The Game-Theoretic Model and Analysis of Patching-Based Worm Defense Strategy

Jia Liu
Computer Science Department
Oklahoma State University
jiliu@cs.okstate.edu

Abstract

In this paper, we analyzed a case in network security - worms events and corresponding patching defense strategy against worms, and then gave out a mathematical model by Game Theory. By this model, we found that there is a balance between the two parties of the worm counterwork game, which can be described by the two parameters of the proportion of patched hosts and the probability of successful attacks, （p^*, q^*）. Moreover, we proved that it's only when the punishment to attackers increases with the losing of users in the worm events that the utility of attackers is lower than their risk. This model also showed us an illuminating result that the best solution to worms is based on the corporation of users, security companies, the government, software manufacturers, and insurance companies to regulate the utility function of users and attackers. We end by discussing how to move the balance of the game to reduce the social losing.

Keywords: Worm Defense, Patching, Game Theory, Nash Equilibrium

1. Introduction

The gradually prevalent use of Internet makes worms a greater threat to the world. In 1988, Morris worm[1] infected 5 to 10 percent of all about 60,000 hosts then[6]. Code Red[3] and Nimda[5] infected hundreds of thousands hosts in 2001. On January 2003, SQL Slammer[13] swept 90 percent of all susceptible hosts in only ten minutes. Blaster in 2003 and Sasser in 2004 also cut down several million of Windows 2000/XP systems. Mass scanning packets result in global DDoS attacks and cause great damage, as shown in Fig.1.

Worm	Damage (in dollars)	Vulnerability
Morris	5%-10% of all 60000 hosts in 1998	BSD Unix Fingerd buffer overflow
Code Red	2.62 billion	IDA/IDQ ISAPI buffer overflow (MS01-33)
Nimda	640 million	MS00-78, MS01-20, MS01-21, Code Red
Slammer	1.25 billion	MS SQL Server buffer overflow (MS02-39)
Blaster	2 billion	MS RPC DCOM remote buffer overflow (MS03-26)

Fig.1 Damage and exploited vulnerabilities of several most devastating worms.
(Some data come from CERT/CC)

Recent research on worm defense focused mainly on techniques. At first, Weaver[2] presented a classification method of worms, based on which, Zou[7] discussed the performance of propagation of different kinds of worms. Moore[4] presented the concept of network telescope, observing security incidents on the Internet by using a small part of IP space. Zou[20] presented a non-threshold based worm detection method to detect the exponential growth trend of a worm's propagation. Zou[19] also presented a feedback dynamic quarantine system for automatic worm attacks mitigation. For worm defense in enterprise networks, Staniford[14] presented CounterMalice. Another worm defense model is cooperative response, presented by Nojiri[15], in which compromised sites warn friendly hosts of the worm attack, helping other hosts to block the worm. Williamson[16] studied worm containment by constraining the outgoing scan rate from infected hosts, which can obviously decrease the worm's propagation speed. Then Zou presented a Firewall Network System, which uses firewalls to divide an enterprise network into many isolated sub-networks, and to block all service requests sent to internal IP addresses of an enterprise network. From another angle, Nicol[17] studied how to use "good" worms in worm defense.

The research on worm defense techniques is still in progress, and more experiments are needed to evaluate the effect of the methods. The key point of security, however, is not technology, but how to use it properly, in other words, the user's attitude, motivation and behavior towards network attacks. Since Microsoft Security Bulletin is able to publish vulnerabilities with security updates in time, worms exploiting these vulnerabilities shouldn't have propagated drastically if most systems were patched with the security updates. But they did, as we know. All the latest five severe worm events took place after Microsoft had published the vulnerabilities and provided security patches. Since the simple strategy failed, it's necessary to find out the cause. What factors resulted in the situation? How to avoid similar things occur in the costly complicated systems?

In this paper, we explained dynamics in the worm events, and predicted the direction of worm defense in the future. The term worm in this paper is considered as the worms propagate through system vulnerabilities, rather than email viruses. For the convenience of expression, we call worm makers attackers.

2. Related Work

Recently, some famous security experts realized that the network security isn't a problem of technology, but a problem of business, since risk management is the core section in security[11]. First held in 2002, the Workshop on Economics and Information Security (WEIS) built a platform for people to discuss the main factors in information security, and to choose appropriate mathematics tools to research. Vila[10] introduced the Game Theory into security, discussed the cyclic instability in the number of websites that sell consumer information. Danezis[12] put forward a game theory model for censorship resistance. Garg[8] used a game-theoretic approach to analyze the congestion in communication networks, and proposed the Diminishing Weight Schedulers (DWS) that punish misbehaving users and reward congestion avoiding well behaved users, to reduce the incentive of users to increase traffic unfairly. The game theory has been gradually a effective tool for network security.

3. Worm Defense by Security Updates

Let Malice be one of the worms makers. There're usually two ways for Malice to get vulnerability to make a worm. The first is to search it in the system or buy it from some hackers, and the other way is to wait for the publication of new vulnerability by software producer, locating them by comparing original files and patching files[9]. The cost of the former is much greater than the latter. In Fig.2 we found that worms always appear after the publication of security updates, which means that attackers always choose to analyze patching files, therefore patching in time is an effective defense method against worms.

Microsoft's Windows 2000/XP/2003 support Automatic Updates Service, which could downloads security updates as soon as they're published. Since worms come later than patch publishing, it's the user's fault to miss updates and expose the system under the attack of worms. The game theory could give us some insight explanation.

Worm's Name	Dates of Publishing Security Updates	Dates of Appearance	Interval Days
Code-Red	2001.6.18	2001.7.12	24
CodeRed II	2001.6.18	2001.8.4	47
Nimda	2001.8.4 Backdoor of CodeRed II	2001.9.18	45
Slammer	2002.7.10	2003.1.25	199
Blaster	2003.7.16	2003.8.11	26
Sasser	2004.4.13	2004.4.30	17

Fig.2 Interval days between the date of publishing vulnerability exploited by a worm and the date of the worm's appearance.

2.1 The Game Theory Model of Worm Defense by Security Updates

In order to describe this model formally, we suppose that users fall into two categories: *Patching* users that download security updates and install them, and users that *Don't* patch for systems. Attackers could select to release *Worm* or to keep the *Peace*.

We assume that once a user decides to install the updates, he could always install it correctly, and be free from infecting the corresponding worm forever. Patching system is not free, since he has to spend some time, use the network, or hire somebody to maintain the system. And it's always possible for the unpatched systems to be infected with worms. Once a system is infected, the user suffers some loss immediately.

Allow for the variables representing:

P = the cost for the user to patch a system

C = the damage by the worm

α = the probability of infection in case the system isn't patched

The user's payoff matrix is

$$\begin{pmatrix} & Worm & Peace \\ Patches & -P & -P \\ Doesn't & -\alpha C & 0 \end{pmatrix} \quad (1)$$

The best situation for a user is (*Doesn't*, *Peace*), when he could get the highest payoff.

Attackers decide the appearance of worms. In this paper, we assume that the worm maker is also the worm releaser. Once programming a worm, the attacker always releases it. Since worm attackers are relatively rare, we regard them as a whole entity, which decides whether to produce a worm. Coding and releasing worm is not easy, since it cost time and the attacker has to undertake some risk of being arrested. Therefore, attackers must look forward to some benefits from the attack, such as reputation, money, power, or exhibiting their techniques to obtain some mental satisfaction.

Similarly, we let

W = the cost for making the worm

S = the benefit for Malice to release the worm

D = the punishment for Malice when arrested

β = the probability of Malice being arrested

The attacker's payoff matrix is

$$\begin{pmatrix} & Worm & Peace \\ Patches & -W-\beta D & 0 \\ Doesn't & -W-\beta D+S & 0 \end{pmatrix} \quad (2)$$

When $S \le W + \beta D$, the attacker won't release his worm, since *Peace* is a Pareto-dominant strategy for him. It's the same to him when $S = W + \beta D$. In case of $S > W + \beta D$, the attacker expects the user not to patch, for he could obtain a higher payoff then.

Obviously, the benefit of the user conflicts with the benefit of the attacker. To study the influence between them, we combine the two payoff matrices and get

$$\begin{pmatrix} & Worm & Peace \\ Patches & -P,\ -W-\beta D & -P,\ 0 \\ Doesn't & -\alpha C,\ -W-\beta D+S & 0,\ 0 \end{pmatrix} \quad (3)$$

Suppose the user has motivation to install patch, which means $P < \alpha C$, and the attacker has motivation to release the worm, which means $(1-\beta)S > (W + \beta D)$. There's no pure strategy Nash equilibrium for this game. For each strategy combination, there's always a transfer to give a higher payoff to a player. Nash proved that there's always mixed strategy equilibrium in any non-corporation game.

3.2 The Mixed Strategy Nash Equilibrium of Worm Defense

Let *p* represent the probability of making worms. We can then compute the relative utility of patching systems instead of leaving them unpatched.

$$
\begin{aligned}
U(Patches) &= p(-P) + (1-p)(-P) \\
&= -P \\
U(Doesn't) &= p(-\alpha C) + (1-p)\cdot 0 \\
&= -p\alpha C
\end{aligned}
$$

$$
\begin{aligned}
U(Patches) - U(Doesn't) &= -P - (-p\alpha C) \\
&= p\alpha C - P
\end{aligned} \tag{4}
$$

When p approaches 1, which means for each available vulnerability there always appear some worms, if $\alpha C > P$, users incline to install security updates. In this case, users' preference of patching systems increases with α. When p approaches 0, and no worm will emerge, users' relative utility is $-P < 0$, no one inclines to patch systems.

We can do similar calculations on the relative utility of attackers. By labeling q as the portion of the users who install updates, we can check the relative benefits of attackers:

$$
\begin{aligned}
U(Worm) &= q(-W - \beta D) + (1-q)(-W - \beta D + S) \\
&= (1-q)S - W - \beta D \\
U(Peace) &= q\cdot 0 + (1-q)\cdot 0 \\
&= 0
\end{aligned}
$$

$$
\begin{aligned}
U(Worm) - U(Peace) &= (1-q)S - W - \beta D - 0 \\
&= (1-q)S - W - \beta D
\end{aligned} \tag{5}
$$

When q approaches 1, and all users patch systems, the relative utility of attackers is $-W - \beta D$, therefore attackers wouldn't like to make and release worms. When q approaches 0, and no user installs updates, if $S > W + \beta D$, attacker prefers to release worms. As long as one of W、D、β increases, his incentive declines.

This analysis suggests a dynamic instable procedure on the proportion of the strategies of the two sides.

a. When no worm is released, no user will install security updates.
b. When no user installs updates, attacker will release worms if any possible.
c. When worms always occur, most users will patch their systems.
d. When most users patch systems, worms will not prevail, and attacker's work will be vain.
e. Return to step a, and repeat this procedure.

This cycle continues with various values of p and q as time moves on. When the relative utilities of both approach 0, we get stable middle ground values of (p*, q*).

$$p^* = P / \alpha C \tag{6}$$

$$q^* = 1 - (W + \beta D) / S \tag{7}$$

If patching and non-patching users reach the balance, then attackers get equal benefits from releasing or not releasing worms. Similarly, if the possibility of releasing worms is certain, then

user's cost to patch systems equals the loss if the systems are infected. This is the only Nash equilibrium in the payoff matrix 3. However, this equilibrium is not most stable, since the inter-influence of the two parties is a slow process, and no motility could draw them back to the balance point. To make the balance more stable, some constraints have to be attached. Since the punishment to a arrested attacker should increase with the damage he caused, expressing formally as $D = f(C)$, with $f' \geq 0, f'' \leq 0$. For the convenience of discussing, let $D = C^{1/k}, (k \geq 1)$, combining Eq 6 and Eq 7 we get

$$C = \frac{P}{\alpha p^*} = \left[\frac{(1-q^*)S - W}{\beta} \right]^k = D^k \tag{8}$$

$$p^* = \frac{P}{\alpha} \cdot \left[\frac{\beta}{(1-q^*)S - W} \right]^k \tag{9}$$

Above suggests that p^* and q^* restrict each other, one increases with another. When C increases, q^* decreases, that is, if the attacker caused severer damage, then he'll suffer more punishment when arrested, which add risks for attacker and therefore reduce the probability of worm event. When C remains stable, as S increases, p^* increase, which means the more benefits the attacker obtain, the higher probability of the appearance of worms.

The laws should play a deterring role in network security, thereby the punishment should be above the utility of attackers. That is

$$(1-q^*)S - W - \beta D < D \tag{10}$$

$$p^* < \frac{P}{\alpha} \cdot \left[\frac{(1+\beta)}{(1-q^*)S - W} \right]^k \tag{11}$$

Inequality 11 comes from Equality 9 and Inequality 10. Hence we get

Theorem 1 If $D = f(C)$, $f' > 0$, then $U(Worm) < D$ is true.

Similarly, we also get

Theorem 2 If $D = f(C)$, $f' < 0$, then $U(Worm) > D$ is true.

4. Discussion

This model gives insight into the worm defense strategy by security updates. Eq 6 indicates that the probability of worm's appearance varies firstly with user's parameter in the balance state, and then influences the attacker reversely by punishment, at last moves the proportion of patching q^*, the user's parameter. Similarly, Eq 7 tells that attacker's parameter firstly influences q^* and then p^*. One of the principles for the government to make laws is to measure the punishment for network crimes, which could make the balance more stable. By current state of anti-crime techniques, it's very hard to get enough evidence to arrest worm attackers, which means p approaches 1. Therefore, to reach the equilibrium, q should also approaches 1. However, current q is far less than 1, and the situation remains far from the equilibrium. By increasing the punishment

and improving the security technology, attackers incline not to code and release worms. Theorem 1 indicates that when punishment increases with user's loss, attacker's utility could be always less than punishment, while Theorem 2 means that when punishment decreases with user's loss, attacker's utility will cover the punishment. Since the latter situation is obviously impractical, it couldn't take place unless the attacker is reasonless. When no definite relation exists between punishment and loss, attacker's utility varies uncertainly with the punishment.

Since worms usually appear after security updates, software producers must have some influence on attackers. Therefore, producers should hide the details of vulnerability and encrypt the patching files, which could increase the cost of reverse engineering and win over more time for users to finish updating. Besides, security companies should develop new techniques to get more possibility to catch attackers, and to distribute updates quickly and securely. Furthermore, insurance companies are expected to reduce the loss of users by providing social insurance. And the government has the responsibility to regulate the equilibrium point by making laws, and to remind users of updating systems while warning potential attackers.

5. Conclusion

We analyzed the current state of worm defense by security updates. Although attackers have two choices, to seek for vulnerability by themselves or to wait until someone publicizes some vulnerability, most worm attackers choose the latter in fact. Usually attackers and users each select strategies with the highest payoff for him, but their beneficial focuses conflict with each other, and their inter-influence determines the trend of a worm event. Therefore it's necessary to calculate them together.

We create the game theoretic model to predict the state variety of both sides in this game, which yielded a single equilibrium point, that is

$$(p^*, q^*) = \left(\frac{P}{\alpha C}, 1 - \frac{W + \beta D}{S} \right)$$

In fact, the states oscillate around the equilibrium point. This is because:

(1) There's some delay in the interaction of the two sides, and one side's decision variety would not influence the other timely.

(2) Every group in worm games has its own utility function. For example, security companies could obtain benefits from worm events. They may lead users from one state to another, which could maximum their profits.

(3) A lot of variables could influence the equilibrium point, and they varies from time to time. For instance, different evaluation methods give diverse values for user's loss; systems in different environment have changing possibility to be infected with worms; reinstalling systems would make a updated system susceptible again; different attackers undertake different risks, costs, and gain various benefits.

In summary, this model has the effect to predict the average trend of state transformation, and equilibrium point could help us to evaluate the social loss at that point, then regulate the balance by laws and techniques, and eventually reduce resource costs.

References

[1] D. Seeley. A tour of the Worm. In Proceedings of the Winter Usenix Conference, San Diego, CA, 1989.

[2] N. Weaver, V. Paxson, S. Staniford, R. Cunningham. A Taxonomy of Computer Worms . WORM'03, Washington, DC, USA, 2003.

[3] D. Moore, C. Shannon, and J. Brown. Code-Red: a case study on the spread and victims of an Internet Worm. In Proc. ACM/USENIX Internet Measurement Workshop, France, November, 2002.

[4] D. Moore. Network Telescopes: Observing Small or Distant Security Events. In 11th USENIX Security Symposium, 2002.

[5] CAIDA. Dynamic Graphs of the Nimda worm. http://www.caida.org/dynamic/analysis/security/nimda/index.html

[6] VENUS INFO TECH INC. Corporate. http://www.venustech.com.cn/tech/focus/20040514/2207.htm

[7] C. C. Zou, D. Towsley, W. Gong. On the Performance of Internet Worm Scanning Strategies.

[8] R. Garg, A. Kamra, V. Khurana. A Game-Theoretic Approach Towards Congestion Control in Communication Networks. ACM SIGCOMM Computer Communications Review, 2002.

[9] M. A. Bashar, G. Krishnan, M. G. Knhn. Low-Threat Security Patches and Tools. IEEE, 1997.

[10] T. Vila, R. Greenstadt, D. Molnar. Why We Can't Be Bothered to Read Privacy Policies — Models of Privacy Economics as a Lemons Market. WEIS03, University of Maryland, College Park, MD, 2003.

[11] B. Schneier. Computer Security: It's the Economics, Stupid. WEIS'03, University of Maryland, College Park, MD, USA, 2003.

[12] G. Danezis, R. Anderson. The Economics of Censorship Resistance. WEIS'04, University of Minnesota, USA, 2004.

[13] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, N. Weaver. Inside the Slammer Worm.

[14] S. Staniford. Containment of Scanning Worms in Enterprise Networks. To appear in Journal of Computer Security, 2003.

[15] D. Nojiri, J. Rowe, K. Levitt. Cooperative Response Strategies for Large Scale Attack Mitigation. In 3rd DARPA Information Survivability Conference and Exhibition, Apr. 22-24, Washington DC, 2003.

[16] M. M. Williamson. Throttling Viruses: Restricting Propagation to Defeat Mobile Malicious Code. In 19th Annual Computer Security Applications Conference, San Diego, 2002.

[17] D. M. Nicol. Models of Internet Worm Defense. In IMA Workshop 4: Measurement, Modeling and Analysis of the Internet, 2004.

[18] C. C. Zou, D. Towsley, W. Gong. A Firewall Network System for Worm Defense in Enterprise Networks. 2004.

[19] C. C. Zou, W. Gong, D. Towsley. Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense. In ACM CCS Workshop on Rapid Malcode（WORM'03）, Oct.27, Washington DC, 2003.

[20] C. C. Zou, L. Gao, W. Gong, D. Towsley. Monitoring and Early Warning for Internet Worms. In 10th ACM Conference on Computer and Communication Security（CCS'03），Oct.27-31, Washington DC, 2003.