Beyond the Blacklist: Modeling Malware Spread and the Effect of Interventions

Benjamin Edwards
University of New Mexico
Albuquerque, NM, USA
bedwards@cs.unm.edu

Tyler Moore
Southern Methodist University
Dallas, TX, USA
tmoore@lyle.smu.edu

George Stelle
University of New Mexico
Albuquerque, NM, USA
stelleg@cs.unm.edu

Steven Hofmeyr
Lawrence Berkley National Laboratory
Berkley, CA, USA
shofmeyr@lbl.gov

Stephanie Forrest
University of New Mexico & Santa Fe Institute
Albuquerque, NM, USA
forrest@cs.unm.edu

ABSTRACT

Malware spread among websites and between websites and clients is an increasing problem. Search engines play an important role in directing users to websites and are a natural control point for intervening using mechanisms such as blacklisting. The paper presents a simple Markov model of malware spread through large populations of websites and studies the effect of two interventions that might be deployed by a search provider: blacklisting infected web pages by removing them from search results entirely and a generalization of blacklisting, called depreferencing, in which a website's ranking is decreased by a fixed percentage each time period the site remains infected. We analyze and study the trade-offs between infection exposure and traffic loss due to false positives (the cost to a website that is incorrectly blacklisted) for different interventions. As expected, we find that interventions are most effective when websites are slow to remove infections. Surprisingly, we also find that low infection or recovery rates can increase traffic loss due to false positives. Our analysis also shows that heavy-tailed distributions of website popularity, as documented in many studies, leads to high sample variance of all measured outcomes. This result implies that it will be difficult to determine empirically whether certain website interventions are effective, and it suggests that theoretical models such as the one described in this paper have an important role to play in improving web security.

Categories and Subject Descriptors

C.2.0 [**Computer Communication Networks**]: General—*Security and Protection*; I.6.5 [**Simulation and Modeling**]: Model Development; I.6.6 [**Simulation and Modeling**]: Simulation Output Analysis

NSPW'12, September 18–21, 2012, Bertinoro, Italy.
Copyright 2012 ACM 978-1-4503-1794-8/12/09 ...$15.00.

General Terms

Security

Keywords

Graduated Response, Blacklist, Search, Web Security, Modeling, Malware, Drive-by-Downloads

1. INTRODUCTION

The network worms that caused havoc ten years ago, such as Code Red, actively spread by 'pushing' themselves onto vulnerable systems through automated scanning. In contrast, a major problem today is computer infections that propagate via a 'pull'-based mechanism. For example, in a drive-by download, an attacker infects a victim computer's web browser without direct interaction [26, 27]. In this scenario, the attacker first compromises an otherwise benign web server, injecting executable code into its web pages, and then waits for users to visit the infected website and acquire the infection. Because many users arrive at websites through search, search engines are a crucial battleground over the distribution of malware.

Search providers have an incentive to defend against such attacks because they degrade search results. A typical approach is that taken by Google, which attempts to detect and blacklist websites that host malicious content [10]. Blacklisting can take the form of displaying a warning message via a client side browser plugin to discourage users from visiting a website, or outright removal from the search results. Blacklisting can be used to combat many types of malicious content, which is important in a web environment where new attacks are developed frequently. However, because blacklisting can dramatically reduce visits to websites, search engines are careful to avoid false positives (i.e., flagging an uninfected website as infected). Such caution can delay responses, which in turn may raise infection rates.

In this paper we devise a concise Markov model to study how web infections spread through large populations of websites, and explore how infections might be contained with blacklisting. We also propose a generalization of blacklisting called *depreferencing*, where a search engine reduces a website's ranking in search results in proportion to the engine's certainty that the website is infected. Depreferencing can be more tolerant of false positives than a binary response such as blacklisting because the scale of the intervention can be adjusted to specific levels of false positives. Depreferencing provides a controllable *depreferencing parameter*, σ, that can be tuned to achieve specific reductions in infections or false positives. We derive exact analytic expressions that relate the depreferencing parameter, σ, to infection rates and traffic loss due to false positives. We also identify critical points for the model parameter values that govern the trade-off between infection and traffic loss.

We believe that modeling is particularly well-suited to the task of examining techniques for controlling malware spread over the web. First, it allows us to examine unconventional interventions, such as depreferencing, at low cost. Given the relatively grim status quo in web security,[1] more radical countermeasures deserve consideration, and modeling offers a good way to assess the impact of new strategies without the expense and commitment of an actual implementation.

Second, modeling can deal with the extreme dynamics of the web better than empirical exploration alone. Our analysis shows that the heavy-tailed distribution of website popularity leads to high variance in outcomes. It is often possible for online services such as search engines to perform experiments by rolling out improvements to subsets of clients. However, high sample dependence makes it extremely difficult to conduct reliable comparative assessments of the benefits of different interventions, especially with a limited number of empirical measurements. For example, the number of known malicious IP's can vary wildly over time, as shown in Figure 1. Extremely high values could conceivably be the result of a very popular website becoming infected. We show that this variance can obscure even large improvements in infection and recovery. With the modeling approach, we can easily run many simulations, and more reliably estimate the comparative impacts of different intervention strategies.

Finally, modeling lets us examine the impact of interventions across many stakeholders and identify tensions that may arise. For instance, improved security for search operators and consumers may be achieved in part at the expense of increased risk of incorrect blacklisting for website operators. Modeling allows us to more precisely quantify these trade-offs.

2. MODELING INFECTIONS

We model a population of servers that is under attack from malicious agents, as depicted in Figure 2. We do not model specific types of infections, assuming that an infection is any event that compromises a website such that it could be used to spread malware to users. Once infected, a server recovers when an administrator notices the infection and clears it. In this paper we explore the impact of search provider interventions and so are only interested in clients that connect to servers via referrals from a search provider. Hence, in our model, client exposure to infection is driven solely by website popularity as determined by the search provider. In an attempt to improve search results, the search provider monitors websites to determine whether they are infected and may incorrectly identify uninfected websites as infected. We assume that an administrator clears false identifications of infection at the same rate as real infections.

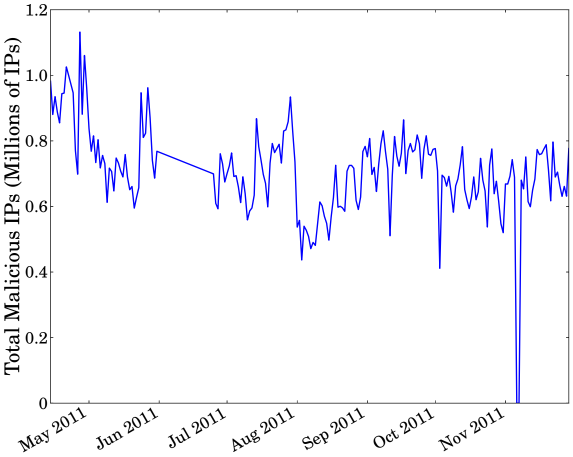


Figure 1: Variation in malicious IP addresses over time; from the Internet Storm Center (`http://isc.sans.org`)

Our model includes a population of n websites[2], each with a popularity, ω_i, drawn at random from a specified distribution. ω_i represents the total number of visits a website receives. The key outcome we are interested in measuring is *client exposure*, which is directly proportional to the expected number of visits that infected websites receive. At any time, a website is in one of three possible states: infected, uninfected, or falsely infected (i.e. classified by the search provider as infected when it is actually not). Each server transitions between these states at discrete time steps, according to the Markov chain depicted in Figure 3. The key parameters are: ρ, the probability of a website becoming infected; γ, the probability of recovering from an infection; and f, the probability of falsely being classified as infected.

We make the simplifying assumption that the probabilities ρ, γ, and f are constant across the population of servers and time invariant.[3] Unfortunately, data on the exact distributions of these parameters are not readily available and often contradictory. For example, there are no data supporting a systematic relationship between a website's popularity and its susceptibility to infection. Although Moore et al. [22] found that more popular web search terms are less likely to include infected websites in their results, it is possible that more popular sites are higher priority targets for exploitation and therefore more likely to be infected. Because in both cases the effects are likely small we argue that assuming constant probabilities is reasonable.

Our model is discrete time; an alternative approach is to model the population of servers using differential equations. In the case of large n, the steady state distribution of infection probability would be exactly the infection rates in a differential equation model [39]. We use a discrete-

[1] A 2011 report found that 84% of websites were vulnerable to attack for more than 30 days of the 2010 calendar year [35].

[2] We use the terms website and web server, or simply server, interchangeably.

[3] In Section 5.5 we briefly explore the impact of relaxing this assumption.

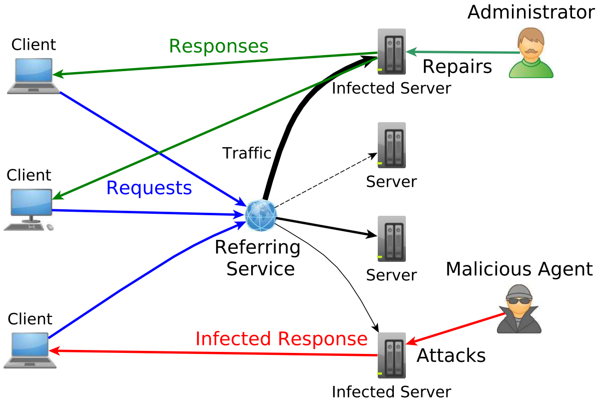


Figure 2: Server and client infections via search engine referral.

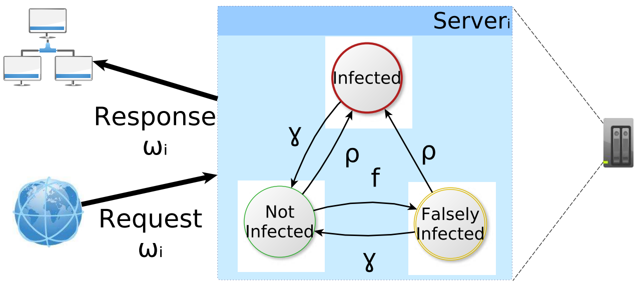


Figure 3: Model of website infections and client exposure.

time model instead because it allows us to easily incorporate time-dependent phenomena (such as interventions) and distributions of values (such as traffic), and it is simpler to explore transient effects.

3. MODELING INTERVENTIONS

We model two forms of intervention: *blacklisting*, which is currently used by search engines, and a hypothetical approach called *depreferencing*, which offers a way to adjust intervention parameters to specifically control the trade-off between infections and traffic loss due to false positives.

3.1 Blacklisting

A common approach taken by search engines that detect a compromised website is to inform the user in the search results or through a client side application before the user has a chance to visit the website, and then to produce further warnings if the user persists in attempting to visit the website. Although research has shown that users will often disregard warnings, such as certificate warnings [30, 31] and phishing attack warnings [5, 18], these studies do not investigate the typical multistep warning process presented in search results. Search result warnings are difficult to circumvent, and we believe that users are likely to simply choose an alternative result (provided they are not trying to reach a specific site). Additionally, certificate warnings are commonplace due to badly designed APIs of SSL implementations [7], which may lead to the high level of circumvention.

For example, if Google identifies a site as malicious, the text “This site may harm your computer”, will appear under the link in the search results. Attempting to proceed will take the user to a warning page with a small text URL that must be copied and pasted (without instruction) into the browser navigation bar to proceed. If the user persists, client side browser tools will present yet another warning page with a small link to “proceed anyway” (Chrome Browser) or “ignore this warning” (Firefox Browser). This is a far cry from a simple click through dialog.

Given that the goal of blacklisting is to prevent a website from receiving all or nearly all of its search traffic, minimizing false positives is essential. For example, Rajab et al. [28] claim that Google’s Safe Browsing infrastructure “generates negligible false positives”, and Google themselves state they “. . . strive for high quality and have had only a handful of false positives” [24]. While the specific amount of resources dedicated to correctly identifying malicious websites remains unknown outside of search providers, Google states that it “. . . invests heavily in the Safe Browsing team.” [24] The fact that any resources are currently dedicated to this problem indicates that Google believes malicious websites to be detrimental to a user experience.

We assume that blacklisting takes a fixed number of time steps to detect a compromised website and blacklist it. We refer to this as the *detection delay*, denoted β. A website infected at time t will be blacklisted at time $t + \beta$. Once blacklisted, the traffic to that the website is set to zero, i.e. $\omega_i = 0$. Formally, if a website, i, is infected at time x, its traffic, $\hat{\omega}_i$, at time $t > x$ is

$$\hat{\omega}_i = \begin{cases} \omega_i, & \text{if } t - x < \beta \\ 0, & \text{if } t - x \geq \beta \end{cases} \tag{1}$$

The time period β captures the notion that it will take a search engine a certain amount of time to determine that a website is compromised with high certainty (negligible false positives). Thus β accounts for how frequently the website is crawled, how much computational time is required to confirm the infection, how much the search engine is willing to invest in malware detection, and other possibilities, such as giving the compromised site a certain grace period to clean up the infection. We do not have good information on the specific costs associated with β. However, the specifics are irrelevant to the outcomes of our model, and search engines can use their own data to evaluate each action that contributes to β.

In the model, we assume that immediately after a website recovers, its popularity is restored to its previous value. That is, once a website has been cleaned, the administrator informs the search engine and the blacklisting is removed without delay. In reality, there would be a small delay before the blacklisting is removed. For example, when an administrator requests Google to run an automated test for malware, it will take at most a few hours to complete, and up to 24 hours for the malware warning to disappear from all search results [25]. Because the time period is small and constant, we can exclude it from our model without significantly changing the results.

3.2 Depreferencing

We explore a generalized hypothetical intervention, called *depreferencing*, which, to the best of our knowledge, is not actually implemented by any existing search engine. The idea is that when a search engine detects a possibility of infection in a website, it reduces the traffic that website receives. This could be implemented by reducing the rank of that website in the search results, or probabilistically providing warnings to users. Because the response does not block all traffic to the website, but rather reduces the volume of traffic, the detection process can tolerate false positives, allowing the search engine to react more rapidly and aggressively. Search providers could use potentially coarser and less precise detectors to crawl websites more frequently, requiring significantly less computation time to classify websites as infected.

We model this intervention by reducing the popularity of a website by a fixed percentage every time step after it is discovered that the website is infected. If a website is infected at time x, an infected website's traffic at time $t > x$ is

$$\hat{\omega}_i = \begin{cases} \omega_i, & \text{if } t - x < \beta \\ \sigma^{(t-x)-\beta+1}\omega_i, & \text{if } t - x \geq \beta \end{cases} \tag{2}$$

where $0 \leq \sigma \leq 1$ is the *depreferencing* /parameter, which controls the strength of the response. Note that Equation 1 is equivalent to Equation 2 when $\sigma = 0$. We believe that adjusting search results is a plausible response that would be easy to implement. For example, a search engine like Google could simply reduce the page ranks of infected websites, which would directly affect their popularity in search results. Similarly to blacklisting, we assume that when a website recovers from an infection, its popularity is immediately restored to its original value. Because depreferencing is a less drastic response, search engines might be able to reduce the detection delay β if they were to adopt this intervention.

Equation 2 is one of an even more general class of methods for combating exposure to infection. We could define a general $g(\omega_i, x)$, such that g is monotonically decreasing in time. For example g could be a linear or logistic function. We choose an exponential decline as it seems a natural fit for our application. Investigation into other forms appropriate for other applications is left for future work.

As a consequence of the potentially more rapid, and hence, imprecise detection of compromised sites, our model includes a constant probability f that an *uninfected*/website is classified as compromised and has its rank reduced. This is in contrast to the blacklisting approach, where we assume there are zero false positives. For depreferencing, we assume that websites that are incorrectly classified as compromised recover at the same rate, γ, as compromised websites. In other words, the process of recovery is the same whether a website is actually infected or not. This requires that the administrator realize that the website is infected (for example, users of Google's Webmaster Tools are notified when their sites are infected) and that appropriate steps are taken to correct the problem.

We do not model false negatives, i.e. infected websites that go undetected, because our model studies the effect of interventions on client infection rates, and we assume that in both blacklisting and depreferencing the detection process has similar levels of false negatives. Hence, the false negative rates should not affect comparison of the outcomes. From a practical perspective, data on false negatives are rare or non-existent because they are extremely difficult to gather. We leave the analysis of false negatives to future work

4. ANALYSIS

This section analyzes the mathematical properties of the model described in the previous section. First we describe the steady state values of the Markov chain shown in Figure 3. Second, we analyze the first and second moments of the random variables that define the traffic loss and the number of clients exposed to infection. We then provide expressions that relate the intervention parameters to the infection exposure and traffic loss, and identify critical control points.

4.1 Steady State Distribution

Let the state of a server i in the Markov chain in Figure 3 be the random variable $S_i \in \{I, N, F\}$, where I denotes infection, N denotes no infection and F denotes a false positive infection. It is easy to see that the Markov chain is ergodic except for some degenerate cases such as $f = 1, \gamma = 1, \rho = 0$. However, such cases are unlikely to occur in the real world.

Because our Markov chain is ergodic it is guaranteed to converge to a unique stationary distribution, which is given by

$$Pr[S_i = I] = \frac{\rho}{\rho + \gamma} \tag{3}$$

$$Pr[S_i = N] = \frac{\gamma}{(f + \rho + \gamma)} \tag{4}$$

$$Pr[S_i = F] = \frac{f\gamma}{(\gamma + \rho)(f + \gamma + \rho)} \tag{5}$$

Moreover, because this is a finite time-homogeneous ergodic Markov chain, it will have a short mixing time. Hence we focus on the steady-state in the remainder of the analysis.

4.2 Client Exposure and Website Loss

The probability that a website becomes infected at a time $t - x$ and remains infected until time t depends on the probability that the website was not infected at time $t - (x + 1)$, became infected at time $t - x$, and remained infected for the next x timesteps. More formally, let I_x denote the event that a server i has been in a state of infection for *exactly* x time steps. Then

$$Pr[S_i = I_x] = \rho(1 - Pr[S_i = I])(1 - \gamma)^x \tag{6}$$

Observe that the events $S_i = I_x$ and $S_i = I_{x'}$, with $x \neq x'$, are mutually exclusive, e.g. a server cannot be infected for exactly 5 and exactly 6 time steps.

Next we derive an expression for the random variable $X_i(\beta, \sigma)$, which describes the number of clients exposed to infection from a website i, when the search provider implements an intervention controlled by the parameters β and σ. Recall that β is the detection delay for infection identification and σ is the depreferencing parameter, i.e., the strength of the response. The expectation of exposure to infection from website i is then

$$\begin{aligned}\mathbb{E}[X_i(\beta,\sigma)] =& \sum_{x=0}^{\beta-1} \omega_i \frac{\rho\gamma(1-\gamma)^x}{\rho+\gamma} + \\ & \sum_{x=\beta}^{\infty} \omega_i \sigma^{x-\beta+1} \frac{\rho\gamma(1-\gamma)^x}{\rho+\gamma} \\ =& \frac{\omega_i\rho\gamma}{\rho+\gamma}\left[\frac{1-(1-\gamma)^\beta}{\gamma} + \frac{\sigma(1-\gamma)^\beta}{1-(\sigma(1-\gamma))}\right] \quad (7)\end{aligned}$$

The above expression simplifies to $\omega_i Pr[S_i = I_i]$ when no intervention is taken, which would correspond to $\beta = \infty$ or $\sigma = 1$.[4]

The other important random variable we are interested in is $L_i(\beta,\sigma)$, which represents the traffic lost by a website i as a consequence of false positives. Following a similar analysis to the earlier one for client exposure, if F_x denotes being in the false positive state for x time steps, we have

$$Pr[S_i = F_x] = fPr[S_i = U](1-(\gamma+\rho))^x \qquad (8)$$

The lost traffic at a specific time will be $\omega_i - \hat{\omega}_i$. Substituting for $\hat{\omega}_i$ as given by Equation 2, the expected traffic loss is

$$\begin{aligned}&\mathbb{E}[L_i(\beta,\sigma)] = \\ &\frac{\omega_i f\gamma(1-(\rho+\gamma))^\beta}{f+\gamma+\rho}\left[\frac{1}{\gamma+\rho} - \frac{\sigma}{1-\sigma(1-(\gamma+\rho))}\right] \quad (9)\end{aligned}$$

We can then define the *infection exposure*, which is the fraction of traffic exposed to infection from all websites, as

$$X(\beta,\sigma) = \frac{\sum_i^n X_i}{\sum_i^n \omega_i} \qquad (10)$$

and the overall *traffic loss* due to false positives as

$$L(\beta,\sigma) = \frac{\sum_i^n L_i}{\sum_i^n \omega_i} \qquad (11)$$

Using linearity of expectation, the expressions for $\mathbb{E}[X(\beta,\sigma)]$ and $\mathbb{E}[L(\beta,\sigma)]$ are simply those in Equation 6 and Equation 8 respectively, while omitting ω_i, specifically:

$$\begin{aligned}&\mathbb{E}[X(\beta,\sigma)] = \\ &\frac{\rho\gamma}{\rho+\gamma}\left[\frac{1-(1-\gamma)^\beta}{\gamma} + \frac{\sigma(1-\gamma)^\beta}{1-(\sigma(1-\gamma))}\right] \quad (12)\end{aligned}$$

$$\begin{aligned}&\mathbb{E}[L(\beta,\sigma)] = \\ &\frac{f\gamma(1-(\rho+\gamma))^\beta}{f+\gamma+\rho}\left[\frac{1}{\gamma+\rho} - \frac{\sigma}{1-\sigma(1-(\gamma+\rho))}\right] \quad (13)\end{aligned}$$

We note that both of the infection exposure and the traffic loss are independent of the distribution from which the ω_i's are drawn, or how many servers there are.

The effectiveness of the depreferencing parameter, σ, and the detection delay, β, in the control strategy for $\mathbb{E}[X(\beta,\sigma)]$, depends only on the recovery rate γ. If $\gamma \approx 1$ (a fast recovery rate), then $\mathbb{E}[X(\beta,\sigma)] \approx Pr[S_i = I]$, i.e. the expected infection exposure is approximately the probability of a single server being in the infected state. Only when websites are slow to react to infections are interventions which alter traffic likely to have significant impact.

Conversely, ρ and γ both affect $\mathbb{E}[L(\beta,\sigma)]$. In particular, a decrease in a particular websites infection rate ρ or the recovery rate γ will cause an increase in loss due to false positives for a fixed false positive rate f. Intuitively, a website that is unlikely to be in the infected state is more vulnerable to being *falsely infected*.

We now determine the variance in $X(\beta,\sigma)$ and $L(\beta,\sigma)$. Because each of the X_i's is independent and the sum of the traffic is a constant,

$$Var[\frac{\sum_i^n X_i}{\sum_i^n \omega_i}] = \frac{\sum_i^n Var[X_i]}{\left(\sum_i^n \omega_i\right)^2} \qquad (14)$$

Additionally, variance can be defined as $Var[X_i(\beta,\sigma)] = \mathbb{E}[X_i(\beta,\sigma)^2] - \mathbb{E}[X_i(\beta,\sigma)]^2$. Using these two facts and some simple algebra we have:

$$Var[X(\beta,\sigma)] = \left(\mathbb{E}[X(\beta,\sigma^2)] - \mathbb{E}[X(\beta,\sigma)]^2\right)\frac{\sum_{i=1}^n \omega_i^2}{(\sum_{i=1}^n \omega_i)^2} \qquad (15)$$

If the ω_i's are drawn from a distribution with finite variance and expectation and n is large, then we can apply the central limit theorem to Equation 15 to rewrite it in terms of the distribution of ω_i's

$$\begin{aligned}&Var[X(\beta,\sigma)] = \\ &\left(\mathbb{E}[X(\beta,\sigma^2)] - \mathbb{E}[X(\beta,\sigma)^2]\right)\left(\frac{Var[\omega_i] + \mathbb{E}[\omega_i]^2}{n\mathbb{E}[\omega_i]^2}\right). \quad (16)\end{aligned}$$

Observe that Equation 16 is monotonically decreasing in the number of servers n. So as the population of websites increases we expect the variance in the fraction of traffic exposed to infection to go to 0.

It is almost certain, however, that the distribution of ω_i for real webservers is heavy-tailed and does not have finite variance or finite expectation [1, 2, 21]. In the case of a heavy-tailed or power-law distribution of ω_i, the variance $Var[X]$ does not converge to a single value for large n, but to a distribution of values. Furthermore, because the sum of power-law i.i.d. random variables exhibits heavy tailed behavior [33] [9], the distribution of $Var[X(\beta,\sigma)]$ will also exhibit heavy tailed behavior.

The sum of power law distributed variables can be approximated by the maximum over the variables [38], which means that the last fraction in Equation 15 can be approximated as 1 for particularly heavy tailed distributions and large n, i.e.

$$\frac{\sum_{i=1}^n \omega_i^2}{(\sum_{i=1}^n \omega_i)^2} \to 1 \qquad (17)$$

If we take this as an upper bound, we see that improving either σ or β to lower infection will also lower the variance in the infection exposure rate. Depending on the value of the exponent in the distribution of traffic, $Var[X(\beta,\sigma)]$ may not have finite variance or expectation. As we discuss later, this is important because it implies that empirical studies of infection exposure (or traffic loss) are likely to be highly sample dependent, and that even significant changes to the variables like ρ and γ can be hard to discern.

[4] Because this expression has two parts, namely infection spread pre and post intervention, we could easily include two recovery rates γ_{pre} and γ_{post} to model the fact that recovery likely occurs more quickly after an intervention is taken. This does not significantly effect our results here or in experiments.

A similar analysis yields slightly different results for traffic loss:

$$Var[L(\beta,\sigma)] =$$

$$\left(2\mathbb{E}[L(\beta,\sigma)] - \mathbb{E}[L(\beta,\sigma^2)] - \mathbb{E}[L(\beta,\sigma)]^2\right) \frac{\sum_{i=1}^{n} \omega_i^2}{\left(\sum_{i=1}^{n} \omega_i\right)} \tag{18}$$

4.3 Critical Values

In general, changing parameter values from one set, (β, σ), to another, (β', σ'), will result in a change in infection exposure, i.e., $\mathbb{E}[X(\beta,\sigma)] \neq \mathbb{E}[X(\beta',\sigma')]$. However, there could be some settings of β' and σ', such that the outcome will not change, i.e., $\mathbb{E}[X(\beta,\sigma)] = \mathbb{E}[X(\beta',\sigma')]$. We call these settings, or transition points, the *critical* values for the parameters.

The critical value, σ_X, for the depreferencing parameter is the most important, because we expect that search providers will have more control over σ than β. For example, a new detection algorithm may require a different β'; the search provider could then use the critical value of σ_X to ensure that the infection exposure did not change.

To derive the critical value for the infection exposure, we first calculate an expression for the precise value of σ needed to achieve a particular infection exposure rate $\mathbb{E}[X(\beta,\sigma)] = \xi$, as

$$\sigma = \frac{\frac{(\rho+\gamma)\xi}{\rho\gamma} - \frac{1-(1-\gamma)^{\beta}}{\gamma}}{(1-\gamma)\left[\frac{(\rho+\gamma)\xi}{\rho\gamma} - \frac{1-(1-\gamma)^{\beta}}{\gamma}\right] + (1-\gamma)^{\beta}} \tag{19}$$

We can then derive the critical value for the infection exposure by substituting $\mathbb{E}[X(\beta',\sigma')]$ for ξ in Equation 19, which gives

$$\sigma_X = \frac{a}{\gamma + a(1-\gamma)} \tag{20}$$

where a is defined as

$$a = 1 - (1-\gamma)^{\beta-\beta'} + \frac{\sigma\gamma(1-\gamma)^{\beta-\beta'}}{1-\sigma(1-\gamma)} \tag{21}$$

Equation 21 shows the critical value needed to ensure the infection exposure does not change when β changes. An alternative goal might be to ensure that the traffic loss due to false positives does not change with a new value for β, i.e. $\mathbb{E}[L(\beta',\sigma')] = \mathbb{E}[L(\beta,\sigma)]$. This will be given by another critical value, σ_L. Once again, we first derive an expression for the precise value of σ needed to attain a particular expected traffic loss fraction $\mathbb{E}[L(\beta,\sigma)] = \lambda$,

$$\sigma = \frac{\frac{1}{\rho+\gamma} - \frac{\lambda(f+\gamma+\rho)}{f\gamma(1-(\rho+\gamma))^{\beta}}}{1 + (1-(\rho+\gamma))\left[\frac{1}{\rho+\gamma} - \frac{\lambda(f+\gamma+\rho)}{f\gamma(1-(\rho+\gamma))^{\beta}}\right]} \tag{22}$$

Setting $\mathbb{E}[L(\beta',\sigma')] = \lambda$ in Equation 22, we get

$$\sigma_L = \frac{b}{1 + b(1-\gamma-\rho)} \tag{23}$$

where b is defined as

$$b = \frac{1}{\gamma+\rho} - (1-\gamma-\rho)^{\beta-\beta'}\left[\frac{1}{\gamma+\rho} - \frac{\sigma}{1-\sigma(1-\gamma-\rho)}\right] \tag{24}$$

As can be seen from Equation 23, the critical value for the traffic loss is independent of the false positive rate f.

Using Equation 20 and Equation 23 in combination, a search provider has the ability to decide how to adjust σ to balance an increase in the traffic loss against an increase in infection exposure.

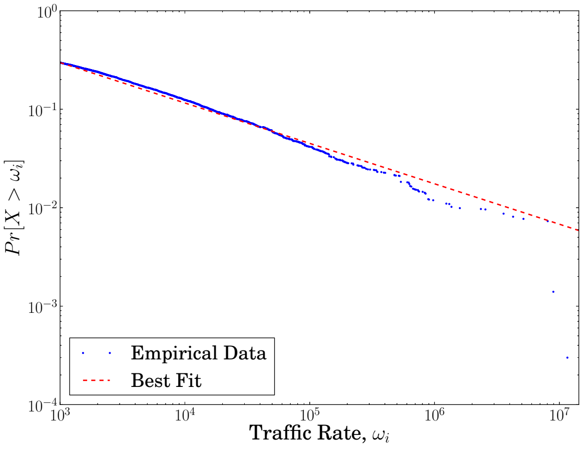


Figure 4: Empirically observed website traffic follows a power-law distribution with $\alpha = -1.4$.

5. EXPERIMENTAL RESULTS

To verify the results derived in Section 4 we used a Monte Carlo simulation of the model described in Section 2. Unless otherwise noted, we used the following parameter settings for all experiments: $\rho = 0.01$, $\gamma = 0.1$, and $n = 1000$. Although we believe that these parameter settings are plausible, our goal is not to provide a precise match with real-world outcomes, but rather to investigate more general consequences of features such as variance and the comparative efficacy of interventions. For each experiment, we conducted 1000 runs, and each run was 75 time steps. This length is sufficient for the model to reach a steady state.

We examine two different distributions throughout the experiments: uniform, with $\omega_i \propto Uniform(0,1)$, and power law with $\omega_i \propto x^{\alpha}$ with $\alpha = -1.4$. Although these two distributions are likely not precisely representative of the real world, they are useful in that they represent two possible extremes of variance (finite and undefined).

In reality, the distribution is likely heavy-tailed, possibly a power-law [1,2,21]. We found that a power-law with an exponent of $\alpha = -1.4$ provides a good fit with empirical data on website popularity, as can be seen in Figure 4. We calculated the exponent for a random sample of 10,000 websites listed in the top 1 million websites according to the web-analytics firm Alexa, using estimates for the daily number of visits obtained by querying the Alexa Web Information Services API.[5]

5.1 Popularity Distribution

According to the analysis in Section 4, distributions of website popularity with undefined variance will result in large fluctuations in client exposure to infection and will be highly dependent on the sample of servers chosen. This is confirmed in our experiments, as can be seen in Figure 5. The uniform distribution of website popularity results in low variance in client exposure (Figure 5a), whereas the

[5] `http://aws.amazon.com/awis/`

power law website popularity results in very high variance, both in a single run of the model and among different runs (Figure 5b).[6] For both popularity distributions, the experimental average of the runs rapidly converges to the expected steady-state value for X (0.091), although power-law distributions can yield X values as high as 0.96 in individual runs, an order of magnitude higher than the expected value.

Figure 6 shows the variation in individual runs more clearly. Figure 6a shows three different runs of the simulation with the same parameters, $\rho = 0.01, \gamma = 0.1$. There are large jumps in client exposure to infection that occur when the more popular websites get infected, followed by plateaus before those websites recover, and then abrupt drops after recovery. Figure 6b shows two runs of the model with different infection and recovery rate parameters. Strikingly, the run with the infection rate cut in half and the recovery rate doubled, seems to exhibit worse infection behavior. This clearly illustrates why it might be difficult to determine whether web security improvements are effective. The high variance in the runs illustrates the importance of modeling, as running experiments in the real world could require many trials over long periods of time to reach conclusions with any confidence.

We also tested distributions other than uniform and power-law and confirmed the theoretical prediction that distributions with finite variance produce low variance in the measured outcome, whereas those with undefined variance produce high variance in the measured outcome (results not shown).

5.2 Interventions

Figure 7 demonstrates the effect of varying the detection delay, β, on the steady state client exposure rate. For both uniform and power-law popularity distributions, blacklisting is effective only if implemented quickly, i.e. before websites have had sufficient time to recover. The likelihood of remaining infected for t time steps is $(1 - \gamma)^t$, which becomes exponentially small for large t. For example, once $\beta > 40$, the steady state expected exposure is very close to the theoretical value with no interventions (around 0.091). Thus, for larger β, most infections will resolve before infected websites are blacklisted. The precise relationship between γ and β is given by Equation 7.

The results of varying the depreferencing parameter, σ, are shown in Figure 8. Because proportional depreferencing of popularity has an exponential impact on the ranking (Equation 2), even large values of σ can reduce infection rates significantly, for example, when $\sigma = 0.9$, the steady state client infection rate is half of the baseline value.

Depreferencing gives finer control to search engines, because adjusting σ should be relatively easy, unlike trying to reduce β, the control parameter for blacklisting. This finer control might allow for algorithms that produce more false positives (which in turn would reduce the number of missed infections), because the effects of being mislabeled as infected could have far less impact on a website that was moved down in the search rankings rather than being blacklisted.

5.3 False Positives

Depreferencing makes it feasible to use imprecise detection algorithms that trade faster detection for higher false positives. In our model, this would translate into a higher value for f, the false positive probability. Figure 9 explores the impact of f on the change in traffic loss due to false positives. Once again, a large variance in the website popularity distribution has a large impact on the outcome, i.e. the traffic loss. Further, as can be seen in Figure 9, reducing the false positive rate is only worthwhile if it can be dropped below a certain value (in this particular example, around 0.2); when f is high enough, every website is mainly in the infected or falsely infected state, and rarely in the uninfected state.

5.4 Exploring the Parameter Space

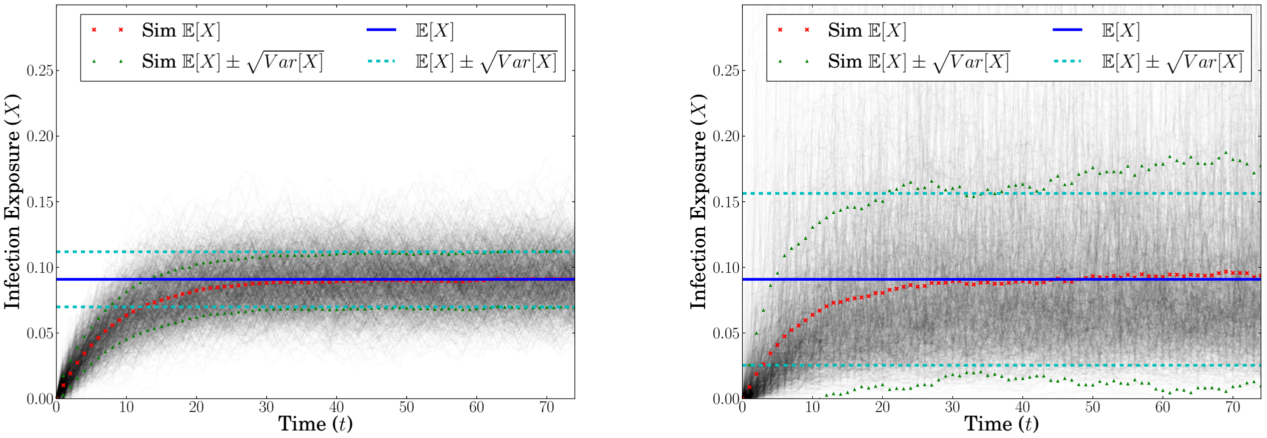
Figure 10 shows how the expected infection exposure and traffic loss change as the parameters σ and β vary from a base setting of $\beta = 10$ and $\sigma = 0.5$. We can see from the solid line at the critical value in Figure 10a that changing the depreferencing parameter, σ, can only correct for a small increase in β, up to $\beta = 11$. Beyond that, the expected exposure increases, regardless of the setting of σ. The value of σ only starts to have a large positive impact if the detection delay, β, drops significantly. We see similar results for the change in expected traffic loss, as shown in Figure 10b. Once again, only the smallest increases in β can be compensated for by increasing σ. However, lack of compensation means a decrease in traffic loss, which is a desirable outcome. We also see that it is easy to adjust σ to ensure that the traffic loss does not increase for almost every change in β.

It is clear that a faster response (reducing β) will reduce the infection exposure rate, and any potential traffic loss can easily be compensated for by changing σ. However, a faster response may be less accurate and result in a higher false positive rate, f. We explore this idea by again calculating the infection exposure with base values $\beta = 10$ and $\sigma = 0.5$, and then calculating the critical value σ_X needed to maintain the same infection exposure rate for a variety of β' values. We then measure the change in traffic loss $\mathbb{E}[L(10, 0.5)] - \mathbb{E}[L(\beta', \sigma_X)]$ for a variety of false positive rates. The results can be seen in Figure 11. Generally, a decrease in detection delay, β, increases the traffic loss for a constant false positive rate. If the false positive rate also goes up as β decreases, the problem is even worse. However, if the false positive rate can be kept sufficiently small (below 0.1 in this example), then there is flexibility to decrease the delay without a major increase in traffic loss.

5.5 Parameter Distributions

We also explored the effect of drawing the parameters ρ, γ, and f from distributions, instead of using constant values, but did not find an easily obtained analytic form for the distribution or moments of X or L. Preliminary simulations, however, suggest that the distribution of these values follows the joint distribution of Equation 7 and Equation 9 (results not shown). Further, our eariler result that there is large variance in the measured outcomes is observed when ρ and γ are related to ω_i. Further investigation into interactions among these paramters is left for future work.

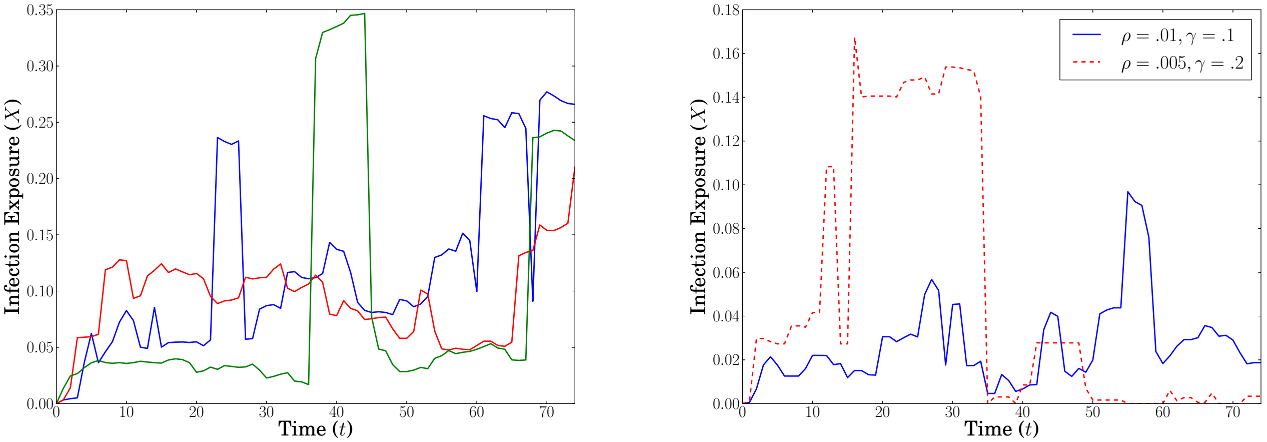
[6] Because the variance is undefined in general for a power-law, we substitute the run sample values of the ω_i's into Equation 15 to compute the theoretical variance shown in Figure 5b.



(a) Uniform website popularity

(b) Power-law website popularity

Figure 5: Variation in client exposure to infection over time. Individual runs are light gray. *Sim* X indicates the results of the simulation. Here $n = 250$ to illustrate the effects of small sample sizes.



(a) Different runs with same parameters, $\rho = 0.01$, $\gamma = 0.1$.

(b) Different runs with different parameters.

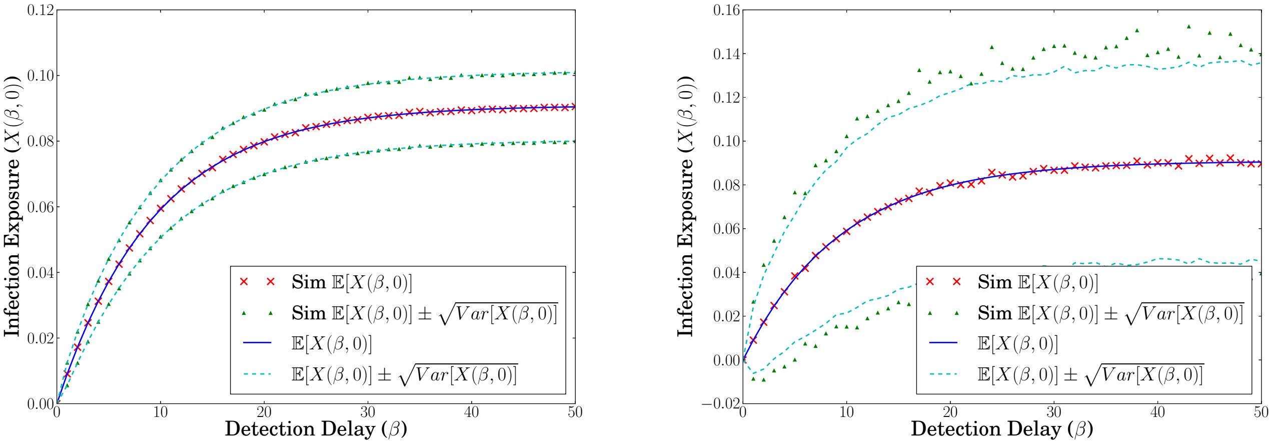
Figure 6: Variation of infection exposure in individual runs for power-law distribution of website popularity.

6. RELATED WORK

There are many approaches to combating web-based malware, including the use of virtual machines or kernel extensions to check for suspicious changes to the operating system [20, 23, 26, 34], emulating browsers to detect malicious JavaScript [3, 4], and detecting campaigns that promote compromised sites to the top of search results [14]. No technique is completely effective at disrupting web-based malware, according to a study of Google's data over more than four years [28]. In our view, one limiting factor is the choice of conservative approaches that minimize false positives at the expense of speedy detection. For example, Provos et al. [26] choose to minimize false positives in a system that allows explicit trade-offs between false and true positives.

Depreferencing of search results is an example of a graduated response, which is different from the binary, all-or-nothing, response methods, such as blacklisting, that are usually taken in cybersecurity. An early implementation of graduated response was a Linux kernel extension called pH [29], which responded to anomalous system call patterns by delaying subsequent system calls in the offending process. Other graduated responses operate by slowing down, or throttling, outgoing requests [13, 36] in active networks [11], Domain Name Service [37], Border Gateway Protocol [15], and peer-to-peer networks [8]. However, this is the first work we are aware of that uses a graduated response outside of the time domain.

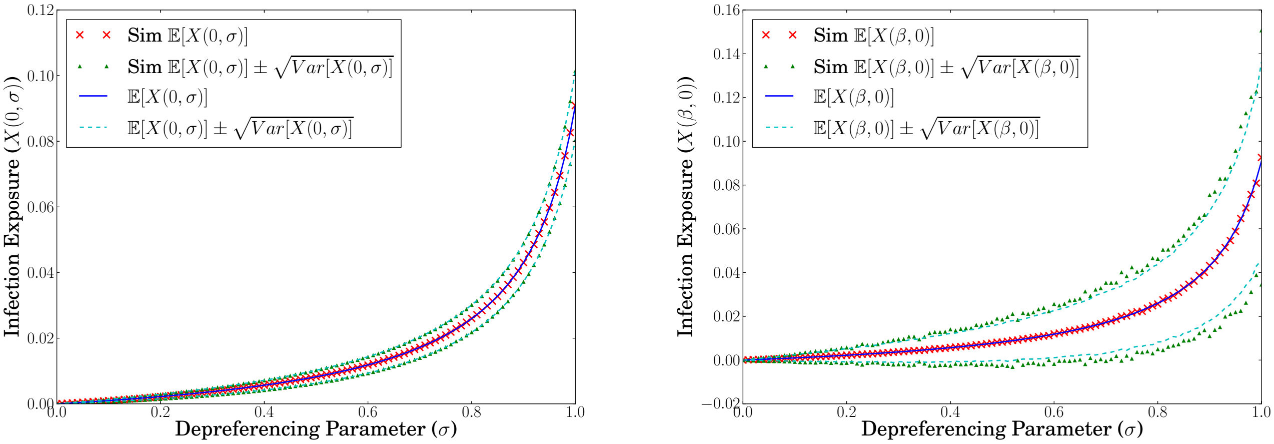
Several studies have focused on alternative intervention strategies, which could potentially be generalized using our depreferencing method. For example, Hofmeyr et al. mod-



(a) Uniform website popularity

(b) Power-law website popularity

Figure 7: Steady state client exposure to infection for various detection delays, β, with $\sigma = 0$.



(a) Uniform website popularity

(b) Power-law website popularity

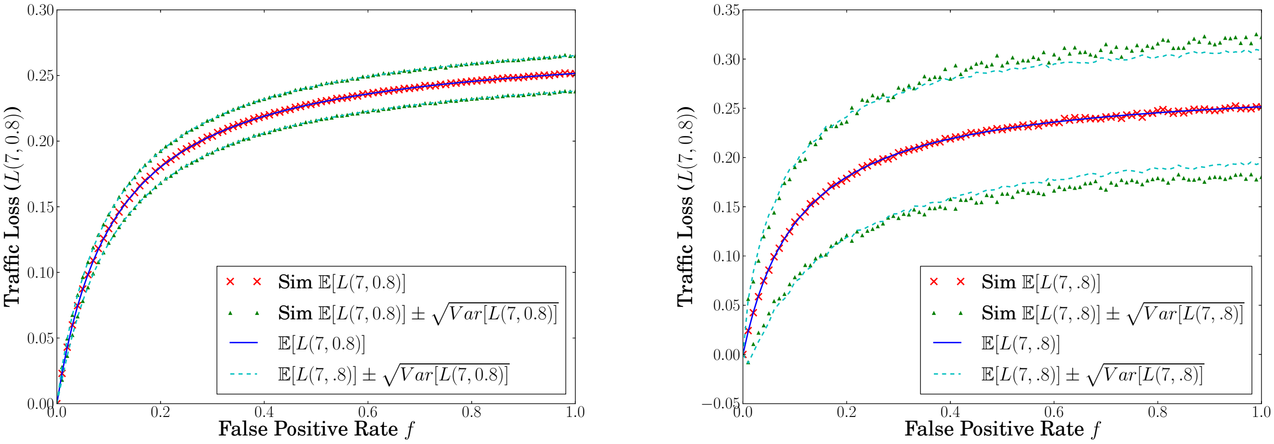
Figure 8: Steady state client exposure to infection for various depreferencing adjustment values, σ, with $\beta = 0$.

eled responses available to ISPs [12]. Other researchers have identified suitable intervention strategies based on empirical research, which might also be amenable to depreferencing. For example, Levchenko et al. [17] found that criminals relied on just three payment processors to collect money from victims, which led the authors to recommend targeting the payment processors as a low-cost intervention. Similarly, Liu et al. [19] empirically measured the effectiveness of pressuring registrars to suspend spam-advertising domain names. In a related intervention, Google has successfully pushed ad-filled sites down the results by changes to its search-ranking algorithm [22], suggesting that a similar effort to depreference malware-infected sites is technically feasible.

7. DISCUSSION

A general theme of this research is the emphasis on modeling. Modeling is a cost-effective way to explore intervention strategies, including investigating novel ideas, without the expense of first implementing them. As our results show, modeling can be particularly helpful for understanding long-term trends in processes with high variance, where direct experimentation can be misleading. Thoroughly testing the interventions we explore in this paper would likely require an unreasonable amount of time and money for any search provider.

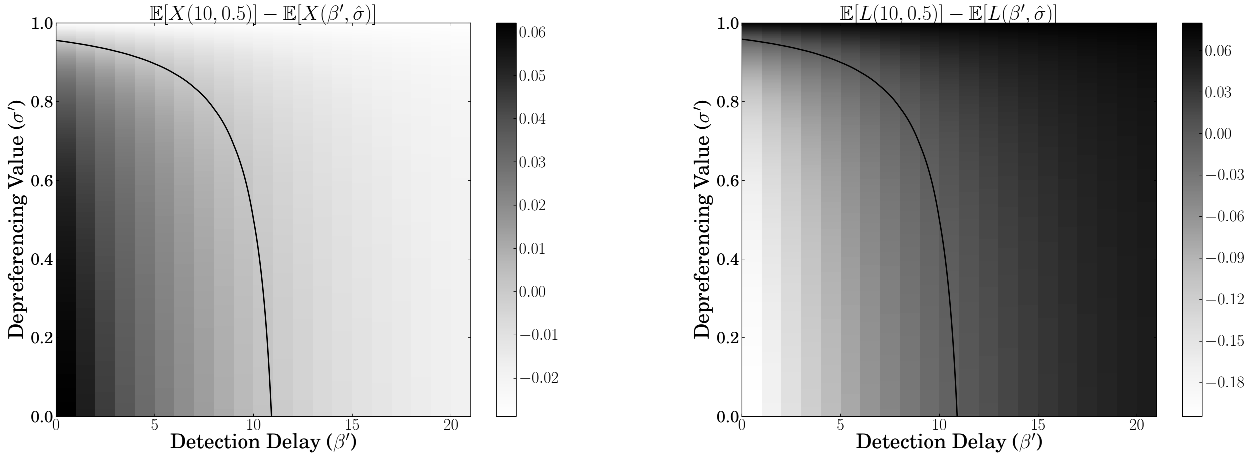
To the best of our knowledge, the idea of website depreferencing to prevent the spread of malware has not appeared in the literature before. Although we believe that depreferenc-



(a) Uniform website popularity

(b) Power-law website popularity

Figure 9: Steady state normalized traffic loss for various false positive rates. Each data point is the average of 1000 runs. The value used for the depreferencing parameter was $\sigma = 0.8$.



(a) Change in expected infection exposure

(b) Change in expected traffic loss

Figure 10: Changes in outcomes when parameters β' and σ' vary from a base of $\beta = 10$ and $\sigma = 0.5$. The solid lines correspond to the critical values, $\sigma' = \sigma_X$ in a) and $\sigma' = \sigma_L$ in b).

ing is technically feasible[7], other issues may arise with this type of response. For example, a policy that explicitly tolerates false positives could trigger accusations of bias against search engines.[8] In the future, government regulations may require search providers to enact measures which both avoid bias and protect users from malicious software.

Another issue is how depreferencing might be gamed. For example, there could be an incentive to deliberately infect competitors' websites, or cause them to appear infected, so their search rankings are demoted. Such industrial sabotage may in fact already happen. However, the scope for it could increase if less precise, false-positive tolerant detection mechanisms are used.

Depreferencing may have other advantages over traditional blacklisting. As web content and attacks become increasingly sophisticated, it may become more difficult to distinguish infected from non-infected states. For example, websites may host malicious advertisements in a frame, in which case only part of the website would be infected. Additionally, sophisticated attacks may hide themselves periodically. In these cases, infection is not a binary state but could be

[7] For example, the Google Penguin update uses a form of depreferencing to decrease rankings for websites that violate Google's quality guidelines (http://insidesearch.blogspot.co.uk/2012/04/another-step-to-reward-high-quality.html).

[8] The European Union is already investigating accusations that Google abused its power by preferring its own results over rivals. See `http://www.time.com/time/business/article/0,8599,2034138,00.html`.

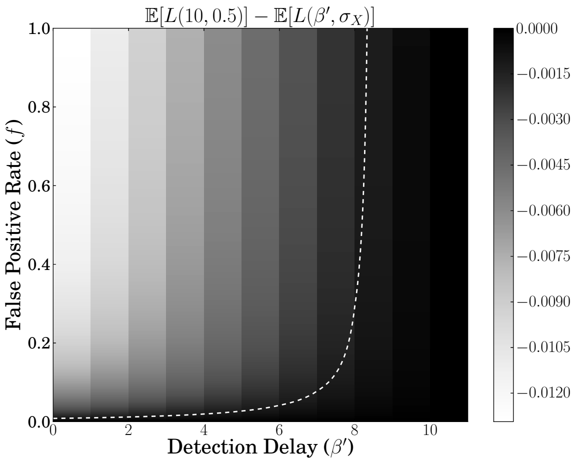


Figure 11: Change in expected traffic loss when expected infection exposure is kept constant, i.e. $\sigma' = \sigma_X$, the critical value. The base for comparison is $\beta = 10$ and $\sigma = 0.5$. The dotted line corresponds to a value of -0.1, i.e. an increase in traffic loss of 10%.

measured by degree. The degree of infection could then be incorporated into the depreferencing parameter allowing for a proportionate response. Finally, if the computational cost of detecting these more sophisticated attacks becomes a bottleneck, faster, less precise methods, which favor depreferencing, will become necessary.

We have made several simplifying assumptions that we believe are reasonable in the absence of more detailed information. For example, we assume that website infection and client infection probabilities are independent. In reality, this may not be the case. One variety of drive-by-download malware steals the login credentials of users who administer websites, enabling the malware to spread to those websites. Hence, when a client is infected, the probability of infecting one or more websites increases, corresponding to a change in ρ. We have chosen not to model this form of malware spread because it has been observed only in a handful of outbreaks (e.g., one Zeus variant in 2009 [6]).

Another assumption is that the distribution of website popularity is time invariant, which is true in general, although the popularity of individual websites can vary over time [16]. However, the popularity of infected websites may change over time when attackers attempt to promote compromised websites in search-engine rankings [14]. In future, with additional information, it may be possible to accurately model this[website popularity] aspect. We believe, however, that even with more accurate information, the heavy tailed nature of popularity will cause similar heavy tailed behavior in infection exposure and traffic loss.

We also assume that users treat all search results as equal, differing only by ranking. This is likely untrue in the case where users are searching for a specific website, and there could be other effects, such as an abrupt cut-off after the first page of results. More data is needed on the exact nature of user-responses. It is possible that depreferencing could be implemented not a as a reduction in ranking, but by some other mechanism. For example, a search engine could provide multiple warnings with different degrees of difficulty to navigate for infected websites. We leave these aspects to future work.

We make the assumption that the recovery rate from false positives and actual infections is the same. It is plausible that actual infections could exhibit other signs which would warn website administrators of an infection, speeding recovery. Conversely, malware could actively hide itself, slowing recovery. Recovery rates might changed based on the type of intervention taken. In blacklisting, a sudden loss of traffic might warn administrators about infections faster than depreferencing's slow reduction in traffic. The search provider could warn the website administrator that its traffic is being altered.[9] This is especially important if higher false positive rates are to be tolerated. Different recovery rates based on these scenarios could be incorporated into the model, but we leave this analysis for future work.

Another area of future work would be to focus on infections that spread in a general network environment where a referral service (such as search) plays a key role. Similar interventions could be applied when infections are spread from website to website, rather than simply exposing a client population. This could be a particularly good model for controlling infections of malicious software in online social networks.

In our analysis and modeling we disregard the effect of false negatives, primarily because we assume that the response methods we explore use the same detection mechanisms, subject to the same false negative rates. Usually, in real detection systems, reducing the accuracy of the system by increasing false positives usually leads to a *decrease* in false negatives, a feature which gives rise to the traditional ROC curve. We have insufficient data to model this effect, but it suggests that the depreferencing mechanism could have additional benefits beyond those shown by the model: increasing tolerance of false positives could also improve the rate of detection of compromised sites.

Our focus in this research has been to develop a plausible model that allows us to assess the impact of different interventions on the spread of drive-by-download malware. Our goal is to show that modeling can be a useful tool for search providers to use when considering different interventions. We do not have access to data that could enable us to make quantitative predictions about interventions. We expect search providers to have much more relevant data, especially information on the distribution of website popularity, the efficacy of infected website detection, the recovery times for infection and the behavior of users.

8. CONCLUSION

By building and analyzing plausible models, like the one presented in this work, we are able to better understand where search providers and web administrators should focus their efforts for reducing infections, while avoiding the large-scale (and potentially expensive) experiments needed to test interventions in the field. When there is a high variance in the underlying distributions, such as the website popularity, corresponding high variance in outcomes can make it difficult to assess the comparative effectiveness of interventions in one-off field experiments.

[9] Administrators are likely to ignore automated warnings [32].

We proposed and explored a novel intervention strategy, called *depreferencing*, where a possibly infected website is moved down in the search results, rather than outright blacklisted. Depreferencing may be an attractive alternative to blacklisting for search providers because it allows them to use less precise detection methods with higher false positive rates, potentially increasing the speed of response to infection and reducing the cost of detection. These results imply great difficulty in determining empirically whether certain website interventions are effective, and it suggests that theoretical models such as the one described in this paper have an important role to play in improving web security.

9. ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of DOE grant DE-AC02-05CH11231. Stephanie Forrest acknowledges partial support of DARPA(P-1070-113237), NSF(EF1038682, SHF0905236), and AFOSR(Fa9550-07-1-0532). Benjamin Edwards would like to thank Terri Oda for her many useful comments.

10. REFERENCES

[1] L. A. Adamic and B. A. Huberman. Power-law distribution of the world wide web. *Science*, 287:2115, 2000.

[2] A. Clauset, C. Shalizi, and M. Newman. Power-law distributions in empirical data. *Arxiv preprint arxiv:0706.1062*, 2007.

[3] M. Cova, C. Kruegel, and G. Vigna. Detection and analysis of drive-by-download attacks and malicious JavaScript code. In *Proc. WWW '10*, pages 281–290, 2010.

[4] C. Curtsinger, B. Livshits, B. Zorgn, and C. Seifert. ZOZZLE: Fast and precise in-browser JavaScript malware detection. In *Proc. 20th USENIX Security Symp.*, Aug. 2011.

[5] S. Egelman, L. F. Cranor, and J. Hong. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 1065–1074, New York, NY, USA, 2008. ACM.

[6] J. Erasmus. Compromised ftp details being exploited by in the wild malware, June 2009. `http://www.prevx.com/blog/132/Compromised.html`.

[7] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, and V. Shmatikov. The most dangerous code in the world: validating ssl certificates in non-browser software. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 38–49, New York, NY, USA, 2012. ACM.

[8] A. Ghosh and A. Schwartzbard. A study in using neural networks for anomaly and misuse detection. In *Proc. USENIX Security Symp.*, 1999.

[9] B. Gnedenko, A. Kolmogorov, K. Chung, and J. Doob. *Limit distributions for sums of independent random variables*, volume 195. Addison-Wesley Reading, MA:, 1968.

[10] Google. Safe browsing api. `http://code.google.com/apis/safebrowsing/`.

[11] A. Hess, M. Jung, and G. Schfer. Fidran: A flexible intrusion detection and response framework for active networks. In *ISCC '03*, 2003.

[12] S. Hofmeyr, T. Moore, B. Edwards, S. Forrest, and G. Stelle. Modeling Internet-scale policies for cleaning up malware. In *Proc. 10th Workshop on the Economics of Information Security*, 2011.

[13] HP. Immunity manager. Website. http://www.hp.com/rnd/pdfs/ProCurve_Network_Immunity_Manager1_0.pdf.

[14] J. John, F. Yu, Y. Xie, M. Abadi, and A. Krishnamurthy. deSEO: Combating search-result poisoning. In *Proceedings of the USENIX Security Symposium 2011*, San Francisco, CA, 2011.

[15] J. Karlin, J. Rexford, and S. Forrest. Pretty good bgp: Improving bgp by cautiously adopting routes. In *Proc. CNP '06*, 2006.

[16] S. A. Krashakov, A. B. Teslyuk, and L. N. Shchur. On the universality of rank distributions of website popularity. *Computer Networks*, 50:1769–1780, August 2006.

[17] K. Levchenko, N. Chachra, B. Enright, M. Felegyhazi, C. Grier, T. Halvorson, C. Kanich, C. Kreibich, H. Liu, D. McCoy, A. Pitsillidis, N. Weaver, V. Paxson, G. Voelker, and S. Savage. Click trajectories: End-to-end analysis of the spam value chain. In *Proc. IEEE Sym. and Security and Privacy*, Oakland, CA, 2011.

[18] E. Lin, S. Greenberg, E. Trotter, D. Ma, and J. Aycock. Does domain highlighting help people identify phishing sites? In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 2075–2084, 2011.

[19] H. Liu, K. Levchenko, M. Félegyházi, C. Kreibich, G. Maier, G. M. Voelker, and S. Savage. On the effects of registrar-level intervention. In *Proc. USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, Boston, MA, March 2011.

[20] L. Lu, V. Yegneswaran, P. Porras, and W. Lee. Blade: An attack-agnostic approach for preventing drive-by malware infection. *Proceedings of the 17th ACM Conference on Computer and Communications Security*, 2010.

[21] M. Meiss, B. Gonçalves, J. Ramasco, A. Flammini, and F. Menczer. Modeling traffic on the web graph. In R. Kumar and D. Sivakumar, editors, *Algorithms and Models for the Web-Graph*, volume 6516 of *Lecture Notes in Computer Science*, pages 50–61. Springer Berlin / Heidelberg, 2010.

[22] T. Moore, N. Leontiadis, and N. Christin. Fashion crimes: Trending-term exploitation on the web. In *Proc. ACM CCS'11*, Chicago, IL, Oct. 2011.

[23] A. Moshchuk, T. Bragin, S. D. Gribble, and H. M. Levy. A crawler-based study of spyware in the web. In *NDSS*, 2006.

[24] Niels Provos. Safe browsing - protecting web users for 5 years and counting. `http://googleonlinesecurity.blogspot.com/2012/06/safe-browsing-protecting-web-users-for.html`.

[25] U. Parasites. Practical guide to dealing with Google's malware warnings. `http://www.unmaskparasites.com/malware-warning-guide/`.

[26] N. Provos, P. Mavrommatis, M. Rajab, and F. Monrose. All your iFrames point to us. In *Proc. 17th USENIX Security Symp.*, Aug. 2008.

[27] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu. The ghost in the browser: Analysis of web-based malware. In *Proc. 1st USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, Cambridge, MA, Apr. 2007.

[28] M. Rajab, L. Ballard, N. Jagpal, P. Mavrommatis, D. Nojiri, N. Provos, and L. Schmidt. Trends in circumventing web-malware detection. Technical report, Google, July 2011. `http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/archive/papers/rajab-2011a.pdf`.

[29] A. Somayaji and S. Forrest. Automated response using system-call delays. In *In Proceedings of the 9th USENIX Security Symposium*, pages 185–197, 2000.

[30] A. Sotirakopoulos, K. Hawkey, and K. Beznosov. I did it because i trusted you": Challenges with the study environment biasing participant behaviours. In *SOUPS Usable Security Experiment Reports (USER) Workshop*, 2010.

[31] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying wolf: an empirical study of ssl warning effectiveness. In *Proceedings of the 18th conference on USENIX security symposium*, SSYM'09, pages 399–416, Berkeley, CA, USA, 2009. USENIX Association.

[32] M. Vasek and T. Moore. Do malware reports expedite cleanup? an experimental study. In *Proceedings of the 5th USENIX conference on Cyber Security Experimentation and Test*, pages 6–6. USENIX Association, 2012.

[33] J. Voit. *The statistical mechanics of financial markets*. Springer Verlag, 2005.

[34] Y.-M. Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, and S. T. King. Automated web patrol with strider honeymonkeys: Finding web sites that exploit browser vulnerabilities. In *NDSS*, 2006.

[35] WhiteHat Security. Whitehat website security statistic report: Winter 2011, 11th edition. Technical report, 2011. `https://www.whitehatsec.com/assets/WPstats_winter11_11th.pdf?doc=WPstats_fall10_10th`.

[36] M. M. Williamson. Throttling viruses: Restricting propagation to defeat malicous mobile code. In *Proc. ACSAC '02*, Las Vegas, Nevada, Dec. 2002.

[37] C. Wong, S. Bielski, A. Studer, and C. Wang. On the effectiveness of rate limiting mechanisms. In *Proc. RAID '05*, 2005.

[38] I. Zaliapin, Y. Kagan, and F. Schoenberg. Approximating the distribution of pareto sums. *Pure and Applied Geophysics*, 162(6):1187–1228, 2005.

[39] C. Zou, W. Gong, and D. Towsley. Code red worm propagation modeling and analysis. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 138–147. ACM, 2002.