World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:1, No:10, 2007

Distributed Detection and Optimal Traffic-blocking of Network Worms

Zoran Nikoloski, Narsingh Deo, and Ludek Kucera

Abstract**—**Despite the recent surge of research in control of worm propagation, currently, there is no effective defense system against such cyber attacks. We first design a distributed detection architecture called *Detection via Distributed Blackholes* (*DDBH*). Our novel detection mechanism could be implemented via virtual honeypots or honeynets. Simulation results show that a worm can be detected with virtual honeypots on only 3% of the nodes. Moreover, the worm is detected when less than 1.5% of the nodes are infected. We then develop two control strategies: (1) optimal dynamic traffic-blocking, for which we determine the condition that guarantees minimum number of removed nodes when the worm is contained and (2) predictive dynamic traffic-blocking—a realistic deployment of the optimal strategy on scale-free graphs. The predictive dynamic traffic-blocking, coupled with the DDBH, ensures that more than 40% of the network is unaffected by the propagation at the time when the worm is contained.

Keywords**—**Network worms, distributed detection, optimal traffic-blocking, individual-based simulation.

I. Introduction

Cyber attacks with network worms have become a common threat to the crucial information networks. The continuous increase in the number and maliciousness of pervasive worms has inflicted considerable damage in terms of unsolicited consumption of network bandwidth, degraded corporate productivity (as a result of nonfunctional networks with thousands of computers), and compromised integrity of valuable data. The global impact that worms have on today's network-dependent society has, therefore, spurred great interest in understanding not only the dynamics of their propagation but also the feasibility of worm detection and control strategies.

A necessary requirement for any control strategy is timely and accurate detection of the worm. Despite the recent surge of research in detection and control of worm propagation, currently, there is no effective defense (control) system against network worms. The existing countermeasures (*e.g.*, anti-virus software, firewalls, software-patching) have been deemed inadequate for effective control of worms [9], [19], [20]. Therefore, devising new control strategies, coupled with an effective automated detection mechanism, is a first step towards a comprehensive network-security solution.

A computer is considered *infected* if it hosts a replica of the worm. An infected computer is *infectious* if the worm (on this computer) scans other computers for the exploited vulnerability. The only immediately observable effect of worm propagation is an increase in the routing-related requests, as the worm keeps scanning different computers [3], [10]. Moreover, recent studies [15] have found the topology on which the worm propagates to be significantly different from the underlining network infrastructure (*e.g.*, the Internet). This is true only for network worms that scan the IP space uniformly at random. Although the introduction of the Internet has arguably made the assumption of sparseness (of the propagation topology) no longer valid, worms propagating in IPv6 will have to use *local propagation* (*e.g.*, sub-net and permutation strategies [18]).

Our contribution here is twofold: (1) mechanism for distributed detection of worms with various propagation strategies. The proposed **Distributed Detection via Blackholes** (DDBH) architecture and detection algorithm employs small blocks of unused IP addresses. Our approach allows for studying the optimal placement of virtual honeypots and its feasibility in detecting the worm, (2) analysis of an optimal control strategy and its near-optimal version for practical deployment—**predictive dynamic traffic-blocking**. The latter employs information about the underlying network structure and characteristics of the worm discerned by the proposed detection mechanism.

II. Detection via Distributed Blackholes

Recently developed approaches for automatic extraction of worm signatures include detection through honeypots, virtual honeypots [17], honeynets [11], and blackholes (also known as network telescopes [13], [21] or darknets). By using these techniques, the traffic observed at unused blocks has successfully been characterized. In these approaches, a portion of (unused) address space is globally announced and routed to a collection infrastructure that records incoming and/or outgoing packets. All captured activities are assumed to be

Manuscript received December 8, 2005. This work was supported in part by the EU DELIS project.

Zoran Nikoloski was a Ph.D. Candidate at the School of Computer Science, University of Central Florida, Orlando, FL 32816. He is now a visiting researcher at the Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Malostranske nam. 25, Prague 1, Czech Republic (e-mail: nikoloski@kam.mff.cuni.cz).

Narsingh Deo is with the School of Computer Science, University of Central Florida, Orlando, FL 32816 (e-mail: deo@cs.ucf.edu).

Ludek Kucera is with Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic (e-mail: ludek@kam.mff.cuni.cz).

unauthorized or malicious as any connection initiated inbound or outbound to these four systems is most likely a result of mis-configuration, or scanning from worms and other network probing. The *worm signatures*—a string of bytes in the traffic that passes through a network link—obtained from traffic analysis, can be used to devise agents that block attacks on real-world networks. However, data from distributed sensors is used in [2] to present evidence that large, distributed unused address-blocks observe significantly different traffic patterns.

Our detection mechanism, *Detection via Distributed Blackholes* (*DDBH*) (1) belongs to the group of threshold-based algorithms, (2) operates via network traffic monitoring, and (3) uses distributed collection of unused address blocks (known as blackholes). The last characteristic renders DDBH deployable via (virtual) honeypots or honeynets. Moreover, DDBH provides the basis for coordinated defense by using only locally available information.

The DDBH architecture is a set of heterogeneous blackhole sensors, aggregators, and responders. Each *blackhole sensor* monitors a dedicated range of unused IP addresses. For each packet sent to the blackhole, the sensor records the source IP address, destination IP address, and the destination port. Because there are no legitimate hosts in an unused address block, the traffic must be a result of poor routing management or scanning/probing activity. Each blackhole sensor is responsible for gathering and storing data, performing queries on its local storage, and generating alerts that are sent to the aggregators. The blackhole sensor looks for erratic activities such as: horizontal scan, vertical scan, or coordinated scans, whose characterization has already been studied [4], [5]. *Aggregators* communicate with the sensors to gather information about the global characteristics of the propagation and plan further actions, *e.g.*, alarming certain responders. Finally, *responders*, through two-way communication with analyzers and aggregators, initiate a pre-specified control strategy.

Let τ denote the threshold for the number of unused IP addresses on which scanning attempts are detected and t_d be the time when the threshold is exceeded. Each *aggregator* keeps a list of pairs (*destination address*, *source address*). When a blackhole sensor detects scanning activity for a particular destination address, it sends the pair of "infected" blackhole address and infectious (source) address to the aggregator. When the number of blackhole destination address, on which scanning was attempted, exceeds the threshold, the aggregator activates the *responders* responsible for handling the associated source addresses. The responders, in turn, initiate a pre-specified control strategy. A responder that has been activated updates its actions based on the data from the corresponding sensor and aggregator. Therefore, a particular control strategy launched by a given responder can be terminated and redirected to another part of the network.

Worms propagate via network communications in a similar way as a virus spreads among people. The DDBH detection mechanism can therefore be formalized via a modification of the Susceptible-Infectious model [1]. Since network communications can be modeled by a graph, in which nodes represent hosts and edges are communication lines, we will use graph-theoretic terms to describe the DDBH algorithm: The propagation takes place on a graph $G=(V,E)$, representing the communication among Autonomous Systems (ASes), with n nodes (ASes) and m edges (communication links). Nodes are divided into two groups—*regular nodes*, representing used IP addresses, and $\varnothing$*-nodes*, modeling unused IP addresses (blackholes). At any time moment, a regular node can be either susceptible or *infectious*, while a $\varnothing$-node can be either susceptible or *infected*. Note that a $\varnothing$-node is never infectious, and thus, it does not affect the propagation. The DDBH algorithm is formally described in Fig. 1.

```
Algorithm Detection via Distributed Blackholes
Input:
$L_\varnothing$, list of pairs (infected $\varnothing$-node, infectious (regular) node neighbor)
$\vartheta$, number of elements in $L_\varnothing$
if pair ($\varnothing$-node $u$, $v$) received from a sensor/child-
  aggregator then
    $\vartheta \leftarrow \vartheta + 1$
end if
if $\vartheta > \tau$ then
    $\vartheta \leftarrow 0$
    empty $L_\varnothing$
    start control strategy at responders responsible
      for $L_\varnothing$
end if
```

Fig. 1. Detection via Distributed Blackholes algorithm

The problem is that of where to place the $\varnothing$-nodes and how many of them to place. We note that the study in [14] concluded that almost all network paths should be monitored in order to effectively control the worm propagation. Given a graph G, let $V'(G)$ be the minimum vertex cover of G. Here, for every node u of highest degree from $V'(G)$, a new $\varnothing$-node with same neighbors as u is added to G (*i.e.*, node u is *copied*). Thus, the $\varnothing$-nodes are added by copying the first $n_\varnothing$ nodes of highest degree in $V'(G)$. As the problem of finding a minimum vertex cover of G is NP-hard, we use a heuristic that at each step chooses a node (not in $V'(G)$) for inclusion in the vertex cover (such heuristic is known to give log-approximation [7]).

The size of the vertex covers for five Autonomous System (AS) graphs are shown in Fig. 2. Given an AS graph G, the number, $n_\varnothing$, of $\varnothing$-nodes added to G is 1% (respectively, 2%) of the order of G when $\varnothing$-nodes comprise 4% (respectively, 8%) of the highest-degree nodes in the vertex cover.

Graph	Order	Size of Vertex Cover (VC)	4% of VC		8% of VC	
AS 08.11.1997	3015	588	24	0.78%	47	1.56%
AS 02.20.1998	4180	829	33	0.79%	66	1.59%
AS 02.07.1999	5357	964	39	0.72%	77	1.44%
AS 02.07.2000	7956	1266	51	0.64%	101	1.27%
AS 03.16.2001	10515	1640	66	0.62%	131	1.25%

Fig. 2 Number of nodes in vertex cover of Autonomous system graphs

Three sets of experiments were performed for each graph and each number of $\varnothing$-nodes added when τ = 1, 3, or 6. An increase in the value of τ, renders the DDBH algorithm more sensitive to false positives, but at the same time, it increases the time before the worm is detected. By varying the parameter τ, one can determine how the DDBH algorithm performs for the three levels of sensitivity to false alarms. With the help of an individual-based simulation we were able to answer questions related to: (1) the time, t_d, necessary to detect the worm, and (2) the number of infectious nodes when the worm is detected. The results of the empirical study for the AS graphs from 08.11.1997 and 02.07.2000 are shown in Fig. 3.

AS 08.11.1997		τ = 1	8% of vertex cover monitored	
β	average td	σ td	average % I(td)	σ % I(td)
1.8	0.271	0.235	0.167%	0.190%
1.5	0.321	0.296	0.154%	0.174%
0.9	0.507	0.419	0.149%	0.177%
		τ = 3	8% of vertex cover monitored	
1.8	0.534	0.450	0.654%	0.392%
1.5	0.572	0.388	0.582%	0.389%
0.9	0.890	0.671	0.632%	0.434%
		τ = 6	8% of vertex cover monitored	
1.8	0.611	0.437	1.210%	0.617%
1.5	0.586	0.391	1.115%	0.512%
0.9	0.802	0.517	1.292%	0.511%
AS 02.07.2000		τ = 3	4% of vertex cover monitored	
1.8	0.203	0.150	0.367%	0.284%
1.5	0.287	0.268	0.417%	0.299%
0.9	0.342	0.303	0.417%	0.271%
		τ = 6	4% of vertex cover monitored	
1.8	0.224	0.211	0.783%	0.350%
1.5	0.230	0.172	0.796%	0.321%
0.9	0.368	0.273	0.875%	0.366%

Fig. 3 Statistical analysis of DDBH on Autonomous System graph from 08.11.1997 and 02.07.2000

We observe that with the increase of τ, (i) the time, t_d to detect the worm, and (ii) the number of infectious nodes, $I(t_d)$ when the worm is detected increase. Moreover, the DDBH algorithm could detect the worm when less than 1.5% of nodes are infected.

III. Dynamic Traffic-Blocking Strategies

Traditional epidemiology has identified three factors determining the outcome of an infection [8]: the size of the susceptible population, the length of the infectious period, and the rate of infections. Like with diseases, there are two potential approaches to mitigate network worms: *prevention*, that includes technologies for reducing the size of the susceptible population, and *control*, that consists of strategies for reducing any of the three factors determining the outcome of the propagation. As system design and implementation is prone to human (logic) errors, any prevention technique, by itself, cannot suffice in countering network worms. In absence of a patch or worm signature, *quarantining mechanisms* (*e.g.*, address-blacklisting [12], [14] and traffic-blocking) can prevent the worm from propagating by disabling communication directed from a computer, suspected to be or detected as, infectious.

We distinguish between *control mechanisms* and *control strategies*. A *control strategy* specifies when and which of the responders are activated. A *control mechanism* specifies how the responders act towards hindering the propagation, *i.e.*, it specifies how a given control strategy is implemented. Here, we focus on two quarantining strategies by traffic-blocking: optimal dynamic traffic-blocking and near-optimal predictive dynamic traffic-blocking. A variation of the Susceptible-Infectious-Removed (SIR) model [1] is used in the analysis of the first control strategy, whereas, the analysis of the second employs an individual-based simulation. The optimization parameter is the loss to the population, expressed through the number of removed susceptible nodes.

Notation: Let $S(t)$ denote the number of susceptible nodes at time t, $I(t)$ the number of infectious nodes, and $R(t)$ denote the number of removed nodes. Lowercase letters will denote the fraction of nodes in a given state. Let β denote the rate with which an infectious node infects its adjacent susceptible nodes, and γ be the rate with which infectious nodes are removed.

A. Optimal Dynamic Traffic-Blocking

The effectiveness of a control strategy, modeled by the SIR, is usually measured in terms of its ability to reduce the average number of new infections produced by an infectious node (during its infectious period) if placed in a population of susceptible nodes. This quantity, known as the *\basic reproductive ratio* R_0, can be expressed as the ratio $\bar{d}\beta/\gamma$, when the network is a $\bar{d}$-regular graph. It is well known from epidemiological studies [1] that when $R_0 > 1$, the number of infections will grow, whereas if $R_0 < 1$, the new infections, on average, will decline and major epidemics cannot occur. Therefore, any control policy should aim at reducing the value of R_0 below one. There are, however, further potential requirements for a control strategy—for instance, spatial containment of the propagation (shown to be NP-hard problem [16]), reduction of the propagation duration, minimization of overall losses to the population, or a combination of these requirements. Interesting, and, yet not investigated are strategies that minimize losses (*i.e.* the number of removed nodes) to the population.

Our dynamic traffic-blocking is a combination of: (1)

dynamic removal (traffic-blocking) of infectious nodes when detected, and (2) dynamic removal of susceptible nodes whose history (*e.g.*, recent established connections) suggests an enhanced risk for getting infected. This strategy prevents further transmission from infectious nodes, but may also result in removal of some susceptible nodes. The latter may leads to a trade-off: increased levels of control result in a greater reduction in transmission, but also in an increase in the number of removed susceptible nodes. Let t_q be the time that traffic is blocked to/from a susceptible node. Soft quarantining with duration t_q has already been studied by Zou [22]. Here, we focus on the optimal level of quarantining by traffic-blocking when $t_q \to \infty$, in order to minimize the overall losses expressed through the number of removed nodes.

We consider the SIR model, where removed nodes arise from: (1) traffic-blocking of detected infectious nodes at rate γ, and (2) traffic-blocking, at rate c, of nodes that has not yet been identified as infectious but whose history shows that they are at greater risk of getting infected. Implementation of this control strategy will remove a fraction f of nodes at risk from a given infectious nodes; since increases levels of control (*i.e.*, larger c) may require greater fraction of susceptible nodes to be removed, we assume that f is a function of c. In turn, this will result in probability $\beta\left(1-f(c)\right)s(t)$ of finding a susceptible node that has not been quarantined. The model on a graph, whose average degree is $\bar{d}$, can be written as:

$$\begin{aligned}\frac{\mathrm{d}s(t)}{\mathrm{d}t} &= -\beta\bar{d}\left(1-f(c)\right)s(t)i(t)-c\bar{d}s(t)i(t),\\ \frac{\mathrm{d}i(t)}{\mathrm{d}t} &= \beta\bar{d}\left(1-f(c)\right)s(t)i(t)-\gamma i(t),\\ \frac{\mathrm{d}r(t)}{\mathrm{d}t} &= c\bar{d}s(t)i(t)+\gamma i(t).\end{aligned} \tag{1}$$

The general results about the SIR model given in equation (1) are: (1) an epidemic can occur only when $R_0=\frac{\beta\bar{d}\left(1-f(c)\right)}{\gamma}>1$, (2) the function $S(t)$ is monotonically decreasing, $R(t)$ is monotonically decreasing, and $I(t)$ is unimodal (has one maximum), (3) The epidemic eventually dies out, with some proportion of susceptible remaining, given by:

$$s(t)=e^{\left(s(t)-1\right)R_0}, \qquad t\to\infty.$$

Let the number of nodes quarantined during an infectious period of an infectious node be denoted by $p=c/\gamma$. The final fraction of the removed nodes can be determined by dividing the first by the second equation from the system (1) and integrating over I, to obtain:

$$\begin{aligned}&R_0\left(S(\infty)-S(0)\right)-n\ln\frac{S(\infty)}{S(0)}=\\ &=\left(-R_0-p\right)\left(I(\infty)-I(0)\right).\end{aligned} \tag{2}$$

Further, by employing the final relationship from model in equation (1), one can get:

$$\begin{aligned}&\ln\frac{n-R(\infty)-I(\infty)}{n-I(0)}=\\ &=R_0\frac{I(0)-R(\infty)-I(\infty)}{n}+\\ &+\left(R_0+p\right)\frac{I(\infty)-I(0)}{n},\end{aligned}$$

or equivalently:

$$r(t)=1-\left(1-i(0)\right)e^{-R_0 r(t)-pi(0)}, \quad t\to\infty. \tag{3}$$

Since the fraction of removed nodes that are expected to turn into infectious (and could spread the worm in case they were not quarantined) is $\frac{R_0}{R_0+p}$, the final fraction of infectious quarantined nodes can be expressed as:

$$r(t)-r_s(t)=\frac{R_0}{R_0+p}\left(r(t)-i(0)\right)+i(0), t\to\infty. \tag{4}$$

Theorem 1. *The total fraction of removed nodes, at the end of propagation, decreases with the increase of the control parameter* c, *if* $\beta\frac{\mathrm{d}f(c)}{\mathrm{d}c}>\frac{i(0)}{r(t)}$.

Proof: The function $r(t)$, $t\to\infty$ depends on c, through $f(c)$ and $p=c/\gamma$. Therefore, we investigate how $r(t)$, $t\to\infty$, changes in respect to the increase of c. To locate the minimum, we look at the conditions under which $\frac{\mathrm{d}r(t)}{\mathrm{d}c}<0$. By differentiating equation (3) with respect to c, one obtains for $\frac{\mathrm{d}r(t)}{\mathrm{d}c}$:

$$\frac{r(t)}{\gamma}\left(1-i(0)\right)e^{-R_0 r(t)-pi(0)}\left[\frac{i(0)}{r(t)}-\beta\frac{\mathrm{d}f(c)}{\mathrm{d}c}\right]. \tag{5}$$

The sign of $\dfrac{\mathrm{d}r(t)}{\mathrm{d}c}$ is determined by the last multiplicand of the right-hand side in equation (5). It follows that $\dfrac{\mathrm{d}r(t)}{\mathrm{d}c}<0$ if and only if $\beta\dfrac{\mathrm{d}f(c)}{\mathrm{d}c}>\dfrac{i(0)}{r(t)}$. □

To conclude: (1) the analysis of model (1) shows that the amount of losses in the population are determined by the function $f(c)$, expressing the fraction of removed nodes that are at risk from an infectious node. This is directly quantified through the change of the basic reproductive ratio $R_0=\dfrac{\beta\bar{d}(1-f(c))}{\gamma}$, and (2) the first derivative $\dfrac{\mathrm{d}r(t)}{\mathrm{d}c}$ determines whether or not an increase in the value of c will cause a decrease in the final fraction of removed nodes $r(t)$—as stated by Theorem 1.

B. *Near-Optimal Predictive Traffic-Blocking*

The predictive dynamic traffic-blocking could be applied when *neither patch nor worm signature is available*. This control strategy employs information about the size and the state of the nodes in a local neighborhood. This strategy is a practical realization of the optimal control strategy analyzed in Section A (the optimal strategy holds on $\bar{d}$-regular graphs, while realistic AS graphs are scale-free [6]). Based on the available information, the predictive strategy assesses the *risk* for a node to become infectious. The *risk* for a node to become infectious is described as a function of probability of becoming infectious and the consequence of being infectious. The probability for a node to become infectious is determined by the number of infectious nodes in the local neighborhood of the node. The consequence of being infectious is determined by the degree of the node—the higher the degree, the greater the effect of propagation from that node. Before presenting the detailed description of the predictive dynamic traffic-blocking, we give some definitions:

Definition 1. For a node u and an integer l, the *local neighborhood* of u, denoted by $N(u)$, is composed of all nodes whose distance from u is no greater than l, *i.e.*

$$N(u)=\{v:d(u,v)\le l\}.$$

Note that $N(u)=\bigcup_{j=1}^{l}N_j(u)$, where the j th-neighborhood $N_j(u)=\{v:d(u,v)=j\}$. The set of infectious nodes in $N_j(u)$ will be denoted by $N_j^i(u)$.

Definition 2. Given a node u, let $\bigcup_{j=1}^{l}N_j^i(u)$ be the set of infectious nodes in $N(u)$. The *risk* of u becoming infectious, denoted by $\rho(u)$, is the weighted sum

$$\rho(u)=\frac{d(u)}{n}\frac{\sum_{j=1}^{l}\frac{1}{j}\left|N_j^i(u)\right|}{\sum_{j=1}^{l}\frac{1}{j}\left|N_j(u)\right|}.$$

Addition of one infectious node in $N(u)$, say at distance k, $1\le k\le l$, increases the risk by

$$\frac{d(u)}{n}\frac{1}{k\sum_{j=1}^{l}\frac{1}{j}\left|N_j(u)\right|}. \tag{6}$$

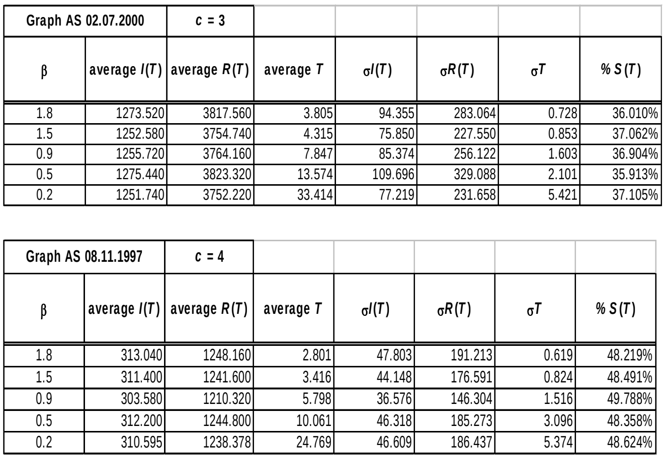
The predictive dynamic traffic blocking is described as follows: Let τ_{ptb} be a given threshold, c be the number of susceptible nodes on which the strategy is applied, and ϑ be a counter of new infectious nodes. Predictive dynamic traffic-blocking is a greedy algorithm that selects c susceptible nodes with *highest risk* of being infected, once the counter ϑ exceeds the value of τ_{ptb}. Whenever an infectious node v is detected, the risk of every susceptible node u in $N(v)$ is updated, by using equation (6). The algorithm is formally given in Fig. 4, below.

```
Algorithm Predictive Dynamic Control
Input:
$G$, graph
$l$, integer
$c$, number of susceptible nodes to be removed
$V_I$, set of infected node
Output:
$V_{R_s}$, list of susceptible nodes
for every node $u \in V(G)-V_I$ do
    calculate $\rho(u)$
    $L \leftarrow L \cup \{(u,\rho(u))\}$
end for
sort $L$ in decreasing order of $\rho(u)$
return the first $c$ elements of $L$
```

Fig. 4 Greedy algorithm for predictive dynamic control

An individual-based simulation of the stochastic (local) propagation process, combined with simulation of the control strategy, is used in the empirical study. Three sets of experiments are performed for each graph and each control strategy: (1) propagation is initiated at a node of maximum degree, (2) random node is chosen as an initial node, while in the third set, (3) propagation starts at a node of minimum degree. In each experiment, for given a given simulation parameters, we perform a simple statistical analysis of the results to estimate the effect of stochastic fluctuations. Here, we use local neighborhood with $l = 2$ that models limited information about the network environment, and $\tau_{ptb} = 1$, *i.e.*, c susceptible nodes are removed per detected infectious node. The results for the first case, presented in Figures 5 and 6, is of particular importance, because it can be use to estimate the effectiveness of a control strategy when the worm has the greatest chance for wide-spread propagation. With the help of the individual-based simulation, we are able to answer the following question regarding each control strategy described in Section III: (1) the maximum number of infectious nodes, (2) the necessary number of removals to contain the propagation, (3) the time T required for containment, and (4) the number of susceptible nodes at time T, when the worm is contained due to traffic-blocking, whose averages over 5000 simulations as shown in Figures 5 and 6.

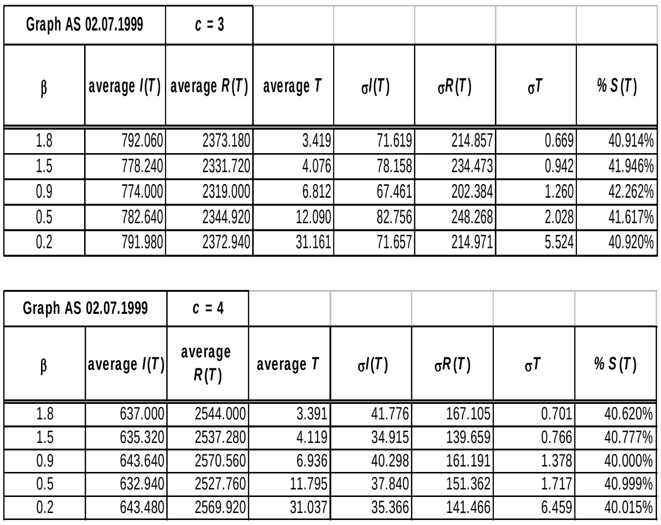
In light of Theorem 1, we also study how the increase in the number, c, of susceptible nodes to which traffic-blocking is applied, might improve the effectiveness of the predictive dynamic traffic-blocking. For the case when $\tau_{ptb} = 1$, $c = 4$ is the optimum value. For this value of c, the predictive dynamic traffic-blocking results in the minimum number of removed nodes and a fraction of susceptible nodes greater than 40% at the end of the propagation.



Graph AS 02.07.2000	c = 3						
β	average $I(T)$	average $R(T)$	average T	σ$I(T)$	σ$R(T)$	σT	% $S(T)$
1.8	1273.520	3817.560	3.805	94.355	283.064	0.728	36.010%
1.5	1252.580	3754.740	4.315	75.850	227.550	0.853	37.062%
0.9	1255.720	3764.160	7.847	85.374	256.122	1.603	36.904%
0.5	1275.440	3823.320	13.574	109.696	329.088	2.101	35.913%
0.2	1251.740	3752.220	33.414	77.219	231.658	5.421	37.105%

Graph AS 08.11.1997	c = 4						
β	average $I(T)$	average $R(T)$	average T	σ$I(T)$	σ$R(T)$	σT	% $S(T)$
1.8	313.040	1248.160	2.801	47.803	191.213	0.619	48.219%
1.5	311.400	1241.600	3.416	44.148	176.591	0.824	48.491%
0.9	303.580	1210.320	5.798	36.576	146.304	1.516	49.788%
0.5	312.200	1244.800	10.061	46.318	185.273	3.096	48.358%
0.2	310.595	1238.378	24.769	46.609	186.437	5.374	48.624%

Fig. 5 Statistical analysis of the predictive dynamic traffic-blocking strategy on the Autonomous System graph from 02.07.2000, where propagation starts at node of degree 1772



Graph AS 02.07.1999	c = 3						
β	average $I(T)$	average $R(T)$	average T	σ$I(T)$	σ$R(T)$	σT	% $S(T)$
1.8	792.060	2373.180	3.419	71.619	214.857	0.669	40.914%
1.5	778.240	2331.720	4.076	78.158	234.473	0.942	41.946%
0.9	774.000	2319.000	6.812	67.461	202.384	1.260	42.262%
0.5	782.640	2344.920	12.090	82.756	248.268	2.028	41.617%
0.2	791.980	2372.940	31.161	71.657	214.971	5.524	40.920%

Graph AS 02.07.1999	c = 4						
β	average $I(T)$	average $R(T)$	average T	σ$I(T)$	σ$R(T)$	σT	% $S(T)$
1.8	637.000	2544.000	3.391	41.776	167.105	0.701	40.620%
1.5	635.320	2537.280	4.119	34.915	139.659	0.766	40.777%
0.9	643.640	2570.560	6.936	40.298	161.191	1.378	40.000%
0.5	632.940	2527.760	11.795	37.840	151.362	1.717	40.999%
0.2	643.480	2569.920	31.037	35.366	141.466	6.459	40.015%

Fig. 6 Statistical analysis of the predictive dynamic traffic-blocking strategy on the AS graph from 02.07.1999, where propagation starts at node of degree 1193

IV. Conclusion

Despite the recent surge of research in control of worm propagation, currently, there is no effective defense system against such cyber attacks. A prerequisite for a defense system is its ability to identify the presence of the worm on the network. Here, we first design a distributed detection architecture called Detection via Distributed Blackholes (DDBH). Simulation results show that the worm can be detected when DDBH comprises only 3% of the network. Moreover, the worm is detected when less than 1.5% of the nodes are infected. We also study two control strategies: optimal dynamic traffic-blocking and predictive dynamic traffic-blocking. For the optimal traffic-blocking, we determine the condition that guarantees minimum number of removed nodes when the worm is contained. For c, the number of susceptible nodes removed per new infectious node, of value 4, the predictive dynamic traffic-blocking on scale-free graphs results in minimum number of removed nodes, in line with the analysis of the optimal traffic-blocking on $\overline{d}$-regular graphs. Moreover, a worm can be contained when more than 40% of the network computers are still susceptible.

References

[1] R. M. Anderson and R. M. May, *Infectious Diseases in Humans*, Oxford University Press, 1992.

[2] E. Cooke, M. Bailey, Z. Morley Mao, and D. McPherson, *Toward Understanding Distributed Blackhole Placement*, Proceedings of the ACM Workshop on Rapid Malcode, 2004, pp. 54–54.

[3] J. Cowie, A. T. Ogielski, B. J. Premore, and Y. Yuan, *Global Routing Instabilities Triggered by Code Red II and Nimda. Available at: www.renesys.com*, (2001).

[4] M. de Vivo, E. Carrasco, G. Isern, and G. de Vivo, *A Review of Port Scanning Techniques*, Operating Systems Review 29 (1999), no. 2, 41–48.

[5] M. de Vivo, G. de Vivo, R. Koeneke, and G. Isern, *Internet Vulnerabilities Related to TCP/IP and T/TCP, Internet Security Attacks at the Basic Level*, Operating Systems Review 32 (1998), no. 2, 4–15.

[6] M. Faloutsos, P. Faloutsos, and C. Faloutsos, *On Power–Law Relationships of the Internet Topology*, Proceedings of SIGCOMM, 1999.

[7] M. R. Garey and D. S. Johnson, *Computers and Intractability*,W. H. Freeman, 1999.

[8] H. W. Hethcote, *Mathematics of Infectious Diseases*, SIAM Review 42 (2000), no. 4, 599–653.

[9] Computer Security Institute, *Ninth Annual Computer Crime and Security Survey. Available at: i.cmpnet.com*, (2004).

[10] K. Lan, A. Hussain, and D. Dutta, *Effects of Malicious Traffic on the Network*, Proceedings of PAM'03, 2003.

[11] J. Levine, R. LaBella, H. Owen, D. Contis, and B. Culver, *The Use of Honeynets to Detect Exploited Systems Across Large Enterprise Networks*, Proceedings of the 2003 IEEE Workshop on Information Assurance, 2003.

[12] M. Liljenstam and D. M. Nicol, *Comparing Passive and Active Worm Defenses*, Proceedings of the First International Conference on the Quantitative Evaluation of Systems (QEST), 2004, pp. 18–27.

[13] D. Moore, *Network Telescopes. Available at: www.caida.org*, (2003).

[14] D. Moore, G. M. Voelker, C. Shannon, and S. Savage, *Internet Quarantine: Requirements for Containing Self– Propagating Code*, Proceedings of the IEEE INFOCOM, 2003.

[15] D. M. Nicol and M. Liljenstam, *Models of Active Worm Defenses*, Proccedings of the IPSI Studenica Conference, 2004.

[16] Z. Nikoloski and N. Deo, *Complexity of Quarantining Network Worms*, Discrete Applied Mathematics, (submitted).

[17] N. Provos, *A Virtual Honeypot Framework*, Proceedings of the 12th USENIX Security Symposium, 2004, pp. 1–14.

[18] N. Weaver, *Potential Strategies for High Speed Active Worms: A Worst Case Analysis. Available at: brass.cs.berkeley.edu*, (2002).

[19] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, *Large Scale Malicious Code: A Research Agenda. Available at: www.cs.berkeley.edu/ nweaver*, (2003).

[20] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, *A Taxonomy of Computer Worms*, Proceedings of ACM Workshop on Rapid Malcode, 2003.

[21] C. C. Zou, W. Gong, D. Towsley, and D. Gao, *Monitoring and Early Detection for Internet Worms*, Proceedings of the 10th ACM Conference on Computer and Communication Security, 2003.

[22] C. C. Zou, W. Gong, D. Towsley, and D. Gao, *Worm Propagation Modeling and Analysis under Dynamic Quarantine Defenses*, Proceedings of the ACM CCS Workshop on Rapid Malcode, 2003.