm{s}}(N)$), file transfer rate ($n_{\mathrm{f}}(N)$), message service time ($T_{\mathrm{s}}^{\mathrm{im}}$), malicious agent messaging rate ($m_{\mathrm{s}}(I_{\mathrm{cs}})$) and message (i.e., attachment or link) opening probability ($P_{\mathrm{r}}^{\mathrm{im}}$), where N and I_{cs} represent the total number of IM users and the set of infected IM users, respectively. Of these, $n_{\mathrm{s}}(N)$, $n_{\mathrm{f}}(N)$, and $T_{\mathrm{s}}^{\mathrm{im}}$ can be derived from IM server logs within an enterprise.

3.2.2.4. P2P model The authors of Ref. [6] present an epidemic simulator that takes as input Gnutella topology graphs and the probability of a node being a guardian node. They denote a guardian node as a member of the P2P network that can detect a worm and forward alerts to its neighbors. Although they consider the effect of node diversity by having a fraction of the nodes as initially immune to the attack, they did not consider the peer-level diversity. The propagation of a file-sharing worm is influenced by such factors as peer uptime ($T_{\mathrm{i}}^{\mathrm{up}}$), peer query activity ($Q_{\mathrm{i}}$), and session duration (S_{i}). If peers tend to be unavailable frequently, a file-sharing worm will not spread quickly. This is because the degree of replication necessary to ensure that the file content is consistently accessible is low for peers with small up-times. Similarly, peer activity levels and how peers issue and respond to queries, influence the probability of an infected file to be downloaded. We adopt the distribution functions for peer up-time, query activity, and session duration described in Refs. [29,30] based on experimental observations of common P2P networks. Similar to mass-mailing worms, a downloaded file must be opened by the user for the file-sharing worm code to be activated. Therefore, we add a file opening probability ($P_{\mathrm{i}}^{\mathrm{f}}$) for each peer.

3.2.2.5. Email model We adopt the model developed by Zou *et al.* [8] based on human behaviors affecting email worms. Their model is based on two key parameters: an email checking time interval (T_i) which is the time interval between checking two consecutive emails at host i, and an opening probability (P_i^{m}) which is the probability of a user on host i opening an email with a worm-infected payload or attachment. As in Ref. [8], we assume that the mean of T_i and P_i are generated from Gaussian-distributed random variables $T(\sim N(\mu_T, \sigma_T^2))$ and $P(\sim N(\mu_P, \sigma_P^2))$, respectively. The parameters used for T and P are: $\mu_T = 40$, $\sigma T = 20$, $\mu_P = 0.5$, and $\sigma_P = 0.3$.

Although there have been recent studies on the modeling of malware propagation using IM, P2P, and Email, our work has important differences from these studies. The usage of real-life traces to construct the service-infection models creates realistic power-law network environments. In our framework, services can be composed for any given agent in the domain, and therefore, hybrid worms using IM and P2P (e.g., Bropia) can be easily simulated. These simulations generate realistic traffic corresponding to IM and P2P messages in topologies that are constructed directly from the traces. As an example of hybrid worms, we will later study a worm that propagates via both email and P2P networks.

4. AGENT COORDINATION IN AMM

4.1. Motivation

In order to model botnets, we have designed AMM with the capability of coordinated agents. This is required for modeling botnets that are behaviorally different from the type of worms and viruses such as Code Red or Bropia. Botnets [5,31] are networks of semi-autonomous software agents or ‘bots’ that are deployed using either centralized or distributed C&C infrastructure. In recent years, they have become synonymous with malicious activities on the Internet, e.g., bots have been deployed to launch denial-of-service attacks, install spyware on compromised hosts, serve adware, and email spams. Botnets are different from self-propagating worms and other malware in several ways. For example, nodes in a botnet can be commanded to perform a set of coordinated tasks by the owner of the botnet (also known as the ‘botmaster’). This makes them far more malicious than traditional malware. Further, bots routinely use traditional malware as their infection and propagation mechanism.

Botnets also utilize increasingly sophisticated evasion techniques, e.g., fast-flux DNS, tunneling over other traffic (VoIP, HTTP, IPv6) [32], and encrypted traffic [33]. These techniques make it very difficult to detect and shut down a botnet even when some of its nodes are exposed. Traditional botnets have had a centralized C&C structure based on the IRC protocol. This type of infrastructure can be brought down easily by finding and eliminating the central node. In contrast, the fast-flux DNS technique [34] used by more recent botnets allows a botnet to hide its malware delivery sites behind a continuously changing network of compromised hosts that act as proxies. This is accomplished by individual nodes within the network re-registering their IP addresses periodically as part of the DNS A record list for a single DNS name. This can also mask the botmaster’s system which exploits the network via a series of proxies, thereby making detection very difficult. While C&C is centralized in IRL botnets, P2P botnets such as Storm have developed a decentralized C&C model. In P2P botnets, each compromised host maintains a list of peers in the network. Upon receipt of a message from the botmaster, a node forwards the message to its peers. As a result, when some of the nodes of a botnet are discovered, the rest of the botnet can still function. P2P botnets have grown tremendously in recent years with the introduction of DHT-based (distributed hash table) peer-to-peer protocols such as Chord and Overnet (using the Kademlia DHT algorithm [35]).

While we have not yet implemented botnet evasion techniques, AMM currently supports both centralized and distributed C&C, via an embedding mechanism called ‘agent coordination network,’ presented later in this section.

4.2. Botnet service-infection model

Botnets are modeled in AMM by extending the service-infection model of a traditional worm. This is due to the fact that the infection phase of a bot such as Peacomm/Storm [36] is very similar to that of a traditional worm. It is only after infection that the bot downloads the secondary injections from a remote host to establish the communication primitive for the botnet, and joins the botnet. From this point on, the botmaster is able to command the newly infected bot, install additional malicious binaries, and to modify C&C addresses. Figure 6 shows the states of a P2P bot such as Storm as implemented in AMM.

Most P2P botnets such as Storm propagate via email spam, by directing the recipient to download an executable from a website. When this executable runs, it can take advantage of known client-side vulnerabilities to install a rootkit. This initial phase creates the initialization file *spooldir.ini* that will later contain the list of peers. At this point, the host (an AMM agent) can be considered infected but not yet part of the botnet. Storm uses an encrypted version of the UDP-based Overnet protocol to find other nodes in the P2P network. The malware tries to initiate the peer-lookup algorithm every 10 min if no hosts in the initial list of peers are responsive [36]. When a peer responds, the bot may undertake one of several activities: (i) update the list of peers in *spooldir.ini*, (ii) download details of a new spam campaign, (iii) download updates of existing executables, and (iv) initiate spam campaigns or denial of service activities. Note that the Overnet/Kademlia protocol is primarily used as a directory service to find other nodes by Storm. For managing C&C activities such as which one of the above actions the bot will perform, Storm uses a custom TCP-based protocol. This is implemented in AMM as an overlay network as described below.

4.3. Agent coordination network

The C&C protocol is implemented in AMM as an additional service running on an infected agent. However, unlike traditional malware, this service is not started on an agent automatically at the start of the simulation. Only after an infected agent has successfully found a peer and joined the botnet, the service is activated. As a result, a secondary network, which we call the 'agent coordination network,' is overlaid on top of the Overnet-based P2P network (in case of Storm). This overlay network changes its structure over time as bots join and exit the C&C service. As shown in Figure 6, all AMM messages are placed in either the 'send' or 'receive' queues of an agent. Depending on whether a message is intended for the Overnet directory service or for the C&C service, the message is delivered to the appropriate service running on agents.

5. SIMULATION OF ATTACK SCENARIOS

In this section, we investigate two likely attack scenarios using the AMM framework. First, we study the potential

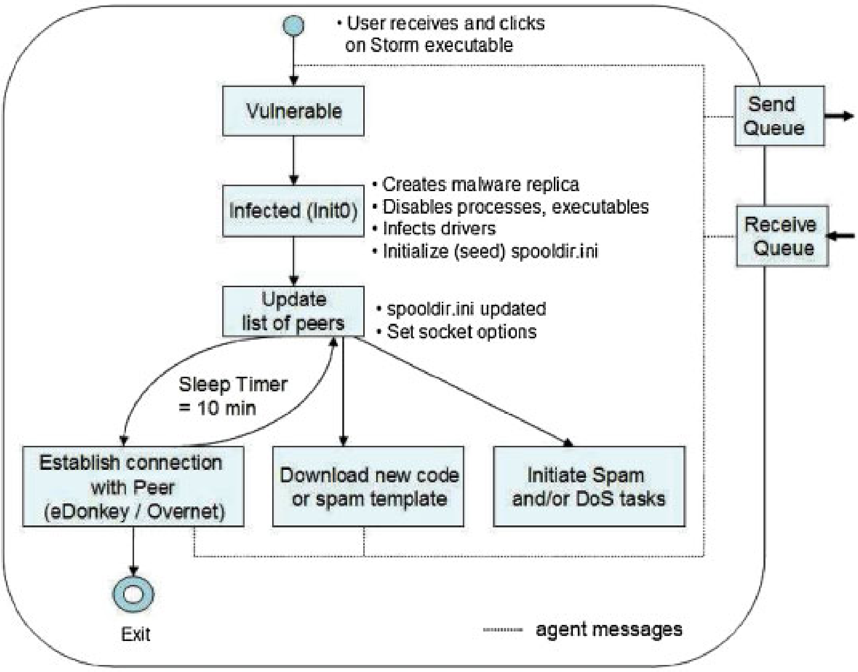
• User receives and clicks on Storm executable
Vulnerable
Infected (Init0)
• Creates malware replica
• Disables processes, executables
• Infects drivers
• Initialize (seed) spooldir.ini
Update list of peers
• spooldir.ini updated
• Set socket options
Sleep Timer = 10 min
Establish connection with Peer (eDonkey / Overnet)
Download new code or spam template
Initiate Spam and/or DoS tasks
Exit
Send Queue
Receive Queue
agent messages

Figure 6. AMM service-infection model for storm malware.

DOI: 10.1002/sec

spread of a Bluetooth-based virus such as Cabir in a multi-cell mobile network. Next, we investigate the spreading rate of a hybrid topological worm that can spread via both Email and P2P file-sharing networks.

5.1. Proximity scanning via Bluetooth

This attack scenario considers subscribers of a mobile data and voice provider. We assume that a fraction of the subscribers have unprotected Class-2 Bluetooth-enabled cell phones, PDAs, and other mobile devices. The range of a Class-2 Bluetooth device is typically 10 m. The coverage area of the subscribers is serviced by 10 base stations. We consider two different channel models: (i) a threshold radius of 10 m and (ii) shadow fading described in Section 3. In the latter case, the connectivity of the mobile nodes is dependent on the terrain conditions. We then simulate the spread of a Cabir-like virus [11], a much-publicized mobile virus that infects unprotected Bluetooth-enabled devices. The mobility of users is modeled using RWP and we consider both 'slow'- and 'fast'-moving users to study the effect of node velocity on the spread of the virus. The various parameters for the simulation, such as velocities for slow (v_{slow}) and fast (v_{fast}) movement, are presented in Table I. The notations are explained in Figure 5.

We denote $E(I(t))$ as the expected number of infected nodes at time t, averaged over 100 trial runs of the simulator. We have used a time step of 200 ms, and all simulations were continued for 1000 time steps unless all nodes in the network were already infected. We consider two cases of initial infection, $I(0)=1$ and 4. Since Cabir affects only devices running the Symbian OS, we have included the vulnerability ratio (v) to denote a fraction of the nodes with this particular OS. Figure 7 shows the effect of node velocity on the spread of the epidemic. At very high velocities, the connectivity of the nodes change very quickly, allowing for a high mixing between infected and vulnerable nodes. This accounts for the large difference in $E(I(t))$ between the slow- and fast-moving experiments, especially at a low vulnerability ratio ($v=0.1$). It is interesting to note that when most of the nodes in the network are vulnerable ($v=0.9$), the spread of the virus is no longer dependent on the node velocity because the majority of the interactions with an infected node result in new infections.

Table I. Parameters for proximity-based propagation.

Parameter	Value
α	3
σ	4 dB
β_{th}	30 dB
r_0	10 m
v_{slow} (RWP)	[2, 24]m/s
v_{fast} (RWP)	[350, 400]m/s
t_{pause}	0
v	[0, 0.1, 0.9]
$I(0)$	[1, 4]

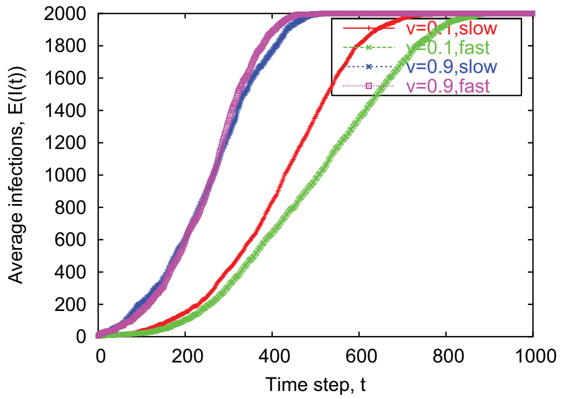

Figure 7. Effect of node velocity on Cabir propagation.

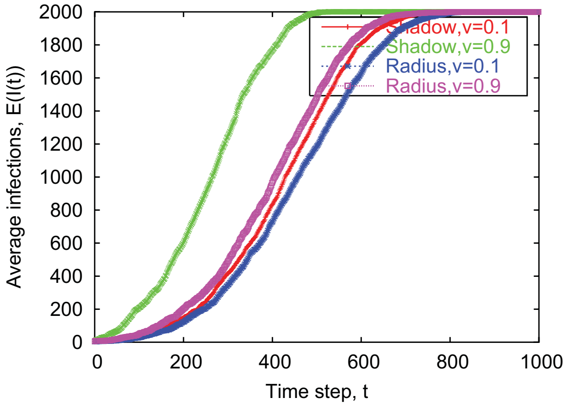

Figure 8. Effect of channel models on Cabir propagation ($v=$ vulnerability ratio).

Figure 8 shows the impact of choosing a particular Bluetooth channel model on the growth of the epidemic. The shadow-fading model results in higher connectivity among the nodes, thereby increasing the probability of contact with an infected node. This is in contrast with infection based on a threshold radius of 10 m. The epidemic growth curves based on the threshold radius model for $v=0.1$ and $v=0.9$ are virtually identical. The data in Figure 8 illustrates the need for accurate modeling of the radio interface in mobile virus spreading. To account for devices running other mobile OSs, we present the results for different values of v and $I(0)$ in Figure 9. The results are intuitive since a higher value of either v or $I(0)$ will result in a higher growth rate of the virus. To understand the effect of pause times, we simulated the virus spreading with $v=0.9$, slow-moving users and pause times of 0, 100 and 1000 ms. Figure 10 indicates that as pause time increases, the mixing among the infected and vulnerable nodes decreases, resulting in a slower spread of the epidemic.

In a recent study [21], the authors studied the spread of a Bluetooth virus in a mobile ad hoc network based on the threshold radius approach. Although they did not consider

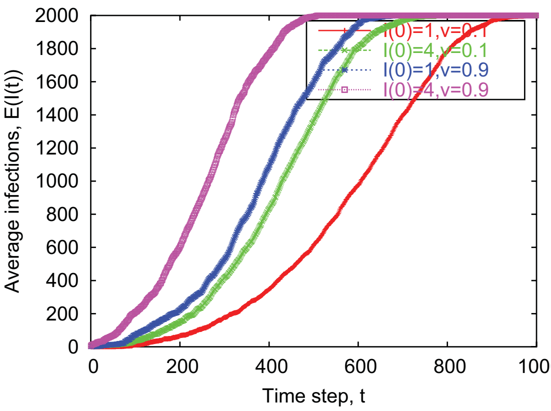

Figure 9. Effect of vulnerability ratio (*v*) on Cabir propagation.

the effect of channel fading, we simulated one of the examples presented in Ref. [21] to compare the results. There is an important difference in the two sets of simulations. The infection model in Ref. [21] consists of a removal rate δ where as our study assumes that a mobile node, once infected, stays infected for the rest of the simulation, i.e. we consider a completely unprotected network. However, we can still compare the average connectivity among the nodes between the two approaches since this is an important parameter not considered by the traditional deterministic SI (Susceptible-Infected) and Kephart–White models. Following Ref. [21], we ran a simulation with 60 mobile nodes in an area of $1000 \times 1000\,\mathrm{m}^2$ and a speed range of [5,20]m/s. We also assumed that the nodes are equipped with a class 1 Bluetooth device ($r_0 = 100\,\mathrm{m}$). After 3000 time steps, we calculated the average connectivity of the nodes to be 2.09 as compared to a value of 2.37 in case of Ref. [21]. We also found the initial growth rates of the two simulations very similar. There is a persistent infection in case of Ref. [21] but the classical KW model overpredicts its magnitude.

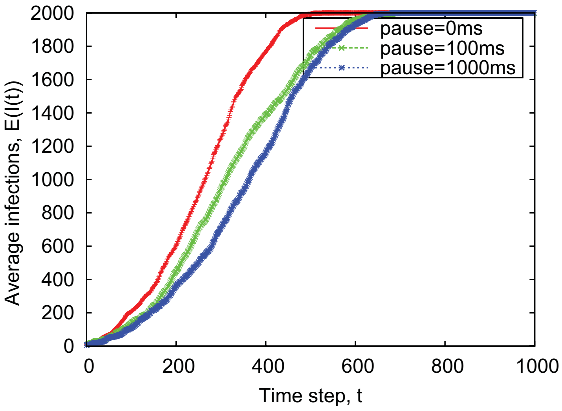

Figure 10. Effect of pause time on Cabir propagation.

Table II. Trace properties.

Parameter	Value
Size (bytes)	5 756 476 279
Duration (s)	3600
# Packets	120 067 838
# IP addresses	11 647
# Vertices (email)	2126
# Edges (email)	2550
# Vertices (P2P)	7150
# Edges (P2P)	7287

5.2. Topological spreading via email and P2P file-sharing

Next, we study the propagation of a hybrid worm that can spread via multiple vectors, in particular email and P2P file-sharing networks. Specific examples of this class of worms are Bagle.AH and Netsky.C. From the collected traces of the Class-B IP network mentioned earlier, our simulation framework reconstructs topologies of email and P2P networks at periodic intervals, corresponding to the respective protocols (SMTP, IMAP, and POP for email; Gnutella, eDonkey, and BitTorrent for P2P). These time-stamped service topologies are then input along with the corresponding infection models to simulate propagation of the hybrid worm. Table II shows the properties of a typical trace we used for this set of simulations.

Figure 11 compares the number of infected hosts for the hybrid (email and P2P) worm with a mass-mailing worm, for an initial number ($N(0)$) of infected hosts $N(0) = 2$ and 10. Initially infected hosts are chosen at random with an equal probability in the email and P2P topologies. We perform 1000 repetitions with each set of simulation parameters to calculate the average values of the number of infections. Figure 11 indicates that a hybrid worm can spread extremely fast through a network by exploiting both services. Since the growth rate of spreading is very high, a fully automated containment system will be necessary to prevent the spread of such worms—human countermeasures

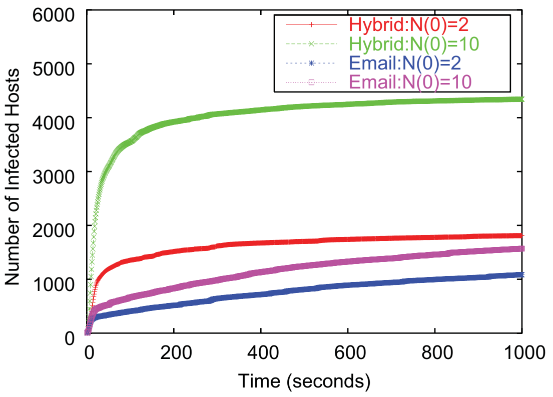

Figure 11. Propagation of single-vector and hybrid topological worms.

DOI: 10.1002/sec

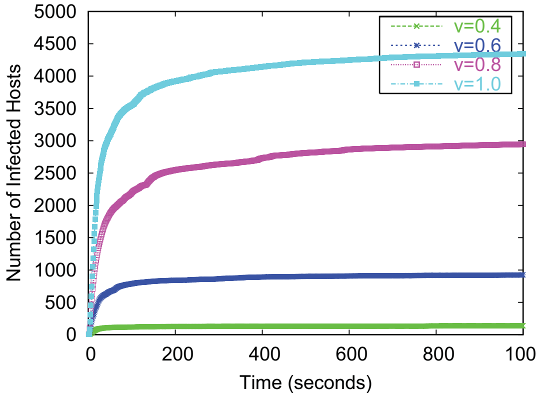

Figure 12. Effect of end-host diversity on hybrid worm propagation ($N(0) = 10$) (v = vulnerability ratio).

will be useless. The results in Figure 11 assume that all email and P2P hosts are equally vulnerable to the worm attack. In practice, there is considerable diversity in client versions, OS, hardware and application software. This diversity will affect hybrid worms targeting multiple services. To account for such diversity, we repeat the above simulations with different numbers of initially immune nodes. Figure 12 shows the number of infected hosts for different fractions of the vulnerable population (denoted as v). The results indicate a significant reduction in the total number of infected hosts due to node diversity.

6. RELATED WORK

The most relevant literature are malware simulators such as Refs. [37–39]. However, these simulators assume a simplified model of Internet connectivity and employ a mathematical model for the worm traffic. They do not consider an *integrated* multi-service and multi-mode network that may contain wired, wireless and cellular segments. As a result, any change in topology due to user mobility is not reflected in the simulations, thus limiting these simulators to traditional worms and viruses. To simulate epidemics in very large networks with millions of hosts (i.e., Internet-scale epidemics such as Code Red and Melissa), several researchers have developed distributed worm simulators. For example, the authors of Ref. [40] presented PAWS, a distributed simulator running on the Emulab testbed. The authors derived inter-Autonomous Systems (AS) bandwidth data from existing literature to simulate the major Internet AS, and developed congestion models from worm traffic. They simulated scanning worms (Code Red v2 and Slammer) and showed excellent agreement with the experimental data from the real world. While very powerful for studying Internet-scale epidemics, PAWS may not be suitable for studying the heterogeneous environment of an enterprise network as well as topological worms. A number of simulators have attempted to recreate the AS topology of the Internet. Our approach is to generate topologies specific to the service provider's network from collected traces. For enterprise-level modeling and vulnerability assessment, trace-based simulations provide a detailed and more realistic method. Another excellent distributed simulation framework for the Internet is presented in Ref. [39]. The authors used GTNetS and PDNS (a parallel version of NS-2) simulators to generate packet-level traces of worm traffic. Due to the large computational overhead, these simulators are typically run on powerful clusters (often deploying 100 CPUs or more). There are many epidemiological models of worm spread reported in the literature. The deterministic models use simplified assumptions of homogeneous topologies and aggregated behavior. We have mentioned some of the relevant literature in Ref. [3].

Recent studies [41,42] have proposed a taxonomy of botnets along several dimensions such as C&C models, communication protocols, evasion techniques, and effectiveness of botnet structures. These dimensions can be used to tailor different response techniques to detect and disrupt botnets. For example, the authors of Ref. [41] observe that targeted responses are highly effective against scale-free botnets where as random-graph botnets (e.g. those using P2P protocols) are highly resistant to both random and targeted responses. The authors of Ref. [43] present a stochastic activity network model of P2P botnets and examine how different factors impact the growth of the network. Recently, [44] has provided new insight into the economics of the Storm botnet and its spam generation agents.

7. CONCLUSION

The traditional epidemic models of malware propagation do not capture several unique characteristics of today's malware and network environments. Interactions among the users/infected hosts at different spatial and time scales create different vulnerable interaction topologies, rather than an average degree of connectivity among the nodes, as assumed by many epidemiological models. Mobile users with laptops, PDAs, and cell phones not only contribute to these time-varying service topologies, but also introduce new vulnerabilities as well, e.g., Mabir-type viruses that can spread via SMS/MMS messages and Bluetooth connections. Further, today's networks consist of diverse network segments with different levels of bandwidth, services, and latencies. All of these factors affect the growth rate of an epidemic. Our agent-based modeling framework captures these factors and uses topologies constructed from traffic traces collected from service providers' networks. Using this framework, an enterprise can perform a realistic vulnerability assessment of its popular services (e.g. SMS, Bluetooth, email, P2P, and IM) given all its network segments. These services are often targeted by malware writers and increasingly, new malware are designed to exploit many of these services simultaneously. Our extensive simulations show that combining these services increases the initial growth rate of the epidemic almost exponentially

DOI: 10.1002/sec

and therefore, human countermeasures will be useless. The simulation study of Cabir also points out the potential vulnerability from unprotected Bluetooth interfaces in a mobile network.

There are several areas of future work. We plan to model botnets in more detail, especially their coordination mechanisms. There are also other emerging web-application-based attacks (see Ref. [10]), mobile attacks targeted to SMS/MMS, and Bluetooth users that are not yet well-understood. The flexibility of AMM allows investigation of these emerging malware by abstracting their behavior in terms of a small number of parameters while keeping their propagation and coordination details at a fine-grained level.

REFERENCES

1. Moore D, Shannon C, Brown J. Code-red: a case study on the spread and victims of an internet worm. In *ACM Internet Measurement Workshop*, 2002. www.caida.org/outreach/papers/2002/codered.
2. Symantec, W32.nimda.a@mm virus description, 2001. securityresponse.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html.
3. Moore D, Paxson V, Savage S, Shannon C, Staniford S, Weaver N. Inside the slammer worm, 2003. www.computer.org/security/v1n4/j4wea.htm.
4. McLaughlin L. Bot software spreads, causes new worries. *IEEE Distributed Systems Online* 2004; **5**.
5. Rajab M, Zarfoss J, Monrose F, Terzis A. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*. ACM New York, NY, U.S.A., 2006; 41–52.
6. Zhou L, Zhang L, McSherry F, Immorlica N, Costa M, Chien S. A first look at peer-to-peer worms: threats and defenses. In *4th International Workshop on Peer-To-Peer Systems*, 2005.
7. Mannan M, van Oorschot PC. Instant messaging worms, analysis and countermeasures. In *3rd Workshop on Rapid Malcode (WORM)*, 2005.
8. Zou CC, Towsley D, Gong W. Email worm modeling and defense. In *13th International Conference on Computer Communications and Networks (ICCCN'04)*, 2004.
9. Symantec, W32.hllw.fizzer@mm worm, 2003. securityresponse.symantec.com/avcenter/venc /data/w32.hllw.fizzer@mm.html.
10. Provos N, McNamee D, Mavrommatis P, Wang K, Modadugu N. The Ghost In The Browser: Analysis of Web-based Malware. In *Workshop on Hot Topics in Understanding Botnets (HotBots)*, Vol. 10, 10 April, 2007.
11. Symantec, Symbos.mabir worm description, April 2005. securityresponse.symantec.com/avcenter/venc/data/symbos.mabir.html.
12. FSecure, F-secure virus descriptions: Cardtrap.a. December 2004. www.f-secure.com/v-descs/cardtrap_a.shtml.
13. Kephart J, White S. Directed-graph epidemiological models of computer viruses. In *Proceedings of the IEEE Computer Symposium on Research in Security and Privacy*, May 1991; 343–359.
14. Newman M. The structure and function of complex networks. In *SIAM Review* 2003; **45**(2): 167–256. http://www.citeseer.ist.psu.edu/newman03structure.html.
15. Barabasi A-L, Albert R. Emergence of scaling in random networks. *Science* 1999; **286**: 509–512.
16. Klemm K, Eguluz VM. Highly clustered scale-free networks. *Physical Review E* 2002; **65**: 036123.
17. Ebel H, Mielsch L, Bornholdt S. Scale-free topology of e-mail networks. In *Physical Review E* 2002; **66**: 035103.
18. Pastor-Satorras R, Vespignani A. Epidemics and immunization in scale-free networks. In *Handbook of Graphs and Networks*, Wiley-VCH, Berlin, 2003.
19. Briesemeister L, Lincoln P, Porras P. Epidemic profiles and defense of scale-free networks. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode (WORM)*, 2003; 67–75.
20. Balthrop J, Forrest S, Newman MEJ, Williamson MM. Technological networks and the spread of computer viruses. *Science* 2004; **304**(5670): 527–529.
21. Mickens JW, Noble BD. Modeling epidemic spreading in mobile environments. In *Proceedings of the 2005 ACM Workshop on Wireless Security (WiSe 2005)*, September 2005; 77–86.
22. Bonabeau E. Agent-based modeling: Methods and techniques for simulating human systems. In *PNAS* 2002; **99**: 7280–7287.
23. Bettstetter C, Hartmann C. Connectivity of wireless multihop networks in a shadow fading environment. *ACM/Springer Wireless Networks* 2005; **11**(5): 571–579.
24. Camp T, Boleng J, Davies V. A survey of mobility models for ad hoc network research. In *Wireless Communications and Mobile Computing* 2002; **2**(5): 483–502.
25. Navidi W, Camp T, Bauer N. Improving the accuracy of random waypoint simulations through steady-state initialization. In *Proceedings of the 15th International Conference on Modeling and Simulation*, March 2004; 319–326.
26. Liang B, Haas Z. Predictive distance-based mobility management for PCS networks. In *Proceedings of the INFOCOM*, March 1999.
27. Ellis DR. Worm anatomy and model. In *WORM'03: Proceedings of the 2003 ACM Workshop on Rapid Malcode*, 2003; 42–50.

28. Samanta V. A study of mobile messaging services. In *UCLA Master's Thesis*, 2005.
29. Saroiu S, Gummadi PK, Gribble SD. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Computing and Networking*, 2002.
30. Schlosser MT, Condie TE, Kamvar SD. Simulating a file-sharing p2p network. In *First Workshop on Semantics in P2P and Grid Computing*, 2002.
31. Cooke E, Jahanian F, McPherson D. The zombie roundup: Understanding, detecting, and disrupting botnets. In *Proceedings of the USENIX SRUTI Workshop*, 2005; 39–44.
32. Singh K, Srivastava A, Giffin J, Lee W. Evaluating Email Feasibility for Botnet Command and Control. In *IEEE International Conference on Dependable Systems and Networks With FTCS and DCC, 2008 (DSN 2008)*, 2008; 376–385.
33. Gu G, Zhang J, Lee W. BotSniffer: Detecting botnet command and control channels in network traffic. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08)*, 2008.
34. Holz T, Gorecki C, Rieck K, Freiling F. Detection and mitigation of fast-flux service networks. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08)*, 2008.
35. Kutzner K, Fuhrmann T. Measuring large overlay networks-the overnet example. In *Proceedings of the 14th KiVS*, 2005, Springer.
36. Porras P, Saidi H, Yegneswaran V. A Multi-Perspective Analysis of the Storm (Peacomm) Worm. SRI International: Menlo Park, CA, 2007.
37. Liljenstam M, Nicol D, Berk V, Gray R. Simulating realistic network worm traffic for worm warning system design and testing. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode (WORM 2003)*, 2003.
38. Wagner A, Dübendorfer T, Plattner B, Hiestand R. Experiences with worm propagation simulations. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode*, 2003.
39. Perumalla K, Sundaragopalan S. High-fidelity modeling of computer network worms. In *Annual Computer Security Applications Conference (ACSAC)*, December 2004.
40. Wei S, Mirkovic J, Swany M. Distributed worm simulation with a realistic internet model. In *Principles of Advanced and Distributed Simulation (PADS)*, 2005.
41. Dagon D, Gu G, Lee C, Lee W. A taxonomy of botnet structures. In *Proceedings of the 23 Annual Computer Security Applications Conference (ACSAC'07)*. Springer, 2007.
42. Incorporated T. Taxonomy of Botnet Threats. Technical Report, Trend Micro Incorporated, 2006.
43. Van Ruitenbeek E, Sanders W. Modeling Peer-to-Peer Botnets. In *Proceedings of the 2008 Fifth International Conference on Quantitative Evaluation of Systems*, IEEE Computer Society Washington, DC, U.S.A., 2008; 307--316.
44. Kanich C, Kreibich C, Levchenko K, Enright B, Voelker G, Paxson V, Savage S. Spamalytics: an empirical analysis of spam marketing conversion. In *Proceedings of the 15th ACM conference on Computer and Communications Security*. Alexandria, VA, ACM New York, NY, U.S.A., 2008; 3–14.