Effects of Security Solutions on Worm Propagation

Mohammad R. Faghani, Hossein Saidi, Mohammad R. Ataei
Department of Electrical and Computer Engineering
Isfahan University of Technology, Isfahan, Iran
faghani@ieee.org, hsaidi@cc.iut.ac.ir, mohamadreza.ataei@ec.iut.ac.ir

Abstract—Automated propagating malwares such as worms have drawn significant attention due to their adverse impact on Internet and network security. According to latest advisories, malware creation has reached epidemic proportions. While much research attention has been paid in developing stochastic models of worm propagation, the real world prevention mechanisms have not been deployed in simulations. Describing the worm propagation cycle in real world and identifying the ways that worms exploit to spread themselves, this article will step through common practices to prevent worm propagation and also examines these results in the analytical worm propagation model.

Keywords- worm; propagation; Anti-Virus; Sticky Honeypots

I. INTRODUCTION

The Internet provides a convenient way to share the information among people. However, it also offers the opportunities for the malware activities, such as propagating malicious programs, e.g., computer worms [1].

Computer worms are malicious programs that self propagate across a network exploiting security or policy flaws in widely-used services [2]. In 2001, the Code Red and Nimda infected hundreds of thousands of computers [3],[4] which cost billions of dollars loss to world societies [5]. Network worms have the potential to infect many vulnerable hosts on the Internet before human countermeasures take place. According to McAfee avert lab report 135,885 unique threat added throughout 2007. 25,438 more detection were added last year than in two previous years combined, Resulting 38% of all detections [6].

This jump is mostly due to the speed at which the new malwares crossed global channels often in a matter of minutes [7]. This problem is based on several reasons, but the main reason is the ongoing expansion of Internet that provides malwares more (and more inexperienced) targets. Thus, there is a need to carefully characterize the spread of worms and develop efficient strategies for worm containment.

Most of studies are based on deterministic epidemic propagation models [8],[9],[10]. Garetto et al. in [1] defined an interactive markov Chains that is able to capture many aspects of propagation, especially the impact of the underlying topology on the spreading characteristics of malware. A recent study shows that without knowing the worm signature in advance or needing to explicitly detect the worm, the branching process model allows developing an effective and automatic worm containment strategy that did not let the worm propagate beyond the early stages of infection [11].

A recent study shows that without advance knowledge of the worm signature in advance or a need to explicitly detect the worm, the branching process model allows developing an effective and automatic worm containment strategy that did not allow the worm propagation beyond the early stages of infection [11].

In this article we used the worm’s propagation strategy to explain the ways malwares e.g. worms use to propagate themselves. Describing prevention mechanisms tailored to worm propagation, the goal of this article is to simulate those mechanisms effects using two-factor worm propagation model [10].

The remainder of this paper is structured as follows. Section II describes what the cycle of worm propagation is. All stages in the propagation cycle have their own specific methods. In section III we will describe these methods separately. Section IV describes prevention mechanisms from two point of views, one is network topological solutions and the other is client side solutions. Section V defines simulation model and section VI explains simulation results.

II. CYCLE OF PROPAGATION

Analyzing most malwares shows us, there are several specific common ways of propagation. By deploying countermeasures tailored to these specific mechanisms it is possible to prevent, contain, and/or slow the propagation of current and future worms. We chose some top threat and widespread worms such as Nimda, Code Red, MyDoom, NetSky, Bagle, Sasser, Blaster and Funlove [12] for our analysis and suggested the model based on their methods of propagation.

Figure 1 shows the generalized model of cyclic worm propagation. As the model shows, there are only three steps for a worm to infect a host.

In the first stage the infected host searches for vulnerable targets. When the target is found the infected host tries to deliver malcode to the selected target. Executing the malcode, the target host would be compromised, making the target an infected host and the cycle goes around.

As the system compromised, some malwares execute additional tasks. Payload refers to those additional task produced by a worm. It may include leaving a backdoor, self replication or performing the Denial of Service Attacks against
websites. Some malwares such as hybrid Worm/Trojan use payloads on compromised systems. To do this, malware writers exploit the propagation speed of worms to distribute their Bot nets. Bot a.k.a. zombie refers to a host that receives commands from its handler. These feature rich Bot nets can be used to perform lots of malicious activities [13].

Compromising the system requires the malcode to be executed. The execution of malcode can be occurred using one of the followings: 1-Email clients vulnerabilities. 2-user interventions. 3-Automatic execution. The latter method happens when the malcode delivery is the result of buffer overrun vulnerability, e.g. blaster exploits DCOM RPC.

As the system compromised the malware may execute some payloads. Payloads effects are viable if the malware itself is removed. For instance, if the malware left a backdoor on the system, this backdoor may live forever even if the malware is removed. Installing kernel level rootkits in some cases requires the operating system to be reinstalled again [17].

IV. PREVENTION MECHANISM

In this section we propose some defense in depth strategies to contain worm propagation methods.

A. Topology Related Solution

In Target Selection phase we saw that there are three common ways that malwares use to propagate. To prevent connection setup between infected host and randomly generated IP Addresses, one can use client side software firewall or Host IPS. For an enterprise network, this solution may not be cost effective, so using Network IPS or hardware firewall is a better solution. Lots of efforts have been done to elevate IPS capabilities such as detection of self decrypting malwares [14] that allows polymorphic worms to be detected within network. Another solution is to deploy honeypots [16]. Honeypots are used to detect traffic anomaly. Some honeypots called Sticky honeypots can be used to slow down Internet worm propagation by putting the malware in holding pattern such as sending TCP window size of zero, e.g. LaBrea Tarpit [15]. Also attacking vulnerable services through buffer overflow can be limited by using perimeter multilayer switches with ACLs. These switches can block outgoing or incoming request to uncommon services. These new techniques can help slowing down worm propagation.

B. Client Related Solution

Deploying client side solution is more effective because propagation starts from infectious hosts. As mentioned in section III targets of attacks are gathered from infected host, e.g. email harvested on local machine are used to send social engineered mail to victims, all top threat worms such as NetSky use their own SMTP Engine [12], to prevent this function, basically one need to write a custom signature on IDS to pick up any SMTP traffic not coming from specified mail servers, another option is using firewall to alert for uncommon SMTP traffic.

Installing Antivirus with heuristic scanning capabilities somewhat ensures detection of zero-day worms. For critical applications, installing rootkit detectors are recommended before installing any softwares [17] because rootkits can even alter API calls and deceive victim user by having no virus found alerts. Deploying patch management solutions [18] would have even better effects on propagation containment, because, most of the active worms use vulnerabilities that are disclosed several months before. By patching vulnerabilities,
the malcode delivery via buffer overrun used for randomly generated IP Addresses would not be effective anymore. The IP Address scanning method is the fastest way that scanning worms used to propagate.

To complete this section it is essential to say that knowledge of users has an important role on defeating malware attacks. Executing social engineered Email Attachments change an innocent victim unwittingly to a devil source of attack. So giving knowledge about security issues to home users would effectively reduce further security problems.

V. SIMULATION MODEL

We used two-factor propagation model for our simulation [10] which is based on Kermack-Mckendrick epidemic model. The following equations describe the model and table I defines the variables.

$$ \frac{dS(t)}{dt} = -\beta(t)S(t)I(t) - \frac{dQ(t)}{dt} $$

(1)

$$ \frac{dR(t)}{dt} = \gamma(t) $$

(2)

$$ \frac{dQ(t)}{dt} = \mu S(t)J(t) $$

(3)

$$ \beta(t) = \beta_0 \left[ 1 - \frac{I(t)}{N} \right]^\eta $$

(4)

$$ N(t) = S(t) + I(t) + R(t) + Q(t) $$

(5)

$$ I(0) = I_0 << N; S(0) = N - I_0; R(0) = Q(0) = 0 $$

(6)

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(t)</td>
<td>Number of susceptible hosts at time t</td>
</tr>
<tr>
<td>I(t)</td>
<td>Number of infectious hosts at time t</td>
</tr>
<tr>
<td>R(t)</td>
<td>Number of removed hosts from the infectious population at time t</td>
</tr>
<tr>
<td>Q(t)</td>
<td>Number of removed hosts from the susceptible population at time t</td>
</tr>
<tr>
<td>N</td>
<td>Total number of hosts under consideration $N = I(t) + R(t) + Q(t) + S(t)$</td>
</tr>
<tr>
<td>J(t)</td>
<td>Number of infected hosts at time t $J(t) = I(t) + R(t)$</td>
</tr>
<tr>
<td>$\beta(t)$</td>
<td>Infection rate at time t</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Infection rate parameter</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>removal rate from infectious hosts</td>
</tr>
<tr>
<td>$\mu$</td>
<td>removal rate from susceptible population</td>
</tr>
</tbody>
</table>

As another solution we have found that using sticky honeypots or NIPS would cause the infection rate to be decreased, this means we can use honeypots and or NIPSs to decrease $\eta$ in the model.

VI. SIMULATION RESULTS

In order to identify how prevention methods have effect on propagation of worms, we used two factor model in [10]. Authors picked the following values to simulate Code Red propagation.

$$ N = 10^6, I_0 = 1, \eta = 3, \gamma = 0.05, \mu = 0.06 \frac{N}{N}, B_0 = 0.8 \frac{N}{N} $$

We used the above parameters to compare how prevention methods described in section IV, have effect on number of infectious hosts and infected hosts over time. The solid line indicates Code Red worm propagation in all next figures.

A. Antivirus

Installing antivirus on network hosts would increase $\gamma$. Figure 2 and 3 shows how using antivirus may affect infectious and infected hosts respectively. As we expected in figure 2, the maximum number of infectious hosts and the life span of worm will be decreased.

The removal rate from infectious hosts or $\gamma$ describes how many of infectious hosts used for example an antivirus to get disinfected. This means having more antivirus protected hosts, increases the removal rate from infectious hosts.

By deploying patch management mechanism on network hosts, number of susceptible hosts will be decreased. This affects $\mu$ parameter in the model.
Figure 3 shows adding more antivirus protection would decrease total number of infected hosts at time $t$ which is $J(t)$.

### B. Patch Managements

Deploying patch management solution will decrease the total number of infected hosts as shown in the figure 4. Indeed it’s more effective than deploying antivirus as compared with figure 3.

![Fig. 4 effect of Patching on infected hosts](image)

Figure 5 shows the effect of patching on the infectious hosts. As we see it has slight effect on decreasing worm life span but it will decrease the maximum number of infectious hosts.

![Fig. 5 effect of Patching on infectious hosts](image)

### C. Sticky honeypots and Network Intrusion Preventions

Using sticky honeypots and NIPS causes infection rate to be decreased. Figure 6 shows the effect of honeypot on the infectious hosts. As it can be seen using honeypots may somewhat decrease the maximum number of infectious hosts and the worm life span. Also figure 7 depicts that using honeypots and NIPSs may decrease the total number of infected hosts. But it is not as salient as using Patch management.

Sticky honeypots can be used as a probe to notify that a worm outbreak is going to happen but they cannot reduce the number of infected host considerably.

![Fig. 6 effect of Sticky Honeypots on infectious hosts](image)

![Fig. 7 effect of Sticky Honeypots on infected hosts](image)

### VII. CONCLUSIONS

In this paper some worm propagation methods were described using strategies that fast spreading worms used to propagate themselves. By defining these methods one can countermeasure them to prevent worm propagation. In order to identify how prevention methods affect on propagation of the worms, we used the two-factor model. In our simulation we found that using the antivirus solution has a remarkable effect on infectious hosts compared to patch management and the sticky honeypot solutions, however patch management has a salient effect on decreasing the total number of infected hosts. Sticky Honeypots may decrease the total number of infectious hosts however they cannot decrease the total number of infected hosts effectively. For a better result in propagation containment, giving knowledge to novice users about the malwares and their dangerous role in network security is recommended, because removal rate from the infectious hosts by installing antivirus and increasing the patching rate by deploying patch management solutions both need human interventions.
REFERENCES


