Modeling Botnet Propagation Using Time Zones

David Dagon[1] Cliff Zou[2] Wenke Lee[1]
[1]College of Computing, Georgia Institute of Technology,
801 Atlantic Dr., Atlanta, Georgia, USA 30332-0280
{dagon, wenke}@cc.gatech.edu
http://www.cc.gatech.edu/

[2]School of Computer Science, University of Central Florida,
4000 Central Florida Blvd. Orlando, FL 32816-2362
czou@cs.ucf.edu

Abstract

Time zones play an important and unexplored role in malware epidemics. To understand how time and location affect malware spread dynamics, we studied botnets, or large coordinated collections of victim machines (zombies) controlled by attackers. Over a six month period we observed dozens of botnets representing millions of victims. We noted diurnal properties in botnet activity, which we suspect occurs because victims turn their computers off at night. Through binary analysis, we also confirmed that some botnets demonstrated a bias in infecting regional populations.

Clearly, computers that are offline are not infectious, and any regional bias in infections will affect the overall growth of the botnet. We therefore created a diurnal propagation model. The model uses diurnal shaping functions to capture regional variations in online vulnerable populations.

The diurnal model also lets one compare propagation rates for different botnets, and prioritize response. Because of variations in release times and diurnal shaping functions particular to an infection, botnets released later in time may actually surpass other botnets that have an advanced start. Since response times for malware outbreaks is now measured in hours, being able to predict short-term propagation dynamics lets us allocate resources more intelligently. We used empirical data from botnets to evaluate the analytical model.

1 Introduction

Epidemiological models of malware propagation are maturing. Earlier work used simple susceptible-infected (SI) models to measure the total infected population over time [ZGT02]. Follow-up work significantly expanded this analysis to include patching behavior (resistance) in susceptible-infected-recovered (SIR) models [KRD04]. Despite these many improvements, much of our understanding of computer worm epidemiology still relies on models created by the public health community in the 1920s [DG99].

Continued improvements in worm models will come from two areas: an improved understanding of the problem domain, and improved ability to respond, which makes new factors relevant to a model. Improvements belonging to the first category can be found in more recent analysis such as [SM04], which traced significant worm outbreaks, and [ZTGC05, WPSC03, WSP04], which examined a specific type of routed worm, and [ZTG04], which examines specific types of propagation (e.g., e-mail). Model enhancements belonging to the second category are far fewer. So far, quarantine-based analysis has been the primary response-oriented improvement to malware propagation models [ZGT03, PBS$^+$04, MSVS03].

Our work belongs to this second category, and builds on recent improvements in response technologies. Over the previous years, efforts at creating Internet-wide monitoring networks have yielded some results. Distributed sensing projects [Ull05, YBJ04, Par04] can take some credit for helping reduce the response time for worms to hours instead of days. Anti-virus companies similarly respond to outbreaks often within hours [Mar04].

This improved response makes time a more relevant factor for worm models. In Section 3 we note how time zones play a critical role in malware propagation. Now that response times take only hours [Mar04], and are often localized, models of malware spreading dynamics must similarly improve.

In addition to time, we also note that location plays a critical role in malware spreading. Some malware tends to focus on particular geographic regions, corresponding to different market segments for vulnerable software (e.g., a language edition of an operating system). We combine both of these factors in models that consider the importance of time zones (literally, time and zone location) in propagation.

Our research looks at propagation dynamics in botnets. We studied dozens of botnets, comprised of millions of individual victims over a six month period. Our study of botnets reveals an intriguing diurnal pattern to botnet activity. Our model explains this behavior, and has two principal benefits: (a) the ability to predict future botnet propagation characteristics, for those botnets using similar vulnerabilities, and (b) the ability to priority rank malware based on time-of-release and regional focus, so that resources are devoted to faster spreading botnets.

Section 2 provides a background discussion of botnets, and details our data collection efforts. In Section 3, we provide a model of botnet propagation. After noting related work in Section 4, the conclusion in Section 5 suggests further areas of study.

2 Background

Using automated scanners and tools, attackers have carved out a large portion of the Internet as continuously infected networks. The victims are bots or zombies in large networks, or botnets, controlled by hackers. There are tens (if not hundreds) of millions of such victims on the Internet [Dag05]. Some estimates hold that over 170,000 new victims are compromised each day [Cip05]. Indeed it is hardly possible for home users to purchase a new computer and successfully update before becoming attacked. The "vulnerability window", or the time before a random infection strikes a new computer, is often less than 20 minutes. As a result, others have observed that a "botnet is comparable to compulsory military service for windows boxes" [The05a]. For a general discussion of botnets, see [CJ05, SS03, The05a].

For purposes of modeling, we can think of botnets as heterogeneous collections of infections. They are composed of the victims reaped from different viruses, worms and trojans. Thus, botnets are correctly referred to as either viruses, worms or trojans, depending on the context. The original infections compel the victims to run bot programs, which allow for remote administration.

Victims are usually spread over diverse parts of the world, but can be concentrated in particular regions, depending on how the underlying infections propagate. For example, some attacks target a particular language edition of an operating system, or use a regional language as part of a social engineering ploy. Such factors tend to concentrate the victim population in a particular location. (We speculate that this may explain why most e-mail virus propagations use simple English, to maximize its appeal.) These regional variations in infected populations play an important role in malware spread dynamics.

2.1 Data Collection

To control or "rally" their botnets, botmasters force their victims to contact command-and-control (C&C) servers (e.g., an IRC server, a webpage, or e-mail). Once connected to the servers, the bots are given instructions, put to work, or made to download additional programs. If such centralized servers are recovered, botmasters can merely update DNS entries to point to a new central server. This practice is known as "herding" a botnet to a new location. While such centralized control may not be the favored topology for much longer [Dag05, CJ05], we can manipulate this common feature of botnets to perform simple data collection.

To gather botnets for study, we identified botnets through various traditional means (e.g., honeypots), and then manipulated the DNS service for the C&C server, so that all traffic was sent to our sinkhole for study. The sinkholes were used to run tarpits [Har02, Lis01], honeypots [Spi03, Pro05], and light-weight responders, e.g., [Pro03, Kre03]. For more discussion of network response options see [YBP05].

Our sinkhole redirection was accomplished by several steps. First, using captured malware (e.g., from a honeypot, spam filter, honeyd, and other commonly available sources), we identify the command and control server used by the botnet. This can be done by unpacking the binary (e.g., with the help of tools such as IDA Pro, or PEiD [JQsx05] and a hex editor) and scanning the binary for DNS resolution operations, (e.g., `gethostbyname(3)`). This is also done less precisely by observing the malware's rallying behavior in an emulator (e.g., a virtual honeypot). The latter is less reliable because malware may selectively resolve one of many encoded C&C domains, or behave differently in the emulator [Hol05]. Hand-driven binary analysis, however, can usually reveal the malware's rallying behavior.

Second, we then identify the DNS Start of Authority (SOA) for the command and control box using well-known techniques [RIP05]. We then contact the registrar for the domain and the DNS authority, and instruct them to either "park" the DNS (so that, for example, an RFC 1918 non-routable address is returned), or to supply an A-Rec for a sinkhole, or a similar suitable Record Response (RR). We followed a strict one-ip-per-botnet rule, to facilitate the study of single botnets. For most bots, we also used layer-7 sinkholes (i.e., honeyd, or similar scripts) instead of layer-4 sinkholes (e.g., routing blackholes) to prevent random scans from being confused with actual botnet participation.

Conceptually, one might think of this capturing technique as a form of DNS self-poisoning, except that altering the DNS entry for the bot domain is done legitimately, in accordance with the DNS operator's policies, and with the permission and cooperation of all relevant authorities. In our study, we worked with several DNS operators who agreed to redirect bot victims to our sinkhole. The operators would enter `CNAME` records in their DNS servers to point victims to our sinkhole.

Since all the botnets being studied used DNS to locate their C&C server, redirection captured most of the botnet members. Through binary analysis, we confirmed that the bots did not use hard-coded IP addresses. We also restricted our study to non-public servers, so no legitimate traffic polluted our data capture. Our sinkholes completed 3-way TCP handshakes with victims, so that random Internet SYN scans did not skew our population counts. Further, by setting a zero TCP window, our sinkhole prevented most bots from disconnecting (e.g., through an application-layer idle timeout), and then reconnecting after changing dynamic addresses. This reduced the number of victims that were double-counted due to DHCP churn.

These techniques yield what we believe is a fairly accurate population count for an infection. Nonetheless, our data probably did have casual, non-malicious connection attempts, and certainly had some amount of DHCP churn.

Thus, while others models use trace files from large "internet telescope" structures to infer which machines scanning the internet share a common infection [Moo02b, YBP05], we believe our simple data collection technique yields accurate trace files for each infection. More importantly, this technique can potentially distinguish two botnets that use the same infection, while scan-based sensors may associate the traffic together based on port numbers. Significantly, we also learn which victims are associated with which botnet, based on the domain they attempt to resolve.

Thus, although our data collection technique focuses on botnets using centralized DNS (currently, the most common rallying technique used by botnets), we do not have to correlate scans from diverse sources to infer the structure of the botnet. We were able to direct some 50 botnets to the sinkhole over a six month period. Our sinkhole captured botnets ranging from just a few hundred victims to tens of thousands of victims. One botnet featured over 350,000 victims, a record [CJ05].

One might wonder whether this redirection technique yields data about worms instead of botnets. After all, many of the botnets are created by worms. The question is: How is redirection different from traditional worm measurement techniques? We believe redirection measures botnets (as opposed to just worms) because the traffic yield is entirely related to the command-and-control of a malicious network. Worm measurement techniques, by contrast, tend to collect scans by worms (i.e., propagation attempts), and do not usually capture the coordinating messages between bots and botmasters (i.e., DNS resolution of the command-and-control domain). Since DNS redirection gives us the opportunity to witness *only* the command-and-control traffic, and not the propagation attempts, our technique measures properties of botnets, regardless of how the underlying infection spreads. Thus, the model we propose is for botnets, albeit botnets created by worms.

The data collection technique is not the focus of the paper, and deserves more careful separate study. We welcome input from the research community on what other factors (besides our use of command-and-control messages) permit the measurement of botnets. Additionally, we acknowledge that there are certain types of botnets that would evade such measurement efforts. We merely use redirection to quickly perform population counts on botnets. In section 3 we discuss particular botnets in detail used to derive our diurnal propagation model.

3 Model of Botnet Growth

Our goal is to use our observations of previous botnets to predict the behavior of future botnets. Botnets are so widespread that we need a technique to comparatively rank them, and help prioritize responses. Existing models let us predict the total botnet population over lengthy periods of time (e.g., over days). But since most viruses used to spread infections are short lived, we need a model that can predict short-term variations in population growth.

Further, existing models treat all vulnerable populations as the same. Our observations of botnets, however, show that they use a heterogenous mix of different infections exploiting different sets of vulnerabilities, often in distinct networks, with variable behavior across time zones. We therefore need a model that can express differences in susceptible populations, and gauge how this affects propagation speed.

There are a variety of reasons why existing models have not examined factors such as time zones. First, converting a network address into a time zone (or geographic region) is difficult, as noted in [Mic05], and there are few available resources, e.g., [Coo03]. Second, since the earlier models were proposed, the state of the art for response and quarantine has improved. Most antivirus companies can issue signature updates in under 12 hours (or less), so understanding the short-term growth of a worm is more relevant.

For our model, we make another observation about botnet behavior. We were first struck by the strongly diurnal nature of the botnets trapped in the sinkhole. Figure 1(a) shows a typical plot of SYN rates over time, broken down by geographic regions, for a large 350K member botnet. This pattern repeated itself for both email-spreading worms and scanning worms observed in the sinkhole. A logical explanation is that many users turn their computers off at night, creating a sort of natural quarantine period, and varying the number of victims available in a geographical region.

Such significant changes in populations over time surely affects propagation rates. To model the strongly diurnal behavior of botnets observed in Figure 1(a), we analyze bots grouped into time zones. Consider a very simplified model represented in Figure 1(b), where one host is shown in a column of time zones, TZ. In the first hour, the infected host in TZ_i infects TZ_{i-1} and TZ_{i+1}; however, since TZ_{i-1} is experiencing a low diurnal phase at $Hour_2$ (e.g., night time, represented by diagonalized shaded boxes), the malware does not spread further until several hours later (indicated by a dashed line). By contrast, the infection sent to TZ_{i+1} spreads immediately, only later entering a diurnal phase.

This conceptual model exaggerates a key property of the diurnal model: different propagation rates, depending on time zone and time of day. Time Zones not only express relative time, but also geography. If there are variable numbers of infected hosts in each region, then the "natural quarantine" effect created by a rolling diurnal low phase can have a significant impact on malware populations and growth.

Below, we describe a model to express the variable number of infected hosts, time zones, and regions of the Internet that we observed in the empirical data. We then test this model against other observed botnets. The model in turn lets us estimate short-term population projections for a given worm, based on its regional focus, and the time of day. The model also tells us when bots spread fastest, and allows us to compare the short-term "virulence" of two different bots. This in turn can be used to improve surveillance and prioritize response.

3.1 Time Zone-Based Propagation Modeling

We model the computers in each time zone as a "group". The computers in each time zone have the same diurnal dynamics, no matter whether they are infected or still vulnerable. In our model, the diurnal property of computers is determined by computer user behavior, not by the infection status of computers. If a user changes his diurnal behavior because he discovers his computer is infected, then we assume the computer will quickly be patched or removed by the user.

The number of infected hosts in a region varies over time. So we define $\alpha(t)$ as the "*diurnal shaping function*", or the fraction of computers (that have the vulnerability being exploited by the botnet under consideration) in a time zone that is still on-line at time t. Therefore, $\alpha(t)$ is a periodical function with the period of 24 hours. Usually, $\alpha(t)$ reaches its peak level at daytime and its lowest level at night when many users go to sleep and shutdown their computers. Not all the computers are shut off at night, of course. So in modeling and experiments, we can derive $\alpha(t)$ for a given time zone based on monitored malicious traffic.

In the following, we first derive the worm propagation diurnal model for a single time-zone by assuming computers in the time zone form a closed networking system. We then derive the diurnal model for the entire Internet by considering multiple time zones.

3.2 Diurnal Model for a Single Time Zone

First, we consider a closed network within a single time zone. Thus, all computers in the network have the same diurnal dynamics. Define $I(t)$ as the number of infected hosts at time t; $S(t)$ as the number of vulnerable hosts at time t; $N(t)$ as the number of hosts that are originally vulnerable to the worm under consideration.

We define the population $N(t)$ as a variable since such a model covers the case where vulnerable computers continuously go online as a worm spreads out. For example, this occurs when a worm propagates over multiple days. To consider the online/offline status of computers, we define $I'(t) = \alpha(t)I(t)$ as the number of online infected hosts; $S'(t) = \alpha(t)S(t)$ as the number of online vulnerable hosts; $N'(t) = \alpha(t)N(t)$ as the number of online hosts among $N(t)$.

To capture the situation where infected hosts are removed, we extend the basic Kermack-McKendrick epidemic model [DG99]. We assume that some infected hosts will be removed from the worm's circulation due to (1) computer crash; (2) patching or disconnecting when users discover the infection. Define $R(t)$ as the number of removed infected hosts at time t. Just as in a Kermack-McKendrick model, we define $\frac{dR(t)}{dt} = \gamma I'(t)$, (where γ is the removal parameter) because in most cases only online infected computers can be removed.

Then the worm propagation dynamics are:

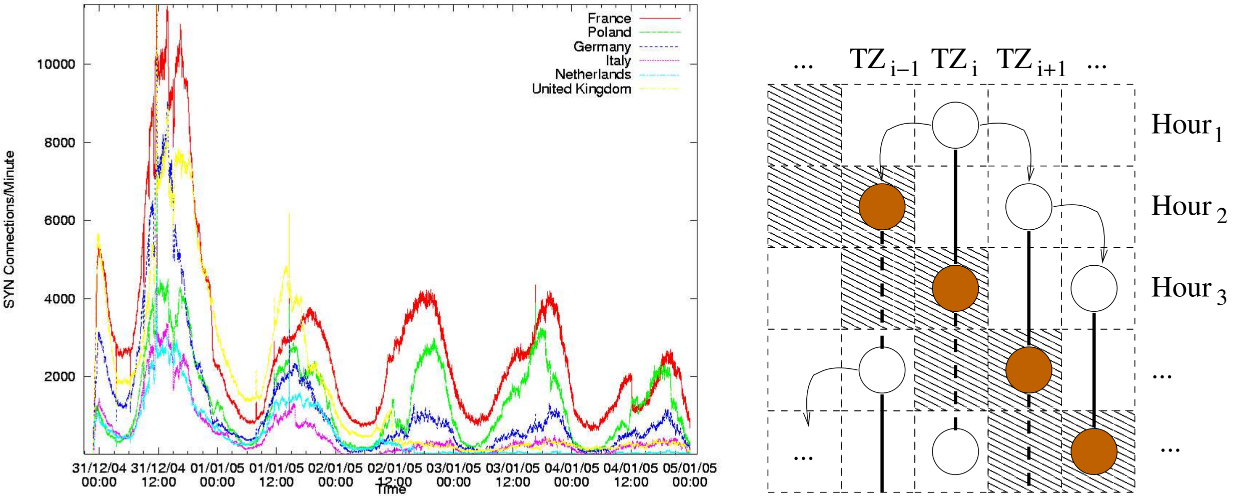
$$\frac{dI(t)}{dt} = \beta I'(t)S'(t) - \frac{dR(t)}{dt} \tag{1}$$

where $S(t) = N(t) - I(t) - R(t)$. β is the pair-wise rate of infection in epidemiology study [DG99]. For Internet worm modeling, $\beta = \eta/\Omega$ [ZTG05] where η is the worm's scanning rate and Ω is the size of the IP space scanned by the worm.

From Eqn. (1), we derive the worm propagation diurnal model:

$$\frac{dI(t)}{dt} = \beta\alpha^2(t)I(t)[N(t) - I(t) - R(t)] - \gamma\alpha(t)I(t) \tag{2}$$

This simple diurnal model can be used to model the propagation of regional viruses or worms. For example, it is



(a) Diurnal Properties

(b) Conceptual Model

Figure 1. (a) Botnet activity by geographic region. (b) General conceptual model of diurnal botnet propagation.

well known that viruses can focus on specific geographic regions [Tre05], e.g., because of the language used in the e-mail propagation system. Similarly, worms can use hard-coded exploits particular to a language-specific version of an OS (e.g., a worm that only successfully attacks Windows XP Home Edition Polish.) For these regional worms, the infection outside of a single zone is negligible and the infection within the zone can be accurately modeled by Eqn. (2).

If we do not consider diurnal effect, i.e., $\alpha(t) \equiv 1$ at any time, then the diurnal model Eqn. (2) is simplified as:

$$\frac{dI(t)}{dt} = \beta I(t)[N(t) - I(t) - R(t)] - \gamma I(t) \qquad (3)$$

This is exactly the traditional Susceptible-Infectious-Removal (SIR) model [DG99].

3.3 Diurnal Model for Multiple Time Zones

Worms are often not limited to a geographic region, however. Some malware contain enormous lookup tables of buffer-overflow offsets for each language edition of Windows [The05b].

Accordingly, we model the worm propagation in the entire Internet across different time zones. Since computers in one time zone could exhibit different diurnal dynamics from the ones in another time zone, we treat computers in each zone as a "group". The Internet can then be modeled as 24 interactive computer groups for ≈ 24 time zones.[1] Since many of the time zones have negligible numbers of computers (such as the zones spanning parts of the Pacific Ocean), we consider worm propagation in K time zones where K is smaller than 24.

Assume $N_i(t)$, $S_i(t)$, $I_i(t)$, $R_i(t)$ as the number of hosts in the time zone i ($i = 1, 2, \cdots, K$) that correspond to $N(t)$, $S(t)$, $I(t)$, $R(t)$ in the previous model Eqn. (2); $\alpha_i(t)$ is the diurnal shaping function for the time zone i; β_{ji} is the pairwise rate of infection from time zone j to time zone i; γ_i is the removal rate of time zone i. Considering the worm infection across different time zones, we can derive the worm propagation for time zone i:

$$\frac{dI_i(t)}{dt} = \sum_{j=1}^{K} \beta_{ji} I'_j(t) S'_i(t) - \frac{dR_i(t)}{dt} \qquad (4)$$

which yields:

$$\begin{aligned} \tfrac{dI_i(t)}{dt} \quad = \quad & \alpha_i(t)[N_i(t) - I_i(t) - R_i(t)] \\ & \cdot \textstyle\sum_{j=1}^{K} \beta_{ji}\alpha_j(t) I_j(t) \\ & - \gamma_i \alpha_i(t) I_i(t) \end{aligned} \qquad (5)$$

For a uniform-scan worm, since it evenly spreads out its scanning traffic to the IP space, $\beta_{ji} = \eta/\Omega, \forall i, j \in K$. For worms that do not uniformly scan the IP space, the authors in [ZTG05] demonstrated that $\beta_{ji} = \eta_{ji}/\Omega_i$ where η_{ji} is the number of scans sent to group i from an infected host in group j in each time unit; and Ω_i is the size of the IP space in group i.

[1] There are more than 24 time zones, but we simplify things for the sake of discussion.

When we discover a new worm propagating in the Internet, we can use the diurnal model Eqn. (5) by inferring the parameter β_{ji} based on monitored honeypot behavior of scanning traffic. As noted above, many honeypot systems can observe all outgoing scans created by a trapped worm [Pro03]. We therefore infer the worm's scanning target address distribution based on reports from multiple honeypots. Then we can derive η_{ji} based on the worm's scanning distribution and rate.

3.4 Model Limitations

There are of course several limitations to our model. First, our diurnal model is not well suited to model worms propagating via email. Unlike scanning worms where malicious codes directly reach victim computers, malicious email are saved in email servers before users retrieve them onto their own computers. When a computer is shut down and its user goes to sleep at night, the malicious email targeting the user is not lost as in the case of scanning worms; the infection effect of these malicious email will show up once the user checks email later. Therefore, the propagation dynamics $I(t)$ at time t will be not only determined by current infection as shown in Eqn. (1), but also determined by previous infection dynamics.

Second, for non-uniform scanning worms, as explained after Eqn. (5), we need to know the worm scan rate and scanning space size in each group (or time-zone) in order to use the multiple time-zone diurnal model Eqn. (5). For this reason, we need to have a sound worm scanning monitoring system in order to use the diurnal model accurately for modeling of non-uniform scanning worms.

3.5 Experiments

We wish to validate our model using empirical data. Further, we wish to explore whether the model can analytically distinguish botnets, based on their short-term propagation potential. We selected a large (350K member) botnet from our collection of observed botnets, since it had the most diverse geographical dispersion of victims. The binary for the botnet was obtained from AV company honeypots, and analysis confirmed that the malware used random scanning for propagation, and a single domain for rallying victims.

Our experiment simplifies the number of time zones to a manageable number. Usually, computers in neighboring time zones have the similar diurnal property — this phenomena has been confirmed by our monitored botnet activities. For example, Figure 1(a) shows European countries with very similar diurnal dynamics. Therefore, it is convenient and accurate to model the Internet as several groups where each group contains several neighboring time zones that have the similar diurnal dynamics.

In the following experiments, we consider three groups of computers because the infected population was mostly distributed in these three groups: North America, Europe, and Asia. The North American group is composed of US, Canada, and Mexico; the European group is composed of European countries; and the Asian group is composed of China, South Korea, Japan and adjacent areas (e.g., Australia). We note that antivirus companies similarly organize Internet monitoring into major groups: Asia, Europe, North America, and so on [Tre05, Ull05].

Figure 2 shows the number of SYN connections sent to the sinkhole per minute by the botnets in each group. The time shown in X-axis is the 00:00UTC time of the labeled date. Since each bot sends out a similar number of SYN connection requests to its botmaster per minute, the number of infected hosts in each group is proportional to the number of SYNs sent from each group. Therefore, the curves in Figure 2 represent the number of online infected computers as time goes on.

As shown in this figure, for the botnet we are studying, the Asian group has about eight times more infected computers than the North American group has (although this is not true for other botnets). In addition, the number of online infected hosts of the Asian group reaches its peak level when this number of the North American group reaches its lowest level since the time difference between these two groups is around 12 hours.

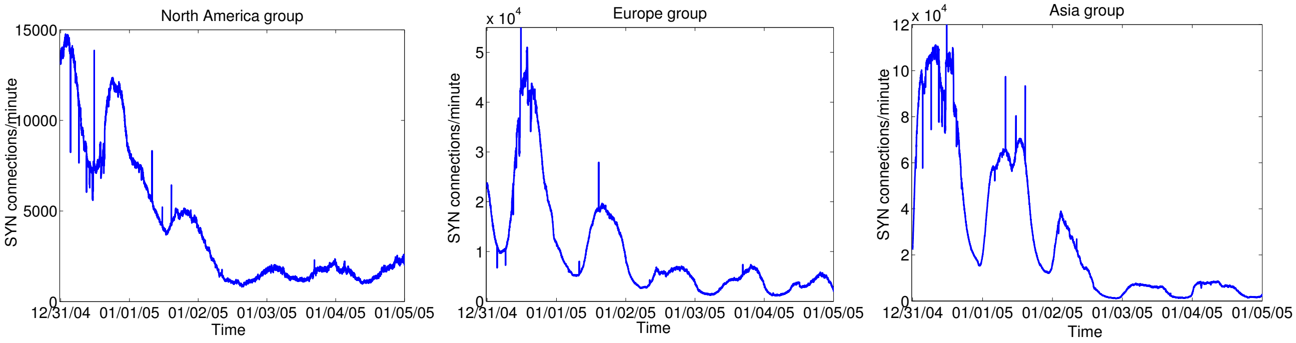
In the following, we study the propagation of a worm based on the diurnal model, Eqn.(5), and the above three groups. For simplicity, we assume the worm uniformly scans the Internet, thus $\beta_{ji} = \eta/\Omega, \forall i, j \in K$. We also assume that all computers in these groups have the same removal rate γ. Since the number of infected hosts is proportional to the number of SYN connections per minute, we choose populations of $N_1 = 15,000$ for the North American group, $N_2 = 45,000$ for the European group, and $N_3 = 110,000$ for the Asian group. Then we deploy Matlab Simulink [Mat05] to derive the numerical solutions for the diurnal model Eqn. (5).

We wrote a program to automatically derive the dynamics $\alpha(t)$ for each group (and also each country). The basic steps for deriving $\alpha(t)$ include:

1. First, observe all botnet traffic, and break down victim membership by geographic region.

2. Second, process the data from a region to derive $\alpha(t)$ through the following steps:

   - Split a monitored dataset into segments for each day. Suppose a monitored dataset spans over n days. Split the dataset into n segments where each segment corresponding to one day containing the data from 00:00:00UTC to 24:00:00UTC in that day.



(a) North America group (b) Europe group (c) Asia group

Figure 2. Number of SYN connections sent to the sinkhole per minute from each group by the botnet

- Normalize the data in each segment so that the maximum value of the data in each segment is one.
- Average the data in all segments to derive a primary $\alpha(t)$;
- In order to remove the monitoring noise, find a polynomial to represent $\alpha(t)$ by minimizing the cumulative square error between the polynomial and the primary $\alpha(t)$ derived in the previous step;
- Normalize the result so that the maximum value of $\alpha(t)$ is one.

The diurnal shaping function $\alpha(t)$ is a periodical function, i.e., $\alpha(0) = \alpha(T)$ where $T = 24$ hours. After the first one or two days, many worms' infected population will drop continuously due to patching and cleaning of infected computers. For this reason, the $\alpha(t)$ derived through the above procedures usually has $\alpha(0) > \alpha(24)$. If this is the case, we need another step to adjust the derived $\alpha(t)$ so that $\alpha(0) = \alpha(24)$. Here we use a heuristic algorithm such that the shape of the $\alpha(t)$ is not distorted much.

3. Third, place the $\alpha(t)$ table and its corresponding vulnerability in a database, keyed by vulnerability.

We followed these steps to derive $\alpha(t)$ for North America, Europe and Asia, as shown in Figure 3(a). Studying the diurnal dynamics of North American group, the time with the fewest computers online is around 11:00 UTC, which is 6:00am in US eastern coast and 3:00am in US western coast. Figure 3(b) shows the cumulative online vulnerable population across all three groups before the worm begins to spread.

Figure 3(a) clearly illustrates the diurnal properties of botnets visually suggested by the SYN activity plot in Figure 1(a). The distinct diurnal behavior of all three time zone groups also shows that combining multiple hour-sized time zones into groups did not make the diurnal patterns indistinguishable from each other.

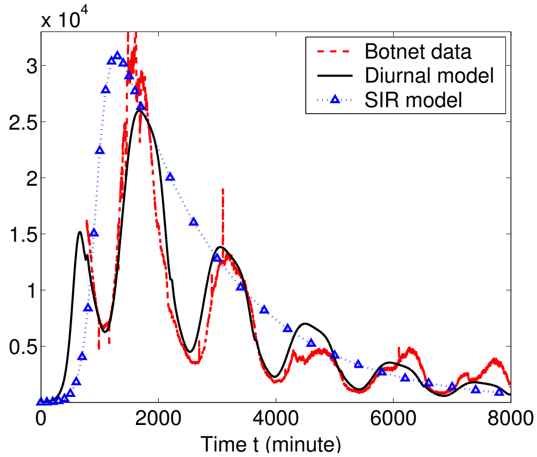
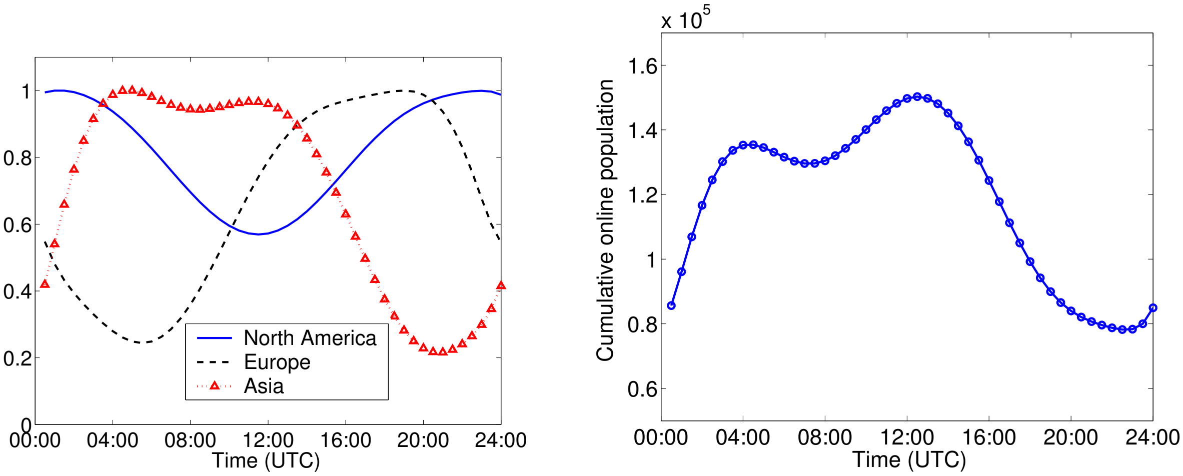


Figure 4. Comparison of models with botnet traffic in the European group

Having derived values for $\alpha(t)$, we can test how well the diurnal model in Eqn. (5) can capture a worm's propagation behavior in the Internet. Figure 4 shows the number of online bot computers in the European group observed by our sinkhole compared with the analytical results from the model Eqn. (5), and the existing SIR model Eqn. (3). At some initial time labeled as time 0 in the figure, the bot began to spread. After a while, the bot was discovered and entered our sinkhole, and our data collection begins. Figure 4 shows that, compared with the SIR model Eqn. (3), the diurnal model Eqn. (5) is much better in capturing the diurnal property of a worm's propagation and the active infective populations in the Internet.



(a) Diurnal dynamics

(b) Cumulative online population

Figure 3. Worm propagation dynamics and population growth

3.6 Practical Uses of Diurnal Models

The diurnal model Eqn. (5) tells us *when* releasing a worm will cause the most severe infection to a region or the entire Internet. For worms that focus on particular regions, the model also lets us predict future propagation, based on time of release. The role that time zones play on propagation is intuitively obvious, but has not been expressed in any previous model.

3.6.1 Forecasting with Pattern Tables

The derived $\alpha_i(t)$ is not limited to the botnet under examination, but instead reflects the type of vulnerability exploited by the botnet. That is, different botnets that both exploit the same vulnerability in Windows 2000 SP2 will likely have similar $N_i(t)$ (and therefore $\alpha(t)$), assuming there are no other region-specific limiting factors. That is, both worms will target the same $S_i(t)$, if there are no differences (e.g., language differences such as Korean versus English language email viruses) that would clearly favor one time zone's population over another.

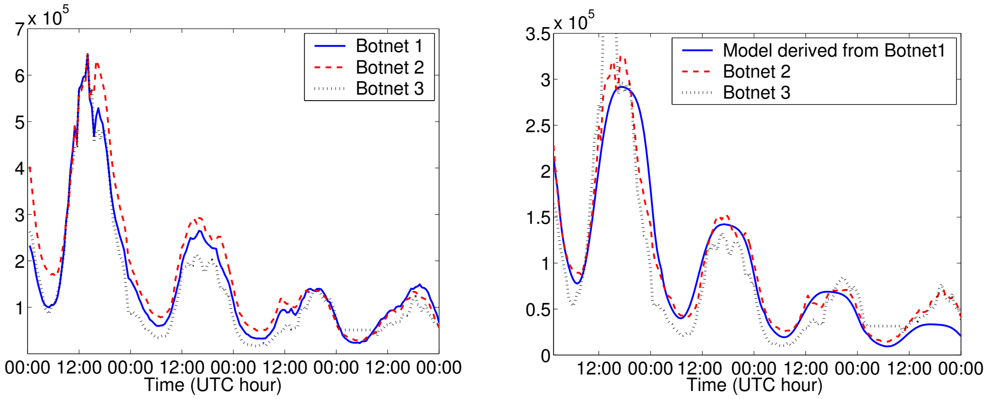
Repeated sampling of botnets using DNS redirection noted in Section 2 (and other techniques) will conceivably yield an understanding of how vulnerabilities are distributed in different zones. Since $\alpha_i(t)$ corresponds to the type of vulnerability being exploited, repeatedly seeing malware target the same OS flaw may assist forecasting. Researchers can infer the growth of future outbreaks based on previous attempts to exploit the same vulnerability. Thus, when a new bot appears targeting a familiar vulnerability, researchers can use timely previous examples to estimate how far and fast the bot will spread.

Accordingly, we can build a table of the derived shaping functions, based on observed botnet data, and key the table based on other heuristics about the worm (e.g., the exploit used, the OS/patch level it affects, country of origin). When a new worm is discovered, these heuristics are often the first few pieces of information learned from a honeypot. One can then consult the table for any prior $\alpha_i(t)$ derivations, and use them to forecast the short-term population growth of the bot, relative to its favored zone and time of release.

To evaluate the forecasting capability of our diurnal model, we collected monitored traces of three botnets that exploited the same vulnerability [Mic04]. The agents for these botnets were released in succession, evidently as enhancements to prior versions. From our discussion in Section 3, these botnets should have similar diurnal shaping functions, $\alpha_i(t)$, for the same time zone or group of zones. We therefore used the diurnal model derived from one botnet to predict the propagation dynamics of other botnets.

Fig. 5(a) shows the propagation dynamics of these three botnets in the European group. Each data point represents the number of SYN connection requests observed by our sinkhole within every half an hour. Because these botnets appeared in different time periods, their infected population were different from each other since the vulnerable population in the Internet varies over time. We therefore show the results by normalizing their SYN connections. Figure 5(a) clearly shows that botnets exploiting the same vulnerability have similar diurnal dynamics. The results of the North American and Asian groups, shown in Figs. 6(a), 7(a), were also similar.

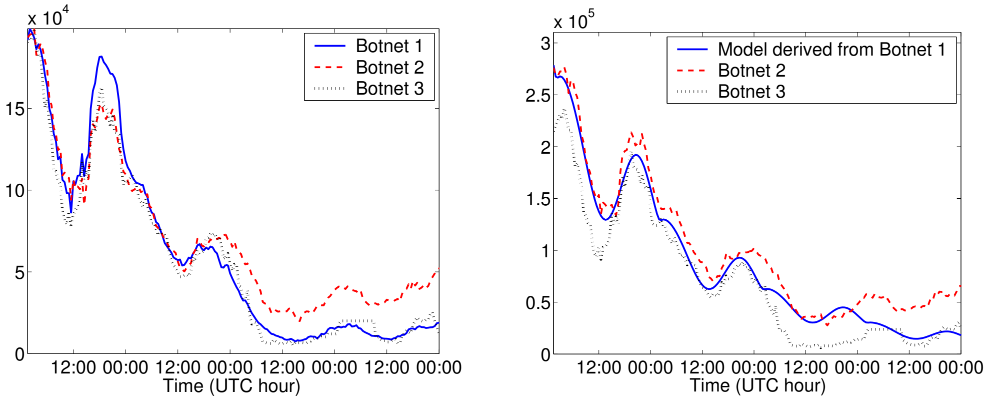
To evaluate the predictive capability of our diurnal model, we derive the parameters for the diurnal model based on curve fitting of data from Botnet 1 for the European group. Then we use the derived diurnal model to predict the dynamics of the other two botnets for the same European



(a) Observed botnet traffic in European group

(b) Predicted and observed behavior in European group

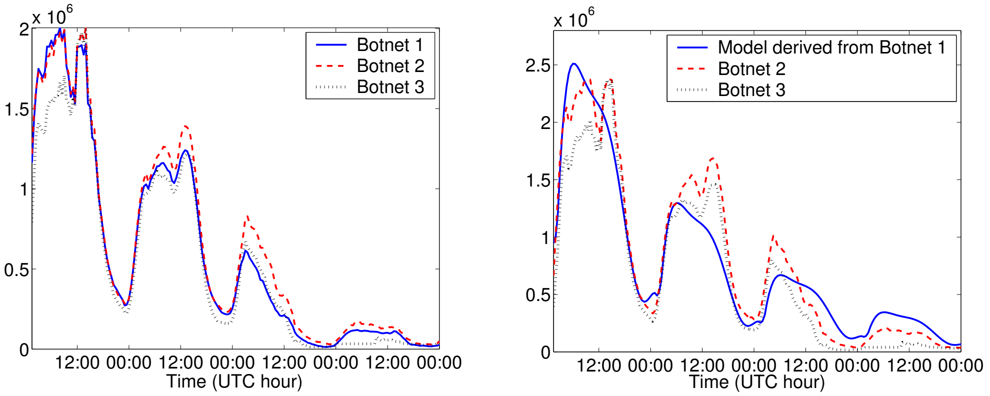
Figure 5. European group



(a) Observed botnet traffic in the North American group

(b) Predicted and observed behavior in North American group

Figure 6. North American group



(a) Observed behavior in Asian group

(b) Predicted and observed behavior in Asian group

Figure 7. Asian group

group. The results are shown in Fig. 5(b). Again, the absolute values of the three curves are normalized to be comparable with each other. This figure shows that we can use the diurnal model to forecast the propagation of botnets using a similar vulnerability. Similar predictions for the North American and Asian groups appear in Figs. 6(b), 7(b). The predictive feature of the diurnal model is not as good as in the European group. Fig. 6(b) shows that the online infected hosts in the North American group is not as smooth as in the European group, and the Botnet 2 infections increased slightly after the first two days instead of dropping. For the Asian group, Fig. 7(b) clearly shows that the first two-days have a different pattern than the third day. We speculate that the North American and Asian groups have more noise because countries in these groups tend to span numerous time zones with large numbers of infected individuals, and China has one time zone for the entire country. By comparison, the European countries tend to occupy a single zone, and most victims are located in the western-most time zones.

As shown in Fig. 5(b), the diurnal model can predict the dynamics of botnets, but not their infected population. (Recall that the model derives $\alpha(t)$ values, which describe the relative fraction of users online.) There are some other ways to predict vulnerable or infected populations for an Internet virus or worm. For example, Zou *et al.* [ZGGT03] presented a method to predict the vulnerable population based on a worm's initial propagation speed and its scan rate η.

We note that the derived diurnal dynamics of a botnet have an unknown shelf life. If a model is derived from a botnet, its predictive power decays over time, since users migrate to new platforms, clean machines, or replace equipment. The botnets studied in the example above all took place within the same 3-week period. Since malware is often released in rapid succession (e.g., version.A, version.B, etc. of the same exploit), long-term changes in victim populations might not affect short-term forecasting. Our data did not permit a longitudinal study of the predictive power of older botnets. Future work will identify factors that affect the validity of derived $\alpha(t)$ values over an extended time period.

Another limiting factor in our model comes from the introduction of additional propagation mechanisms. Many instances of malware, e.g., phatbot [LUR04], spread using many different infection vectors, such as e-mail, random scanning and local exploits. Our model does not address malware that combines additional types of propagation techniques in subsequent releases. Future work will explore techniques to identify dominant propagation mechanisms used in malware, and hybrid models derived from different botnets with distinct $\alpha(t)$ values.

3.6.2 Release Times

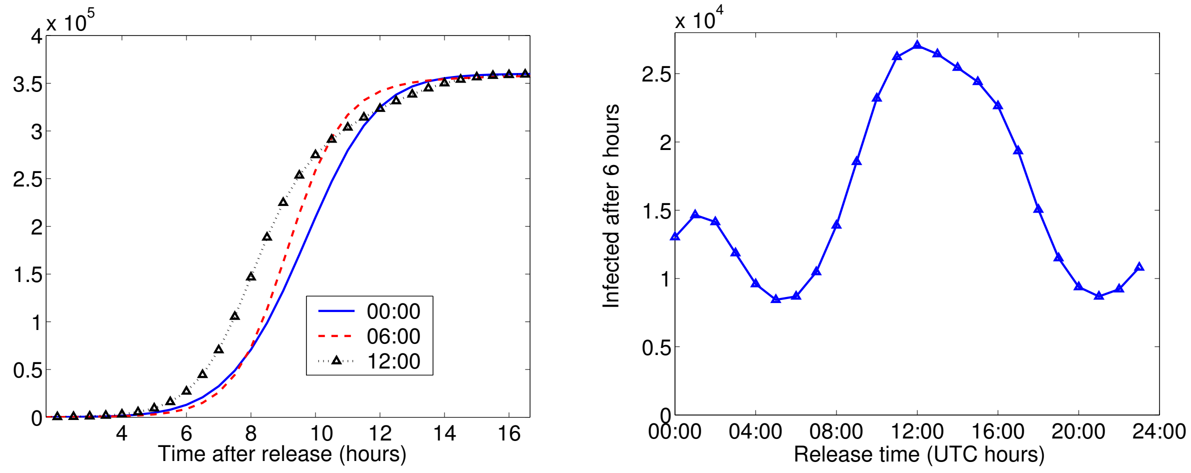
The short-term spread of a worm will vary, depending on the time of release and the distribution of the affected population across different time zones. Knowing the optimal release time for a worm will help us improve surveillance and response. To identify the optimal release time, we perform the following steps:

- Obtain the scan rate η and scanning distribution, and vulnerable population for each zone;
- Obtain the $\alpha(t)$ values for each zone; and
- Using the diurnal model Eqn. (5) to calculate (numerical solution) the infected population six hours after release for different release time to derive the optimal release time.

As an example, we identify an optimal release time in a scenario where the worm uniformly scans the Internet and all three diurnal groups have the same number of vulnerable population, i.e., $N_1 = N_2 = N_3$. The diurnal dynamics of different groups will not matter much for a very slow spreading worm that needs to spread out with at least several days. It also does not matter much for a very fast spreading worm that can finish infecting all online vulnerable hosts within an hour — its infection range is solely determined by the population of current online computers. Therefore, we study the propagation of a middle-speed worm that can spread out in several hours. For example, Code Red is one such worm, which finished its infection in 14 hours [Moo02a]. For this reason, we study a Code Red-like worm that has the total vulnerable population $N_1 + N_2 + N_3 = 360,000$, and $\eta = 358$/min [ZGGT03]. For the purpose of studying worm release time, we assume $\gamma = 0$.

Figure 8(a) shows the propagation of the worm when it is released at 00:00, 06:00 and 12:00 UTC time, respectively. It clearly shows the impact of the diurnal phenomenon on a worm's propagation speed. Refer to the diurnal dynamics shown in Figure 3, the worm released at 12:00 UTC propagates faster than the other worms at the initial stage, because it catches the largest portion of the vulnerable population online in the following several hours. Note that these results are particular to the botnet under consideration, and not all bots. Other botnets will of course have different growth patterns, based on their unique $\alpha(t)$ values.

Figure 8(b) shows the same phenomenon from a different perspective. Here we consider the maximum infected population six hours after a worm is released. (We select six hours as an estimated time required for antivirus or worm monitoring efforts to generate a signature for a new worm [Mar04].) The worm propagates most widely within six hours when it is released around 12:00 UTC, which



(a) Worm propagation under different release time

(b) Number of infected 6 hours after release

Figure 8. Worm propagation when released at different time

is 9:00pm in Tokyo and South Korea, 8:00pm in China, 7:00am in US Eastern. When the botnet starts to grow, it captures some of the evening users in Asia, the mid-day population in Europe, and the early morning users in North America. Six hours later, the Asian population has decreased, but has been substantially replaced by the evening European and mid-day North American users. Thus, by releasing at 12:00 UTC, the worm captures significant portions of all three population groups within six hours.

If we compare the propagation speed when a worm is released at 00:00 UTC and 06:00 UTC, we can see that the worm released at 00:00 UTC propagates faster in the first several hours (as shown in Figure 8(a)). However, it will slow down its infection speed and infects slower than the other one after 8 hours.

This interesting observation has important implications for network administrators. Suppose two worms break out, with the similar infection ability and diurnal properties, and are released at 00:00 and 06:00 UTC, respectively. We notice the spread of the 00:00 worm seems more rapid at first than the other one. (We might observe this by witnessing lots of sensor alerts). Just using η or an alert rate, we might conclude that somehow this worm is spreading rapidly, and is more urgent. So we might want to prioritize response over the 06:00 worm. But, if we know both worms have a similar diurnal property, we know that the 06:00 worm is a higher priority, even though it is spreading at a slightly slower rate in the first few hours.

Being able to distinguish worms based on their optimal release times is useful to security researchers. For example, it can better determine the defense priority for two viruses or worms released in sequence. As noted, malware often goes through generational releases, e.g., worm.A and worm.B, where the malware author improves the code or adds features in each new release. The diurnal model lets us critically consider the significance of code changes that affect $S(t)$ (the susceptible population). For example, if worm.A locally affects Asia, and worm.B then adds a new feature that also affects European users, there clearly is an increase in its overall $S(t)$, and worm.B might become a higher priority. But *when* worm.B comes out, relative to when worm.A started, plays an important role. For example, if the European users are in a diurnal low phase, then the new features in worm.B do not pose an immediate near-term threat. In such a case, worm.A could still pose the greater threat, since it has already spread for several hours. On the other hand, if worm.B is released at a time when the European countries are in an upward diurnal phase, then worm.B could potentially overtake worm.A with the addition of the new victims. The diurnal model exposes this non-obvious result.

Our model lets researchers calculate optimal release times for worms and therefore rank them based on predicted short-term growth rates. We note worm writers cannot similarly use the model to maximize the short-term spread of their malware. Being able to calculate the appropriate time of day to maximize an infection requires the botmaster to know the diurnal shaping function for each time zone. Worm writers might know η, and other important variables in Eqn. (5). But $\alpha(t)$ is necessary to find an optimal release time, and is hard to know. In effect, worm writers would have to create their own distributed monitoring projects like [Ull05, YBJ04, Par04] to accurately derive diurnal shaping functions for selected regions. In this respect, administrators potentially have one advantage over botmasters. Appropriate detection and response technologies can leverage this knowledge.

4 Related Work

Botnets are a fairly new topic for researchers, but have been around for almost a decade [CJ05]. Some work focuses on the symptoms caused by botnets instead of the networks themselves. In [KKJB05], the authors designed sets of Turing tests (puzzles) that users must solve to access over-taxed resources. We further distinguish our work from the extensive literature on DDoS traceback and detection, [MVS01], in that our approach attempts to predict botnet dynamics *before* they launch attacks.

A few researchers have noted techniques for detecting bots using basic misuse detection systems [Han04], and IRC traces [Bru03]. These investigations focus on tracking *individual* bots (e.g., to obtain a binary), while ours focuses on capturing the *network* cloud of coordinated attackers. The only other research directly on countering bot*nets* (as opposed to individual bots) is [FHW05]. The authors in [FHW05] use honeypots to infiltrate the C&C network of botnets.

Our modeling work is part of a long line of computer virus propagation studies. In [TAC98], the authors presented models for the spread of viruses and trojans. Epidemic modeling of viruses was discussed in [KW91], and later in [MSVS03, WW03]. Models have also been proposed for a few famous worms, including CodeRed [ZGT02, Moo02a, Sta01] and Slammer [MPS+03]. In [ZTG04], the authors noted the need to create new models that capture new transmission capabilities (e.g., email) used by worms.

Our study of diurnal behavior in malware has implications for research into worm epidemics. In [MVS05], the authors speculated about the ability of worms to halt spreading (and thereby become more stealthy) after sensing that the vulnerable population had saturated. The pronounced diurnal behavior we noted suggests that self-stopping worms may become mislead about the absence of victims online, particularly if their spread time is less than one diurnal phase (i.e., than 24 hours).

A significant early work on botnets is [CJ05], which notes the centralized control structures used for data collection in Section 2. We agree with [CJ05] centralized botnet C&C is not always guaranteed, and more research is needed. Our model tracks propagation, and is orthogonal to this view.

Bots are often special purpose worms, and so our work relies on much of the existing worm literature. The utility of our model assumes administrators can detect and analyze worms in a somewhat automated fashion to derive the scanning rate and identify the target vulnerability. We have not discussed this in detail, because tools like honeyd [Pro03] and others [YBP05, DQG+04] have convincingly demonstrated the required detection capability.

Biological models of epidemics have of course noted the importance of dormancy in propagation [DH00]. This corresponds to the diurnal factors in our model, which models night-time as a form of limited natural quarantine or dormancy in the malware. Similarly, biological models have noted the importance of spatial dispersion, demography, and other other categorical factors in propagation [DG99]. To a limited extent, this corresponds to the role played by zones (geographic location) in our time zone model. Computer models of malware, and our model in particular, are different from these approaches, since contact is not restricted in a computer network, and transmission may occur between any peers on the Internet.

5 Conclusion

Botnets will continue to grow and evolve, and the research community needs to keep pace. Time zones play an important role in botnet growth dynamics, and factors such as time-of-release are important to short-term spread rates.

The data we observed in our sinkhole revealed the importance of time zones and time of day, and motivated the creation of a diurnal model. The model was more accurate than the basic SIR models currently used, and accurately predicted botnet population growth. Further, knowledge of the diurnal shaping functions lets one identify release times that maximize malware. This allows one to compare two given botnets, and priority rank them based on short-term propagation potential. Since deriving the diurnal shaping function ($\alpha(t)$) for each time zone requires extensive data collection, botmasters are unlikely to accurately predict optimal release times.

5.1 Future Work

Our future work will also extend the diurnal model to address email spreading viruses. By studying the rate of propagation and new victim recruitment observed in sinkhole studies, we hope to derive a more accurate model of email virus propagation. We will also identify new techniques to sample botnet populations, so that we can further study botnets that do not use centralize C&C systems.

Our work so far has identified time zone and time of release as two key factors in short-term virus propagation. We plan to investigate other possible variables, such as the mix of operating systems, hot patch levels, and the mix of applications used on infected systems.

Acknowledgments

This work is supported in part by NSF grant CCR-0133629 and Office of Naval Research grant N000140410735. The contents of this work are solely the responsibility of the authors and do not necessarily represent the official views of NSF and the U.S. Navy. The authors would like to thank the anonymous reviewers for helpful comments and the shepherd of this paper Professor Fabian Monrose at The Johns Hopkins University for very valuable suggestions.

References

[Bru03] David Brumley. Tracking hackers on IRC. `http://www.doomdead.com/texts/ircmirc/TrackingHackersonIRC.htm`, 2003.

[Cip05] CipherTrust. Ciphertrust's zombiemeter. `http://www.ciphertrust.com/resources/statistics/zombie.php`, 2005.

[CJ05] Evan Cooke and Farnam Jahanian. The zombie roundup: Understanding, detecting, and disrupting botnets. In *Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI '05)*, 2005.

[Coo03] Cooperative Association for Internet Data Analysis (CAIDA). Netgeo - the Internet geographic database. `http://www.caida.org/tools/utilities/netgeo/`, 2003.

[Dag05] David Dagon. The network is the infection. `http://www.caida.org/projects/oarc/200507/slides/oarc0507-Dagon.pdf`, 2005.

[DG99] D.J. Daley and J. Gani. *Epidemic Modeling: An Introduction*. Cambridge University Press, 1999.

[DH00] O. Diekmann and J.A. P. Heesterbeek. *Mathematical Epidemioloogy of Infection Diseases*. John Wiley and Sons, 2000.

[DQG$^+$04] David Dagon, Xinzhou Qin, Guofei Gu, Wenke Lee, Julian Grizzard, John Levine, and Henry Owen. Honeystat: Local worm detection using honeypots. In *International Symposium on Recent Advances in Intrusion Detection (RAID)*, 2004.

[FHW05] Felix C. Freiling, Thorsten Holz, and Georg Wicherski. Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks. Technical Report ISSN-0935-3232, RWTH Aachen, April 2005.

[Han04] Christopher Hanna. Using snort to detect rogue IRC bot programs. Technical report, October 2004.

[Har02] John D. Hardin. The scanner tarpit howto. `http://www.impsec.org/linux/security/scanner-tarpit.html`, 2002.

[Hol05] Thorsten Holz. Anti-honeypot technology. `www.ccc.de/congress/2004/fahrplan/files/208-anti-honeypot-technology-sl%ides.pdf`, 2005.

[JQsx05] Jibz, Qwerton, snaker, and xineohP. Peid. `http://peid.has.it/`, 2005.

[KKJB05] Srikanth Kandula, Dina Katabi, Matthias Jacob, and Arthur W. Berger. Botz-4-sale: Surviving organized ddos attacks that mimic flash crowds. In *2nd Symposium on Networked Systems Design and Implementation (NSDI)*, May 2005.

[KRD04] Jonghyun Kim, Sridhar Radhakrishnan, and Sudarshan K. Dhall. Measurement and analysis of worm propagation on Internet network topology. In *IEEE International Conference on Computer Communications and Networks (ICCN'04)*, 2004.

[Kre03] Christian Kreibich. Honeycomb automated ids signature creation using honeypots, 2003. `http://www.cl.cam.ac.uk/˜cpk25/honeycomb/`.

[KW91] J.O. Kephart and S.R. White. Directed-graph epidemiological models of computer viruses. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 343–359, 1991.

[Lis01] T. Liston. Welcome to my tarpit - the tactical and strategic use of labrea. `http://www.hackbusters.net/LaBrea/LaBrea.txt`, 2001.

[LUR04] LURHQ. Phatbot trojan analysis. http://www.lurhq.com/phatbot.html, 2004.

[Mar04] Andreas Marx. Outbreak response times: Putting av to the test. *Virus Bulletin*, pages 4–6, February 2004.

[Mat05] Mathworks Inc. Simulink. http://www.mathworks.com/products/simulink, 2005.

[Mic04] Microsoft, Inc. Microsoft security bulletin ms04-011 security update for microsoft windows (835732). http://www.microsoft.com/technet/security/Bulletin/MS04-011.mspx, 2004.

[Mic05] George Michaelson. Rir delegation reports and address-by-economy measures. http://www.caida.org/projects/oarc/200507/slides/oarc0507-Michaelson.pdf, 2005.

[Moo02a] D. Moore. Code-red: A case study on the spread and victims of an Internet worm. http://www.icir.org/vern/imw-2002/imw2002-papers/209.ps.gz, 2002.

[Moo02b] D. Moore. Network telescopes: Observing small or distant security events. http://www.caida.org/outreach/presentations/2002/usenix_sec/, 2002.

[MPS+03] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer worm. *IEEE Magazine on Security and Privacy*, 1(4), July 2003.

[MSVS03] D. Moore, C. Shannon, G. M. Voelker, and S. Savage. Internet quarantine: Requirements for containing self-propagating code. In *Proceedings of the IEEE INFOCOM 2003*, March 2003.

[MVS01] David Moore, Geoffrey Voelker, and Stefan Savage. Inferring Internet denial-of-service activity. In *Proceedings of the 2001 USENIX Security Symposium*, 2001.

[MVS05] Justin Ma, Geoffrey M. Voelker, and Stefan Savage. Self-stopping worms. In *WORM'05: Proceedings of the 2005 ACM workshop on Rapid Malcode*, 2005.

[Par04] Janak J Parekh. Columbia ids worminator project. http://worminator.cs.columbia.edu/, 2004.

[PBS+04] Phillip Porras, Linda Briesemeister, Keith Skinner, Karl Levitt, Jeff Rowe, and Allen Ting. A hybrid quarantine defense. In *Workshop on Rapid Malcode (WORM)*, 2004.

[Pro03] Niels Provos. A virtual honeypot framework, 2003. http://www.citi.umich.edu/techreports/reports/citi-tr-03-1.pdf.

[Pro05] Honeynet Project. Know your enemy: Honeynets. http://project.honeynet.org/papers/honeynet/, 2005.

[RIP05] RIPE NCC. DISI Tools. http://www.ripe.net/projects/disi/code.html, 2005.

[SM04] Colleen Shannon and David Moore. The spread of the witty worm. *Security & Privacy Magazine*, 2(4):46–50, 2004.

[Spi03] Lance Spitzner. *Honeypots: Tracking Hackers*. Addison Wesley, 2003.

[SS03] S.E. Schechter and M.D. Smith. Access for sale. In *2003 ACM Workshop on Rapid Malcode (WORM'03)*. ACM SIGSAC, October 2003.

[Sta01] S. Staniford. Code red analysis pages: July infestation analysis, 2001. http://www.silicondefense.com/cr/july.html.

[TAC98] Harold Thimbleby, Stuart Anderson, and Paul Cairns. A framework for modelling trojans and computer viruses. *The Computer Journal*, 41(7):445–458, 1998.

[The05a] The Honeynet Project and Research Alliance. Know your enemy: Tracking botnets. http://www.honeynet.org/papers/bots/, 2005.

[The05b] The Metasploit Project. Metasploit. http://www.metasploit.com/, 2005.

[Tre05] Trend Micro. Regional breakdown. http://wtc.trendmicro.com/wtc/report.asp, 2005.

[Ull05] Johannes Ullrich. Distributed intrusion detection system (dshield). http://www.dshield.org/, 2005.

[WPSC03] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham. A taxonomy of computer worms. In *2003 ACM Workshop on Rapid Malcode (WORM'03)*. ACM SIGSAC, October 2003.

[WSP04] Nicholas Weaver, Stuart Staniford, and Vern Paxson. Very fast containment of scanning worms. In *Proceedings of the 13th Usenix Security Conference*, 2004.

[WW03] Yang Wang and Chenxi Wang. Modeling the effects of timing parameters on virus propagation. In *Proceedings of ACM CCS Workshop on Rapid Malcode (WORM'03)*, October 2003.

[YBJ04] Vinod Yegneswaran, Paul Barford, and Somesh Jha. Global intrusion detection in the domino overlay system. In *Proceedings of NDSS*, 2004.

[YBP05] Vinod Yegneswaran, Paul Barford, and Dave Plonka. On the design and utility of Internet sinks for network abuse monitoring. In *In Proceedings of Symposium on Recent Advances in Intrusion Detection (RAID'04)*, 2005.

[ZGGT03] C. C. Zou, L. Gao, W. Gong, and D. Towsley. Monitoring and early warning for Internet worms. In *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*, October 2003.

[ZGT02] C. C. Zou, W. Gong, and D. Towsley. Code red worm propagation modeling and analysis. In *Proceedings of 9th ACM Conference on Computer and Communications Security (CCS'02)*, October 2002.

[ZGT03] C. C. Zou, W. Gong, and D. Towsley. Worm propagation modeling and analysis under dynamic quarantine defense. In *Proceedings of ACM CCS Workshop on Rapid Malcode (WORM'03)*, October 2003.

[ZTG04] Cliff C. Zou, Don Towsley, and Weibo Gong. Email worm modeling and defense. In *13th International Conference on Computer Communications and Networks (ICCCN'04)*, October 2004.

[ZTG05] C.C. Zou, D. Towsley, and W. Gong. On the performance of Internet worm scanning strategies. *Elsevier Journal of Performance Evaluation*, 2005. (to appear).

[ZTGC05] Cliff C. Zou, Don Towsley, Weibo Gong, and Songlin Cai. Routing worm: A fast, selective attack worm based on ip address information. June 2005.