Hierarchical Models for Indirect Observation in Botnet Population Estimation: A Case Study Using Conficker-C

Rhiannon Weaver
rweaver@cert.org

July 29, 2010

1 Introduction

This proposal describes a model for the hourly number of peer-to-peer connection requests generated by a single machine in the Conficker-C botnet, as viewed by an observer who can monitor a fixed proportion of Internet address space. The goal of developing this peer-to-peer behavioral model is to estimate the number of infected machines in the botnet when machines cannot be directly observed. Network behavior is often observed through the filter of IP addresses, where the mapping between machines and addresses is not one-to-one. Because all machines in the botnet are infected with the same malicious code base, their malicious behavior can act as a more stable representative for a single machine than an IP address.

Parameter estimation for a single infected machine can be achieved using informed priors and MCMC methods with Metropolis-within-Gibbs sampling to account for time-dependent heterogeneity in connection rates. For population estimation in the presence of indirect observation, a *confusion matrix* is introduced, that can be used to represent different mappings of machines to IP addresses commonly used in network infrastructure. We propose MCMC methods for addressing estimation of single-host parameters and population hyperparameters, as well as population estimation across a large set of independent networks.

Section 1.1 and Section 1.2 motivate the population study and give an overview of the main challenges for applying traditional population estimation methodology to network phenomena. Section 2 introduces the Conficker-C population and the methods for observing it in more detail. Section 3 describes the model for observing a single infected machine, and introduces the framework for extending the single-host model to a network model. Section 4 describes data collection and proposes the steps to be taken for the completion of the dissertation.

1.1 Botnets and Conficker-C

A botnet is a collection of computers that have been compromised by a malicious software (malware) program, putting them all under the control of a single malicious operator or small group of operators. These operators, or “bot herders,” can use an army of machines for a variety of purposes, such as gathering intelligence about other networks through scanning, sending spam email on a large scale, or crippling local servers or even national network infrastructures by distributed Denial of Service (DDoS) attacks that throttle network communications with millions of illegitimate connection requests. Tracking botnet size is important in order to understand the scope and spread of an infection. Security analysts rely on population estimates to prioritize threats and to measure the efficacy of clean-up strategies, but the methods and differences among published numbers are often substantial enough to leave analysts unsure of which numbers are realistic or trustworthy.

Conficker (also known as Downadup) is a malware program that first appeared in October of 2008. Conficker targets machines running Microsoft Windows operating systems. By January of 2009, estimates of the number of Conficker-infected machines ranged between 9 million and 15 million, as reported by security companies and news outlets. Estimation methodology varies among researchers, and estimates are often published as daily raw counts with no measures of uncertainty. Little attempt has been made to simultaneously incorporate seasonal effects and variability due to hosts' availability, as well as rates of new infections or cleaned machines with time, into Conficker population estimates.

The Conficker-C variant that propagated through the botnet in March 2009 introduced a specific pattern of decentralized peer-to-peer (P2P) activity in its communications. When a host infected with Conficker-C comes online, it initiates a search for peers by randomly scanning Internet address space. Several independent researchers developed a behavioral signature that identifies Conficker-C P2P connection requests with high reliability in large-scale summary information of network traffic called *network flow data* (Claffy et al., 1989). The visibility of Conficker-C in this data set gives us a unique window into its behavior, population size, and birth and death rates.

1.2 Population estimation and network threats

Statistical population estimation is based on mark-recapture models and their extensions to a wide class of generalized linear models (Fienberg et al., 1999). In network analysis, simple mark-recapture techniques, which reduce to counting intersections among overlapping sets, have been applied to study botnet populations (Chan and Hamdi, 2003; Horowitz and Malkhi, 2003; Li et al., 2009), as well as to other phenomena such as peer-to-peer file sharing networks (Mane et al., 2005; Psaltoulis et al., 2005). But this technique, an application of log-linear models, is valid only for closed populations with direct observation of individuals of interest, and equal probability of capture for all individuals. Internet phenomena often violate these assumptions.

Extending mark-recapture models to open populations is widely addressed in the literature (Schwarz and Arnason, 1996), but botnets often admit several complications of direct observation:

- Individuals are marked based on their observable behavior over a network. Both behavior and observability are often stochastic in nature.

- Though populations are often reported in terms of the number of infected machines, these machines are distinguishable only by the network connection points, called *IP addresses*, through which they communicate, and the link between machines and IP addresses is not one-to-one.

Applying mark-recapture models to machines, as opposed to IP addresses, requires a model that links behavior by address to behavior by machine. On the other hand, applying mark-recapture models directly to IP addresses introduces heterogeneity among individuals; for example, an address shared in parallel by 100 infected computers is observed if at least one of its underlying hosts is observed, whereas an address leased serially to a network of one infected machine and 99 clean machines is observed only if it is currently allocated to the infected machine, and the machine's behavior is observed.

Dupuis and Schwarz (2007) present a solution for the case when heterogeneity can be modeled as a series of observable, nominal classes. In this case, posterior inference is based on multinomial models arising from categorical heterogenous classes and the binary response associated with marked or unmarked animals at each sampling period. But heterogeneity in botnet behavior often arises from variations in continuous, latent behavioral traits such as scan rates. Though machines can be recorded as simply marked or unmarked in any sampling period, this marking loses information regarding the stochastic behavior that is used to identify individuals. We not only observe an individual y_i, but we observe it because it is performing some stochastic action $x_{it} \sim f$, where f is a (possibly parametric) pdf or pmf for the sampling period t, and x_{it} is correlated across sampling periods $t, t+1, t+2,$ and so forth. This data admits a sparser multinomial model than the Jolly-Seber model described by Dupuis and Schwarz, due to a more complex set of observable categories.

2 Observing Conficker-C Machines

Unlike wildlife studies, where individuals are directly observable, machines in a botnet are visible only by their network behavior. Physically, infected machines can be located anywhere in the world, but the malicious applications running on these machines need to communicate over the Internet–with each other or with a central controller–in order for the botnet to survive as a co-ordinated large-scale threat. Thus for botnets, communication methods and signature detection define the space of directly observable behavior. Section 2.1 describes the communication method that Conficker-C hosts use, and the resulting observable artifacts. Section 2.2 describes how an outside observer can use a monitored network to detect Conficker-C machines. Section 2.3 motivates a basic probability model for a single host's P2P scanning process.

2.1 Communication Methods

A set of well-known instructions and rules that defines how software applications communicate with each other is called a *protocol*. Similar to most non-malicious applications in the world, the malicious software behind Conficker-C communicates using either the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP). Both of these protocols also rely on Internet Protocol version 4 (IPv4), that defines the recognizable *IP address* for network communication.

An IP address is a 32-bit integer that identifies a machine to others in a network, for the duration of a connection. IP addresses are generally represented in the dotted decimal format of four octets ranging from 0 to 255 (for example, "192.168.12.2"). In any network connection, the initiator (or source) specifies its own IP address, and specifies an intended target (or destination) IP address with which to establish a connection.

Both TCP and UDP protocols also require that a *port* is associated with both the source and destination IP addresses in a network connection. A port is an integer between 0 and 65535 that is used as a channel to disambiguate connections from multiple applications between the same communicating hosts.

Often, network traffic monitors cannot identify infected machines directly, but they can identify IP addresses, protocols and ports through which infected machines are communicating. Like people and "snail mail" addresses, the map between machines and IP addresses is not one-to-one. Two widely adopted network administration practices complicate the relationship between IP addresses and individual machines:

- Network Address Translation (NAT, one-to-many): A network of many machines is configured to access the Internet through a single machine with one external-facing IP address, often called a *gateway* or *proxy*. Gateway traffic is also often shuffled among two or more IP addresses over time to balance bandwidth across several assets, a technique known as *load-balancing*.

- Dynamic Host Configuration Protocol (DHCP, many-to-one): Internet service providers (ISPs) often have a pool of IP addresses, any one of which can be provided dynamically to a machine via a temporary lease. Depending on the network configuration, DHCP leases can be valid for hours or days. One-day leases are common.

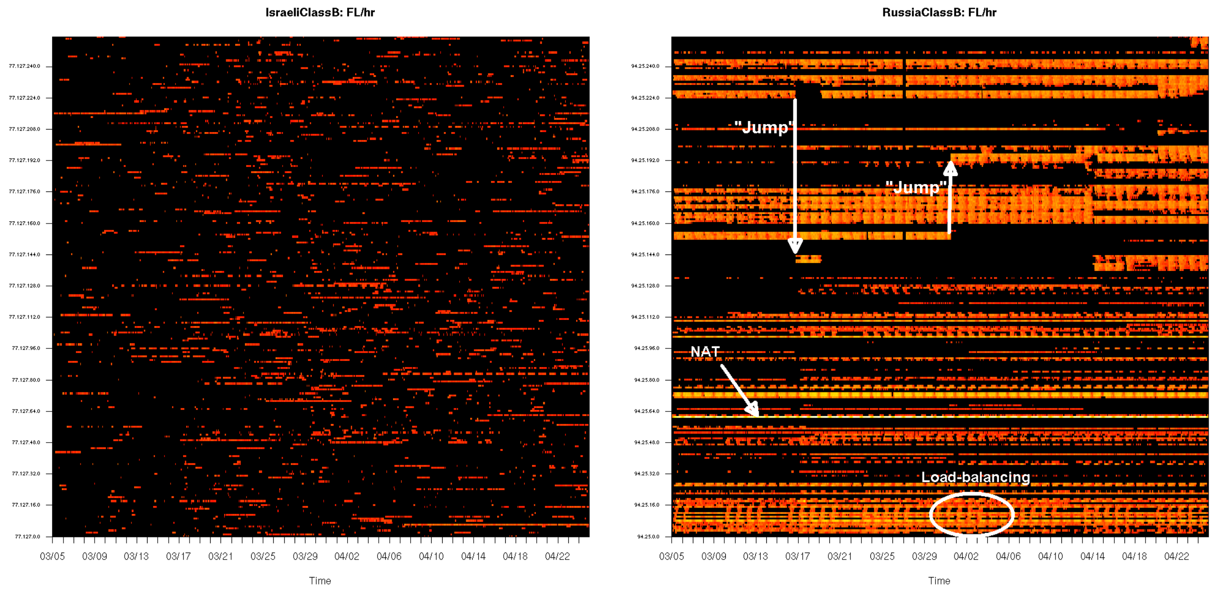

Figure 1: Connection attempts per hour originiating from a /16 net block allocated to an Israieli ISP (left) and a Russian Telecom company (right). Each row represents one of the 256 /24 net blocks that comprise the /16 network. Time ranges from March 5 through April 24, 2009.

Networks of IP addresses are usually allocated in contiguous blocks to entities such as companies, countries, academic institutions or Internet service providers. These blocks can range widely in size. A */N net block* (read "slash N net block") is the collection of addresses obtained by fixing the first N bits of an IP address. Commonly, net blocks are sized by octets. A /24 net block, denoted for example by $192.168.12.0/24$, collects together the 256 IP addresses in the last octet range 0 through 255. A /16 net block, denoted for example by $192.168.0.0/16$, collects together 256^2 or 65536 IP addresses.

2.2 Passive Scan Detection

One way that infected hosts maintain contact in the Conficker-C botnet is by peer-to-peer (P2P) communication. Each host keeps a list of up to 2048 peers that it occasionally connects with, looking for software updates or propagation of instructions from a control channel. But an infected host also has a method for bootstrapping new addresses into its peer list, in case its own list is lost, cleaned, or corrupted.

Whenever an infected host is turned on, with one or more open Internet connections, it surreptitiously uses those connections to continuously scan the Internet. It determines connection ports using an algorithm based on the source IP address and date, which was cracked by several independent researchers (Faber, 2009; Porras et al., 2009a). As a result, Conficker-C P2P requests can be identified with high reliability in large-scale network summary information, such as flow data, with no need to inspect the content of messages.

For the outside observer, Conficker-C hosts are visible (up to IP addresses) when they choose to connect to an IP address that the observer can monitor. Because there is a large population of Conficker-C hosts, and each host generates many requests per hour, an observer monitoring even a relatively small network has a good chance of seeing some randomly generated P2P requests.

An example of observed passive scan data by network is displayed in Figure 1. Color intensity in both figures represents the number of connection requests per hour sent from Conficker-C infected machines in a /16 net block, to uninfected machines located on a large monitored network (approximately 0.15% of IP address space). Each row in the figure corresponds to a /24 net block, with brighter yellow colors indicating higher numbers of connection requests initiated from that net block.

The leftmost figure shows a network allocated to an Israeli ISP. The /24 blocks in this network show patterns that might be expected with single-machine activity per net block. Intensity is uniformly low-scale, with daily trends evident in some blocks, and little correlation of activity between rows. Comparing these patterns to the figure on the right, which is allocated to a Russian Telecom company, we can start to see the effect of administration policies such as NAT, load-balancing, and DHCP. Intensity varies among blocks, with the highest belonging to a single net block ($94.25.61.0/24$) that appears to be acting as a proxy for a comparatively large number of infected hosts. Daily patterns are also evident as users come on- and off-line, but the effect seems to be correlated across multiple adjacent blocks, evidence of DHCP leasing or load-balancing. There are also several places where large blocks of activity appear to "jump"– scans cease abruptly in one location and appear abruptly in another–possibly due to load-balancing or other administrative decisions.

2.3 Observing a single infected machine

Suppose an observer monitors a proportion δ of IP addresses. In one hour t, a single infected machine actively scanning at rate λ_t intitiates M_t random UDP connection requests, of which y_t fall within the monitored network. The quantity M_t is modeled as a Poisson process:

$$M_t \sim \text{Poisson}(\lambda_t)$$

This a reasonable model for small-packet scanning activity programmed at regular intervals. Network traffic is often cited as "bursty" in behavior, but Paxson and Floyd (1995) note that this self-similarity is more common in packet inter-arrival times once connections have been established, as opposed to multiple connection requests. Published experiments with the Conficker-C malware in controlled settings (Porras et al., 2009b) show relatively smooth scanning rates, within both 30-minute and 6-hour time frames.

The Poisson count M_t is assumed to be conditionally independent from M_{t-1} given its rate λ_t. The conditional distribution $\pi(y_t \mid M_t, \delta)$ is Binomial(M_t, δ), which yields a Poisson marginal model for y_t in terms of λ_t and δ:

$$y_t \sim \text{Poisson}(\lambda_t \delta). \tag{1}$$

The proportion δ is measured empirically, and the scan rates are unknown; without loss of generality, the rate λ_t can be modeled as an observed hit rate (subsuming the parameter δ).

In September 2009, Porras et al. (2009c) provided a de-obfuscated reverse engineering of the Conficker-C P2P binary image as it appeared on March 5, 2009. When initiated, the P2P module spawns a UDP scanning process for each valid network connection discovered, in order to bootstrap a peer list of up to 2048 peers. Up to 32 processes can run simultaneously. Each process alternates between a 5-second sleep cycle and a scan phase where it randomly generates a list of up to 100 IP addresses to contact. At each selection, the machine chooses an IP address from its list of n peers with probability equal to

$$Pr(\text{existing peer chosen} \mid n) = \left(1000 - \left\lfloor \frac{950n}{2048} \right\rfloor\right)^{-1}. \tag{2}$$

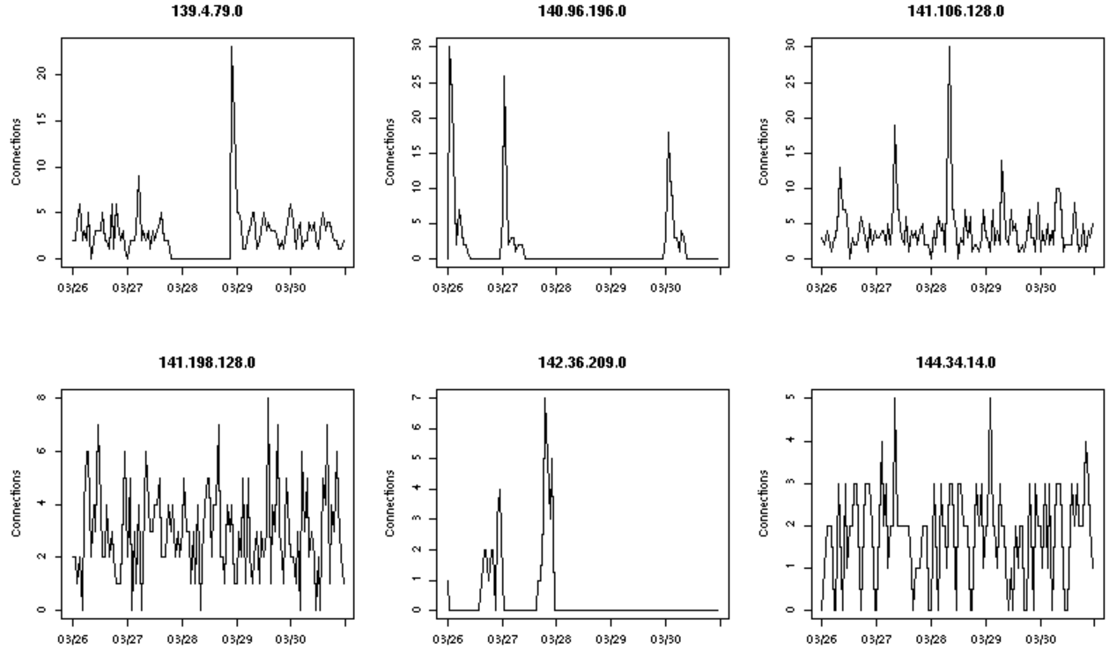

Figure 2: Connection attempts per hour over a 5-day period, from six different IP addresses.

The speed at which UDP connection requests are sent out over the wire depends on the hardware and network capabilities of the infected machine, as well as the amount of bandwidth and drop percentage of the network. The P2P protocol has a maximum of 1200 scanning connection attempts per minute, but observed accounts of Conficker-C P2P scan activity cite lower numbers. Porras et al. (2009b) performed a sandbox test of an infected Conficker-C host with a single network interface and observed scanning rates starting at approximately 1000 to 2000 IP addresses per 5 minute interval, and decreasing over the first two hours of activity to a steady rate of approximately 200 IP addresses per 5 minutes.

Figure 2 shows the number of connection requests per hour sent from six different Conficker-C infected IP addresses, to uninfected machines located in a large monitored network (approximately 0.15% of IP address space), over a five-day period. Several of the series display spikes of activity. It is speculated that each of these time series plots represents a single machine; empirically, low overall scan rates in these net blocks suggest they are not NAT gateways. Each address also resides in an “isolated” section of otherwise dormant (non-scanning) IP addresses that do not appear to be DHCP pools. Based on the information from the reverse engineering and sandbox tests, two reasons for spikes can be postulated:

- Clearing of peer lists: The proportion of random vs. peer-directed connection attempts depends on the size of the machine’s peer list. When a peer list is lost or cleaned (for example after a machine is rebooted), the rate of random scanning will increase due to the effects of equation 2 and gradually decrease as the machine rebuilds its peer list.

- User-driven network connections: Reboots and other user behavior can initiate multiple network connections (for example, requesting IP addresses from servers, downloading Windows updates, or opening web browsing sessions), that cause a multiplicative increase in scanning rates due to the initiation of multiple threads.

3 Formal Modeling

This section outlines the basic model for Conficker-C's observable P2P scanning. The probability model from Section 2.3 is formalized and parameterized in Section 3.1. Section 3.2 derives the likelihood for a single host, and outlines a basic estimation strategy. Section 3.3, Section 3.4 and Section 3.5 outline prior distributions and complete conditional distributions and sampling strategies for the unknown parameters in the single-host model. Section 3.6 presents thoughts and a preliminary strategy for expanding this single-host model to a network model that maps an unknown number of individual machines to observable IP addresses, and suggests three typical NAT and DHCP configurations to model. Section 3.7 discusses strategies for evaluating the model.

3.1 Mathematical model and parameterization

The overall hit rate λ_t at time t can be parameterized by a mixture of a dormant (zero) rate, a baseline rate q that represents the stable rate of connection attempts sent by a machine when it is turned on, a parameter $\omega\ > 1$ that multiplicatively increases the baseline rate during a spike, and a parameter α that controls the geometric decay of spike rates back to the baseline.

Denote by $\mathcal{O}$, $\mathcal{S}$, and $\mathcal{D}$ the set of hours at which the machine is off, spiking, or decaying, respectively. The rate of observed connection attempts λ_t is equal to 0 for all $t \in \mathcal{O}$. Given $t \in \mathcal{S}$, define the spike rate as

$$\lambda_t = q\omega, \quad \omega > 1 \tag{3}$$

For an hour $t \in \mathcal{D}$, define

$$\lambda_t = q + \text{max}[0, \alpha(\lambda_{t-1} - q)], \quad 0 < \alpha < 1 \tag{4}$$

The overall hit rate $\lambda_t^{\mathcal{D}}$ for decay states can also be expressed in terms of the most recent state change. If a decay state directly follows an off state ($t+1 \in \mathcal{D} \mid t \in \mathcal{O}$), then the rate is equal to the baseline rate q until the machine either spikes or is turned off. For any decay state that follows k steps after a spike state with no intervening off states ($t+k \in \mathcal{D} \mid t \in \mathcal{S}, t+1, \cdots t+k-1 \in \mathcal{D}$) , the rate can be expressed as $q(1+\alpha^k(\omega-1))$. A spike state corresponds to $k=0$, resulting in the rate $q\omega$ as described in equation 3.

A discrete 3×3 transition matrix is used to characterize the state changes (spiking, decaying, or off) from hour to hour. Depending on the configuration and use of the infected hosts, rate spikes can appear at relatively regular intervals (for example, approximately every 24 hours, on weekdays only), at sporadic intervals corresponding to rare or unscheduled user events, or as a mixture of both scheduled and unschedued activity. Rate spikes are also more likely to follow periods of inactivity, regardless of the hour at which a machine turns on.

Let $a \in \{o, s, d\}$ be an index representing the state at time t. To capture varying levels of periodicity on a 24-hour scale from host to host, the transition probabilities from each state are defined with a periodic component that controls for amplitude with a scaling parameter $0 \leq \rho_a \leq 1$ and a baseline parameter $\nu_s > 0$, as well as a non-periodic conditional choice parameter $0 \leq \gamma_a \leq 1$. Let t^* be a centered time index ($t^* = t - d$) such that a value of 0 accounts for time zones and aligns the periodicity with the desired hour of the day, for example, aligning troughs with midnight and peaks with noon. Define $p(\rho_a, \nu_a, t^*)$ as a 24-hour periodic probability function parameterized by ρ_a and ν_a relative to t^*:

$$p(\rho_a, \nu_a, t^*) = \frac{\rho_a}{\nu_a + 2} \left[\sin(2\pi t^*/24) + \nu_a + 1\right]$$

The following parameterization is defined for off state transition probabilities $\pi_{\cdot|o}(t) = Pr(t+1 \in \{\mathcal{O}, \mathcal{S}, \mathcal{D}\} \mid t \in \mathcal{O})$:

$$\begin{aligned} \pi_{o|o}(t) &= p(\rho_o, \nu_o, t^*) \\ \pi_{s|o}(t) &= \gamma_o(1 - p(\rho_o, \nu_o, t^*)) \\ \pi_{d|o}(t) &= (1-\gamma_o)(1 - p(\rho_o, \nu_o, t^*)) \end{aligned} \tag{5}$$

This parameterization indicates that the probability of a machine staying off from an off state depends on the time of day, and conditional that the machine turns on at any time, the probability of a spike or simple baseline is independent of time.

For spike states, we assume a time-independent probability of transitioning to a decay state, and time-dependent probabilities of either turning off or spiking again, conditional on a non-decay transition and the hour of the day. This yields the following transition probabilities $\pi_{\cdot|s}(t) = Pr(t+1 \in \{\mathcal{O}, \mathcal{S}, \mathcal{D}\} \mid t \in \mathcal{S})$:

$$\begin{aligned} \pi_{o|s}(t) &= (1-\gamma_s)(1 - p(\rho_s, \nu_s, t^*)) \\ \pi_{s|s}(t) &= (1-\gamma_s)(p(\rho_s, \nu_s, t^*)) \\ \pi_{d|s}(t) &= \gamma_s \end{aligned}$$

This parameterization captures the tendency of user activity to revert toward the stable baseline relative to its initialization, regardless of the hour of the day.

From decay states, machines tend to either spike periodically, or to turn off periodically. If the machine turns off periodically from decay states, then transition probabilties can be parameterized as with transitions from off states (equation 5). Otherwise, a time-dependent transition to a spiking state can be implemented for $\pi_{\cdot|d}(t) = Pr(t+1 \in \{\mathcal{O}, \mathcal{S}, \mathcal{D}\} \mid t \in \mathcal{D})$:

$$\begin{aligned} \pi_{o|d}(t) &= \gamma_d(1 - p(\rho_d, \nu_d, t^*)) \\ \pi_{s|d}(t) &= p(\rho_d, \nu_d, t^*) \\ \pi_{d|d}(t) &= (1-\gamma_d)(1 - p(\rho_o, \nu_o, t^*)) \end{aligned}$$

3.2 Likelihood and estimation strategy for a single infected machine

This section develops the likelihood for all observed hours of communication for a single infected machine. For brevity, the model is restricted to the set of hours for which the machine is known to be infected. New infections and cleanup (births and deaths) are addressed in the dissertation work proposed in Section 4.

Let η_t be the state of the machine at time $t : \eta_t \in \{\mathcal{O}, \mathcal{S}, \mathcal{D}\}$, and define $\boldsymbol{\eta} = \{\eta_0 \cdots \eta_T\}$. Let y_t be the observed count at time t, and define $\boldsymbol{y} = \{y_0 \cdots y_T$. Let $\psi = (\rho_{\{o,s,d\}}, \nu_{\{o,s,d\}}, \gamma_{\{o,s,d\}})$ be the collection of the transition probability parameters and $\theta = (q, \omega, \alpha)$ be the collection of rate parameters. The likelihood $\ell(\psi, \theta, \boldsymbol{\eta}; \boldsymbol{y})$ can be written as the product of an initial probability $p(\eta_0)$, state-to-state transition probabilities, and Poisson count probabilities conditional on the rates defined for each underlying state:

$$\ell(\psi, \theta, \boldsymbol{\eta}; \boldsymbol{y}) = p(\eta_0) \prod_{t=1}^{T} p(\eta_t \mid \eta_{t-1}, \psi) \prod_{t=0}^{T} p(y_t \mid \eta_t, \theta)$$

Collecting like terms, let $\mathcal{T}_{s_1|s_2}$ be the set of hours t that transition from state s_1 to state s_2, and define the following sets of lag states relative to spike and decay states:

$\mathcal{L}_B$: This set represents all decay times t that follow an off state with no intervening spike states. At any hour $t \in \mathcal{L}_B$, the infected machine broadcasts at the baseline rate q.

$\mathcal{L}_k, k = 0, 1, 2, ...$: This set represents all states that follow k steps after a spike state, with $k = 0$ indicating spike states themselves. At any hour $t \in \mathcal{L}_k$, the infected machine broadcasts at the decaying rate $q(1 + \alpha^k(\omega - 1))$.

Suppose that K_{max} is the largest lag k observed in the data. The likelihood can then be expressed as the product:

$$\left[\prod_{s_1, s_2 \in \{o,s,d\}} \prod_{t \in \mathcal{T}_{s_1|s_2}} \pi_{s_1|s_2}(t) \right] \left[\prod_{t \in \mathcal{O}} 1_{\{y_t = 0\}} \prod_{t \in \mathcal{L}_B} \frac{e^{-q} q^{y_t}}{y_t!} \prod_{k=0}^{K_{\text{max}}} \prod_{t \in \mathcal{L}_k} \frac{e^{-q(1+\alpha^k(\omega-1))} \left[q(1 + \alpha^k(\omega - 1)) \right]^{y_t}}{y_t!} \right]$$

Posterior distributions can be estimated using Markov Chain Monte Carlo techniques designed for data augmentation (Tanner and Wong, 1987), that iterate between the following steps:

1. Update $\eta_0, \cdots \eta_T$ in blocks between spiking and off states, from the distribution $p(\eta_w, \cdots \eta_{(w+v)} \mid \psi, \theta, \boldsymbol{y})$, using Metropolis-Hastings sampling.

2. Update parameters ψ from the complete conditional distribution $p(\psi \mid \boldsymbol{\eta})$ using a combination of Gibbs sampling and Metropolis-Hastings sampling.

3. Update parameters θ from the complete conditional distribution $p(\theta \mid \boldsymbol{\eta}, \boldsymbol{y})$ using a combination of Gibbs sampling and Metropolis-Hastings sampling.

Step (1) can be considered a data augmentation step, where the states facilitate the parameter updating described in steps (2) and (3). Section 3.3, Section 3.4 and Section 3.5 describe the priors and complete conditionals for the single-host model.

3.3 Priors

The prior $p(\eta_0)$ is modeled as a simple discrete uniform with equal weight for states $\mathcal{O}$, $\mathcal{S}$ and $\mathcal{D}$. The priors for (q, ω, α) and $(\rho_{\{o,s,d\}}, \nu_{\{o,s,d\}}, \gamma_{\{o,s,d\}})$ are chosen as follows:

$$\begin{aligned}
\pi(q) &= \text{Gamma}(\kappa, \tau) \\
\pi(\omega - 1) &= \text{Gamma}(\iota = 1.5, \beta = 15) \\
\pi(\alpha) &= \text{Uniform}(0, 1) \\
\pi(\nu_a) &= \text{Gamma}(\xi_a = 1, \phi_a = 100) \\
\pi(\gamma_a) &= \text{Uniform}(0, 1) \\
\pi(\rho_a) &= \text{Uniform}(0, 1) \\
& a \in \{o, s, d\}
\end{aligned}$$

Uniform priors are chosen for α, γ_a, and ρ_a to account for the possibility of large individual differences among networks. These priors may be replaced with more flexible Beta priors when more prior information about transitions or decay rates is available for a particular network. The parameter ν_a controls the amount of periodicity in the 24-hour cycle, with a value of 0 resulting in the highest peaks and lowest valleys as hours change, and a value of 100 resulting in near independent and identical transition probabilities for each hour. A diffuse Gamma distribution is chosen to highlight the tendency toward periodicity, but also account for individual differences among networks.

The most important prior in terms of estimating population size is the baseline rate q. Prior information for q comes from sandbox tests that measure the connection request rates, and the measurement of the size of the monitored network. This information becomes critical when extending the single-host model to multiple hosts across multiple networks, in order to gain information about a network-specific number of infected machines as opposed to estimating a broadly flexible network-specific baseline rate.

The hyperparameters set for q depend on the size of the monitored network. Porras et. al. measured a connection request rate of approximately 2400 per hour; this yields an approximate baseline hit rate of 3.6 hits per hour for a monitored network comprising a proportion 0.0015 of IP space. The researchers also observed spikes in activity by a factor of 8 to 10 times the baseline rate.

3.4 Complete conditionals and sampling strategy for θ and ψ

Denote by $N_{\mathcal{A}}$ the cardinality of a set $\mathcal{A}$. For transition probabilities, the distributions for off state ($\mathcal{O}$) parameters are shown; distributions for spike and decay state parameters are comparable.

For $q > 0 \quad$:

$$p(q \mid \boldsymbol{\eta}, \boldsymbol{y}, \xi, \phi, \theta) \quad \propto \quad q^{(\xi + \sum_T y_t - 1)} \exp\left[-\frac{q}{\phi}\left[(T - N_{\mathcal{O}}) + \sum_{k=0}^{K_{\max}} N_{\mathcal{L}_k} \alpha^k (\omega - 1)\right]\right]$$

This distribution is the kernel of a Gamma distribution and can be sampled in the chain using a Gibbs step.

For $\omega > 1 \quad$:

$$p(\omega \mid \boldsymbol{\eta}, \boldsymbol{y}, \iota, \beta, \theta) \quad \propto \quad (\omega - 1)^{\iota - 1} \exp\left[\frac{1 - \omega}{\beta} - \omega q \sum_{k=0}^{K_{\max}} N_{\mathcal{L}_k} \alpha^k\right] \prod_{k=0}^{K_{\max}} \Big(1 + \alpha^k (\omega - 1)\Big)^{\sum_{t \in \mathcal{L}_k} y_t}.$$

This distribution can be sampled using a Metropolis-Hastings step with a truncated Gamma proposal distribution.

For $0 < \alpha < 1 \quad$:

$$p(\alpha \mid \boldsymbol{\eta}, \boldsymbol{y}, \theta) \quad \propto \quad \exp\left[-q(\omega - 1) \sum_{k=0}^{K_{\max}} N_{\mathcal{L}_k} \alpha^k\right] \prod_{k=0}^{K_{\max}} \Big(1 + \alpha^k (\omega - 1)\Big)^{\sum_{t \in \mathcal{L}_k} y_t}.$$

This distribution can be sampled using a rejection method on the interval $(0, 1)$,or with a Metropolis-Hastings step with a proposal distribution defined on the interval$[0, 1]$, such as a Beta or a triangular distribution centered at the current value.

For $0 < \rho_o < 1 \quad$:

$$p(\rho_o \mid \boldsymbol{\eta}, \psi) \quad \propto \quad (\rho_o)^{N_{\mathcal{T}_{o|o}}} \prod_{t \in \mathcal{T}_{o|s} \cup \mathcal{T}_{o|d}} \left(1 - \frac{\rho_o}{\nu_o + 2}\left[\sin(2\pi t^*/24) + \nu_o + 1\right]\right).$$

This distribution can be sampled using a rejection method (for example based on a Beta distribution that replaces the periodic probability $p(\rho_o, \nu_o, t^*)$ with the value $\max_{t^*} p(\rho_o, \nu_o, t^*)$), or with a Metropolis-Hastings step with a proposal distribution defined on the interval$[0, 1]$, such as a Beta or a triangular distribution centered at the current value.

For $\nu_o > 1$:

$$\begin{aligned} p(\nu_o \mid \boldsymbol{\eta}, \psi, \xi, \phi) &\propto (\nu_o - 1)^{\xi - 1} \exp\left[\frac{\nu_o - 1}{\phi}\right] \prod_{t \in \mathcal{T}_{o|o}} \frac{\rho_o}{\nu_o + 2}\left[\sin(2\pi t^*/24) + \nu_o + 1\right] \\ &\times \prod_{t \in \mathcal{T}_{o|s} \cup \mathcal{T}_{o|d}} \left(1 - \frac{\rho_o}{\nu_o + 2}\left[\sin(2\pi t^*/24) + \nu_o + 1\right]\right) \end{aligned}$$

This distribution can be sampled using a Metropoilis-Hastings step.

For $0 < \gamma_o < 1$:

$$p(\gamma_o \mid \boldsymbol{\eta}, \psi) \propto (\gamma_o)^{N_{\mathcal{T}_{o|s}}} (1 - \gamma_o)^{N_{\mathcal{T}_{o|d}}}$$

This distribution is the kernel of a Beta distribution, and can be sampled in the chain using a Gibbs step.

3.5 Complete conditional and sampling strategy for η_t

The rate structure $\lambda_1, ..., \lambda_T$ of the time series $y_1, ..., y_T$ is a deterministic function of the placement of spike states and off states. Changing the state for a single hour t, for example from decay to spike, also changes the likelihood of the observed values y_{t+1} until y_{t+v-1}, where v is the next instance of either a spike state or an off state. Because of the strong dependency between rates in decay states, a sequential updating of states one hour at a time (using for example the conditional distribution $Pr(\eta_t \mid \eta_{t-1}, \eta_{t+1}, y_t, \theta, \psi)$) is not feasible. Instead, let b be the current MCMC iteration that requires an update of state values, with corresponding parameter values θ^b and ψ^b, and path parameters $\boldsymbol{\eta}^b$. An updating scheme is suggested as follows for the path parameters at step b:

1. Define a suitable probability distribution $g(t; \theta, \psi, \boldsymbol{\eta})$ over $t \in [0, \cdots, T]$, for example a uniform distribution, or a discrete probability proportional to the residual value $y_t - \lambda_t^b$ obtained from the current set of path parameters and states. Select an hour t to update at step b with probability $g(t; \theta^b, \psi^b, \boldsymbol{\eta}^b)$. Let η_t^b be the current state value at iteration b of the hour t chosen for updating.

2. For any hour t and a state value $a \in \{o, s, d\}$, let $[t, t + v - 1]$ be the interval of hours affected by a change in state at time t. Let s_{t-1} be the state at time $t - 1$ and s_{t+1} be the state at time $t + 1$. Define the function $h(t, a, \boldsymbol{\eta}, \theta, \psi)$ as follows:

$$h(t, a, \boldsymbol{\eta}, \theta, \psi) = \pi_{s_{t-1}|a}(t)\pi_{a|s_t+1}(t + 1) \prod_{i=t}^{t+v-1} p(y_i \mid \lambda_{i-1}, \eta_t = a, \theta), \quad (6)$$

where $\pi_{\cdot|\cdot}(t)$ is the state-to-state transition probability, and $p(y_t \mid \lambda_{t-1}, \eta_t = a, \theta)$ is the Poisson pdf with rate determined by the parameter vector θ and the lag from the most recent spike or off state as determined by setting the state value at time t to a. For a state η_t, let $a^1_{\eta_t}$ and $a^2_{\eta_t}$ be its two complementary states.

For the chosen hour t, draw a candidate value η^*_t from the set $\{a^1_{\eta^b_t}, a^2_{\eta^b_t}\}$ with probability

$$Pr(\eta^*_t = a^i_{\eta^b_t}) \quad = \quad \frac{h(t, a^i_{\eta^b_t}, \boldsymbol{\eta}^b, \theta^b, \psi^b)}{h(t, a^1_{\eta^b_t}, \boldsymbol{\eta}^b, \theta^b, \psi^b) + h(t_b, a^2_{\eta^b_t}, \boldsymbol{\eta}^b, \theta^b, \psi^b)} \tag{7}$$

In cases where y_t is non-zero, this distribution selects the complementary non-off state (spike or decay) with probability 1.

3. Let $\boldsymbol{\eta}^*$ be the set of path parameters described by setting the state $\eta_t = \eta^*_t$. Accept $\eta^*_{t_b}$ according to the Metropolis-Hastings ratio:

$$r \quad = \quad \min\left(1, \frac{g(t; \theta^b, \psi^b, \boldsymbol{\eta}^*)}{g(t; \theta^b, \psi^b, \boldsymbol{\eta}^b)} \frac{\left[h(t, a^1_{\eta^b_t}, \boldsymbol{\eta}^b, \theta^b, \psi^b) + h(t_b, a^2_{\eta^b_t}, \boldsymbol{\eta}^b, \theta^b, \psi^b)\right]}{\left[h(t, a^1_{\eta^*_t}, \boldsymbol{\eta}^*, \theta^b, \psi^b) + h(t_b, a^2_{\eta^*_t}, \boldsymbol{\eta}^*, \theta^b, \psi^b)\right]}\right).$$

This ratio reduces to a basic likelihood ratio (Metropolis step) when the selection of t is uniform and the observation y_t is non-zero.

Any number of path update steps can be performed in sequence for obtaining good mixing rates in the chain.

3.6 Preliminary thoughts on a network model

Suppose H infected machines reside on a network that has a total of M external-facing IP addresses. At each time t,the observations y_{it}, $i = 1, \cdots, H$ are mapped to IP addresses via a $H \times M$ confusion matrix $\mathbf{W}_t$. The index W_{tim} describes the proportion of all traffic from host i that was assigned to IP address m over the time interval t. This interpretation requires that $\sum_{im} W_{tim} = H$ for all t, and that $\sum_m W_{tim} = 1$ for all t and i. The new observations by IP address, z_{mt} are linear combinations of the counts from the original machines:

$$z_{mt} = \sum_{i=1}^{H} W_{tim} y_{it} \tag{8}$$

A confusion matrix $\mathbf{W}_t = I_H$ for all t represents the simplest mapping of one static IP address per machine. A DHCP pool may show structure in the changes of $\mathbf{W}_t$ with t, although it is unlikely to map more than one machine at a time to an IP address. It is also unusual for a single host to repeatedly obtain multiple DHCP addresses (eg, more than 2 or 3) per hour. A single NAT proxy would be represented as a $H \times 1$ unit vector for $\mathbf{W}_t$ so that $z_t = \sum_i y_{it}$.

The network model adds a layer of obfuscation to the single-host model. For H machines behind a single network, the joint likelihood of observations z_{mt} and machine parameters is still a product of transition probabilities and Poisson probabilities:

$$\begin{aligned} \ell(\psi_1 \cdots \psi_H, \theta_1 \cdots \theta_H, \boldsymbol{\eta}_1 \cdots \boldsymbol{\eta}_H; \boldsymbol{z}) \quad &= \quad \left[\prod_{i=1}^{H} p(\eta_{i0}) \prod_{t=1}^{T} p(\eta_{it} \mid \eta_{it-1}, \psi_i)\right] \\ &\times \quad \prod_{t=0}^{T} \left[\prod_{m=1}^{M} \frac{\exp\left[\sum_{i=1}^{H} W_{tim} \lambda_{it}\right] \left(\sum_{i=1}^{H} W_{tim} \lambda_{it}\right)^{z_{mt}}}{z_{mt}!}\right] \end{aligned}$$

In the multi-host network, machine-specific transition probabilities, baseline rates, and spike and decay rates can be considered as random effects drawn from the prior distributions outlined in equation 6, where the common network hyperparameters $\chi = (\xi, \phi, \kappa, \tau, \iota, \beta)$ can also be described by prior distributions and updated within the MCMC estimation chain.

For population studies, it is most important to estimate the unknown H on the network. When H is large relative to M, or when $\mathbf{W}_t$ yields many short, random jumps among DHCP addresses, there may be little information available about the hour-to-hour dependence of the unobserved machine rates λ_i. For these settings, it may be more advantageous to use a simplified or marginal model as a function of the hyperparameters χ that averages across the state-to-state variation in λ_i from the underlying machines (see eg. Weaver, 2010).

When the number of unobserved machines H is small, it may be feasible to use a second data augmentation step that updates H using a complete conditional distribution, and disambiguates the counts z among the current number of machines. However, this exercise quickly grows computationally complex. For a network with M active IP addresses, each proxying a large number H_m of machines, estimation can be based on a marginal rate λ^*_{tm}, such that the counts z_{mt} are described by

$$\ell(\chi, H; z_0, \cdots, z_T) \quad = \quad \prod_{t=0}^{T} \prod_{m=1}^{M} \frac{\exp\left[H_m \lambda^*_{mt}\right] \left(H_m \lambda^*_{mt}\right)^{z_{tm}}}{z_{tm}!}$$

and, based on regularity conditions for large H_m, an approximate marginal transition model

$$p(\lambda^*_{mt} \mid \lambda^*_{mt-1}, \chi) \quad = \quad \text{Normal}\left(\mu_m(\chi, t), \sigma_m(\chi, t)\right)$$

This model assumes that networks that use large proxies have generally the same kinds of machines and activity as networks that do not (that is, the same set of hyperparameters is suitable for both types). A first step toward including network models is to parameterize three basic network administration policies: a static NAT, a DHCP pool with a 24-hour lease, and a load-balancing scenario that alternates among D NAT devices. These network models are also applicable to other situations beyond the Conficker-C botnet where counting takes place through the telescope of NAT and DHCP in IP space.

3.7 Model evaluation

Model mis-specification is most likely to occur with the format of the spike and decay functions, and the time-independence of the parameters ψ and θ. Introducing time dependence among ψ and θ is a case of nested modeling, whereas checking the form of the decay and spike functions is not.

Regression-inspired diagnostics, as outlined by Gelfand (1996), can be useful for examining the suitability of the spike-decay model. One example may be to use autocorrelation functions based on the posterior predictive distribution of λ_t to examine structure in the residuals resulting from a mis-specification of the shape of the decay and spike patterns. Posterior predictive residual plots or more formal Bayes factors can be used to assess the need for time-dependence in parameter values.

When both H and θ are ambiguous, there is little information available to distinguish between differing numbers of hosts and widely variable baseline rates. One way to assess model interpretation could be to measure the posterior distribution of the dispersion of baseline rates against the prior informed by the behavior outlined in Section 2.3 in order to check for a large discrepancy in prior vs. posterior spread.

4 Data collection and proposed work

From the period of March 5th through April 24th, 2009, Conficker-C UDP P2P connection attempts sent into a large private network (approximately $21,000$ /24 net blocks, or 0.15% of IP address space) were collected using the SiLK Conficker.C Plug-In (Gates et al., 2004; Faber, 2009). A total of 33.6 million unique IP addresses were observed performing Conficker-C P2P scans over the 2-month window. Hourly counts were collected, and aggregated by /24 net block to reduce the sparsity and size of the data set, as well as to account for ephemeral DHCP leases allocated over a small number of addresses. A total of 1.1 million unique /24 net blocks were observed performing Conficker-C P2P scans. Geolocation information for each block was obtained using a database of country codes associated with IP addreses. Each net block was also assigned roughly to a time zone based on its country code, with 1% of blocks remaining unassigned due to satellite locations or unavailable country codes.

To complete the dissertation, the following work is proposed:

1. **Incorporate births and deaths in the single-host model**: To track births and deaths in the population, the sets $\mathcal{O}$, $\mathcal{D}$ and $\mathcal{S}$ must be augmented to include two more states: $t \in *$, a prior infection state, and $t \in \dagger$, a post-cleanup "death" state. Incorporating states $t \in \{*, \dagger\}$ is a straightforward extension of the 3×3 transition matrix to include more possibilities for observed 0 counts prior to any observed activity or following all observed activity.

2. **Implement the expanded single-host model**: Using a suitably fast programming language (C or python, for example), implement the estimation scheme for the single-host model described in Section 3.1 through Section 3.5.

3. **Assess estimation method as applied to the collected data**: Use the single-host model implementation to estimate netblock-specific parameters for the two-month data set, relaxing the prior on the baseline rate q to account indirectly for the heterogeneity of machines behind net blocks. Assess model convergence, mixing rates, and the suitability of the geometric spike-decay rates.

4. **Formalize the network model**: Determine conditional distributions and appropriate marginalizations (see Section 3.6) for efficient estimation of H in the presence of multiple networks and confusion matrices. Provide parameterizations of three "typical" confusion matrices over time: a NAT, a large DHCP pool with average lease of 24 hours, and a load-balancing network that switches disjointly between a number D of NATs.

5. **Implement the network model**: Expand the code from step (2) to include network modeling.

6. **Obtain population estimates**: Use the informed baseline rate parameters and the network model to obtain population estimates from the two-month data set. Assess suitability of time-dependent network parameters.

References

Chan, M. and Hamdi, M. (2003). An active queue management scheme based on a capture-recapture model. *IEEE Journal on Selected Areas in Communications*, 21(4):572–583.

Claffy, K., Polyzos, G., and Braun, H.-W. (1989). Internet traffic flow profiling. Technical Report TR-CS93-328, University of California San Diego.

Dupuis, J. and Schwarz, C. (2007). A Bayesian approach to the multistate Jolly-Seber capture-recapture model. *Biometrics*, 63:1015–1022.

Faber, S. (2009). Silk Conficker.C Plug-in. `http://tools.netsa.cert.org/wiki/display/tt/SiLK+Conficker.C+Plugin`. CERT Code release.

Fienberg, S., Johnson, M., and Junker, B. (1999). Classical multilevel and bayesian approaches to population size estimation using multiple lists. *Journal of the Royal Statistical Society: Series A*, 162(3):383–405.

Gates, C., Collins, M., Duggan, M., Kompanek, A., and Thomas, M. (2004). More netflow tools for performance and security. In *LISA '04: Proceedings of the 18th USENIX Conference on System Administration*, pages 121–132, Berkeley, CA, USA. USENIX Association.

Gelfand, A. (1996). Model determination using sampling-based methods. In Gilks, W., Richardson, S., and Spiegelhalter, D., editors, *Markov Chain Monte Carlo in Practice*, chapter 9. Chapman and Hall/CRC.

Horowitz, K. and Malkhi, D. (2003). Estimating network size from local information. *Information Processing Letters*, 88:237–243.

Li, Z., Goyal, A., Chen, Y., and Paxson, V. (2009). Automating analysis of large-scale botnet probing events. In *ASAICCS '09*.

Mane, S., Mopuru, S., Mehra, K., and Srivastava, J. (2005). Network size estimation in a peer-to-peer network. Technical Report TR 05-030, University of Minnesota Department of Computer Science and Engineering.

Paxson, V. and Floyd, S. (1995). Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244.

Porras, P., Saidi, H., and Yegneswaran, V. (2009a). Conficker C Actived P2P scanner. `http://www.mtc.sri.com/Conficker/contrib/scanner.html`. SRI international Code release/document.

Porras, P., Saidi, H., and Yegneswaran, V. (2009b). Conficker C analysis. Technical report, SRI International.

Porras, P., Saidi, H., and Yegneswaran, V. (2009c). Conficker C P2P protocol and implementation. Technical report, SRI International.

Psaltoulis, D., Kostoulas, D., Gupta, I., Briman, K., and Demers, A. (2005). Decentralized schemes for size estimation in large and dynamic groups. Technical Report UIUCDCS-R-2005-2524, University of Illinois Department of Computer Science.

Schwarz, C. and Arnason, A. (1996). A general methodology for the analysis of capture-recapture experiments in open populations. *Biometrics*, 52(3):860–873.

Tanner, M. and Wong, W. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540.

Weaver, R. (2010). A probabilistic population study of the Conficker-C botnet. In *PAM 2010, LNCS 6032*, pages 181–190. Springer-Verlag Berlin Heidelberg.