Published in the Proceeding (533) of the
International Conference of Communications, Internet, and Information 2006 (ICCII06)
US Virgin Islands, USA, December 2006, pp 533-809.

MODELING AND ANALAYSIS OF WORM ATTACKS WITH PREDATOR AND PATCHING INTERPLAY

Zakiya M. Tamimi[1] & Javed I. Khan[2]
[1]Arab American University-Jenin, Faculty of Information Technology
Jenin, Palestine, P.O. 240
[2]Media Communications and Networking Research Laboratory
Department of Math & Computer Science, Kent State University
233 MSB, Kent, OH 44242
ztamimi|javed@kent.edu

ABSTRACT

Internet is increasingly seeing the emergence of very fast propagating worms capable of infecting significant part of the Internet in short few hours. Predators – a type of good-will self-replicating codes has been proposed as a class of anti-worm defense which can potentially counter formidable speed of these newer worms. In this paper we expand our pervious study of predator's effectiveness by considering more complex situations. We model the predator behavior in presence of system patching, which are operating system hot fixes. In another scenario we model the predator behavior against a smart malicious worm (called Backdoor Blocker Worm or BBW) that blocks the backdoor after penetrating into a machine. In all cases, we use mathematical modeling and numerical solution to drive results.

KEY WORDS: Internet worm, system patch, backdoor, and modeling.

1. Introduction

Since the release of Code-Red in 2001 there have been many very fast worms that span the Internet in few hours or even minutes. In response, some people have released anti-worm worms such as: Code-Green and Welchia to fight Code-Red and MS-Blast, respectively. Relaying on typical antivirus software and downloadable updates proved to be inadequate and thus the idea of the predator worm was born. A predator worm, also called killer worm or anti-worm worm, is a good-will worm that battles a malicious worm on our behalf [1], [2], [6]. The controversial idea of releasing a predator worm was discussed from efficiency [6], effectiveness [1], and validity [4] point view. Unless otherwise is stated, throughout this paper, we refer to a malicious worm as worm and to a predator worm as predator.

Predator worms are like other worms that propagate over a network through some vulnerability (or security hole). Predators find their "victims" by either actively seeking them (active-scanning predators) or by waiting for the infectious machine to scan "by mistake" a machine with a predator worm, called (passive-scanning predators), e.g. Code-Green [1]. Once inside a machine, a predator will "kill" the malicious worm. Some predators may infect clean machines as well as infectious ones; however such vulnerability-driven predators, e.g. the Welchia worm, induce inefficiency. On the other hand, predators that only infect infected machines are called infection-driven [1]. While protecting a machine against some security hole is the job of a predator, it won't be able to propagate if such security hole doesn't exist upfront. This brings to attention the following question: how effective will a predator be if system patching are being applied? System patch is binary code made available by some system software provider that will fix some security hole in that software. Usually, such system patches secure an existing vulnerability, which otherwise can be used by a worm to penetrate a machine. Another issue to consider is a Backdoor Blocker Worm (BBW), which is a malicious worm that closes the backdoor after penetrating into machine to protect itself against a predator worm. Although all BBW instances have such the blocking capability incorporated in their code, not all instances will succeed in closing the backdoor. However, both successful and unsuccessful BBW instants can breed both successful and unsuccessful BBW instances.

The goal of this research is to investigate the effectiveness of a predator worm when some the security hole that to be used by the predator is closed

Published in the Proceeding (533) of the
International Conference of Communications, Internet, and Information 2006 (ICCII06)
US Virgin Islands, USA, December 2006, pp 533-809.

by some other party. We model two interesting scenarios of the interaction between a worm and a predator. First, we model the fight of a predator worm against a Backdoor Blocker Worm (or BBW). The second scenario catches on the fight of a predator worm against a worm in the presence of system patching. Following the step of others [1], [5] and [6], we use mathematical modeling and incorporate multiple factors in order to predict the outcome of possible predator's release.

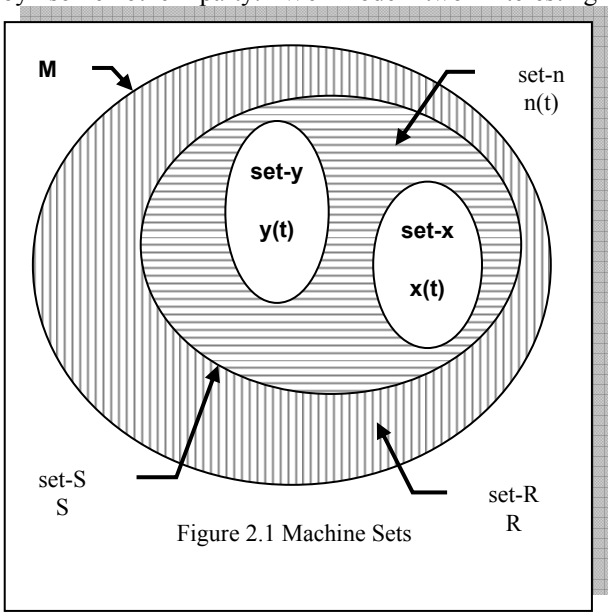


Figure 2.1 Machine Sets

Following this introduction, in section 2 a background of the predator and worm interaction model is presented. The fight of a predator against a worm in the presence of system patching is presented in section 3. In section 4 we present a model the use of predator to fight against BBW. Finally we conclude in section 5.

2. Propagation Model

We assume that only two worms are propagating on a network: a malicious worm (called worm-x) and a predator worm (called worm-y). Both the worm and its predator use the same vulnerability to penetrate into a machine. The network has fixed number of machines M that can be classified according to their status into:

- **Set-R**: set of immune or removed machines which cannot be infected by a worm or a predator worm.
- **Set-S**: set of machines that are susceptible to infection by a worm or a predator. This set consists of there sub-sets: set-n, set-x, and set-y.
- **Set-n**: sub-set of susceptible machines that are have no infection (or simply clean).
- **Set-x**: sub-set of susceptible machines that are infected by a worm.
- **Set-y**: sub-set of susceptible machines that are infected by a predator worm.

Figure 2.1, see [1], shows the sets and their cardinalities.

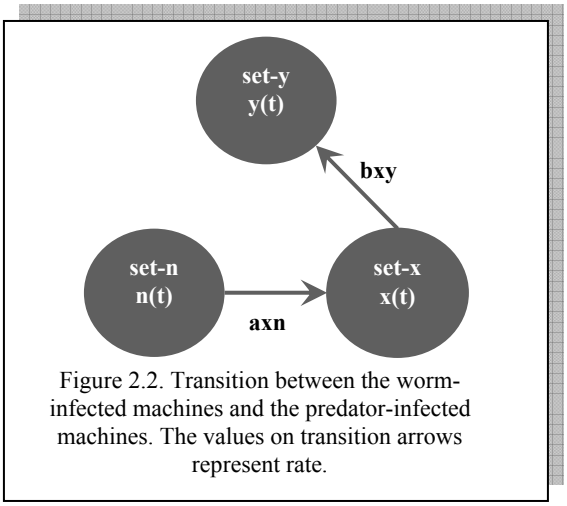


Figure 2.2. Transition between the worm-infected machines and the predator-infected machines. The values on transition arrows represent rate.

As discussed in our previous work [1] a worm and a predator interact according to equations 2.1 through 2.4. The transition rate between set-x, set-y, and set-n is shown in figure 2.2. Clean machines become infected and move to set-x at rate proportional to number of clean machines and number of worm-x infectious machines. However, worm-x infectious machines are taken over by worm-y and are cleaned and immunized at rate proportional to the number of worm-x and worm-y infectious machines.

$$\frac{dx}{dt} = axn - bxy \tag{2.1}$$

$$\frac{dy}{dt} = bxy \tag{2.2}$$

$$\frac{dn}{dt} = -axn \tag{2.3}$$

$$x(0) = x_0, y(0) = y_0, n(0) = n_0 \tag{2.4}$$

a and b are positive constants and their values are dependent on worms or predator scanning rate and network size. Assuming the scanning rate of a worm is r the value of a is given in equation 2.5 . Similarly the value of b is given in equation 2.6, assuming the scanning rate of the predator is v. In this paper we assume that a predator is active-scanning infection-

Published in the Proceeding (533) of the
International Conference of Communications, Internet, and Information 2006 (ICCII06)
US Virgin Islands, USA, December 2006, pp 533-809.

driven worm. In other words, the predator does scanning on its own to find a worm infected machine and the predator does infect only infected machines.

$$a = \frac{r}{M} \tag{2.5}$$

$$b = \frac{v + r}{M} \tag{2.6}$$

Figure 2.3 shows that initially worm-x increase exponentially as it would without the existence of worm-y. Worm-y will increase proportional to worm-x increase. The increase in worm-y population, however, will result in decreasing worm-x population. Curve $x(t)$ stops increasing, hits a maximum, and starts declining. Curve $y(t)$ continues to increase until it uses up all available worm-x members, where it hits its maximum and freeze thereafter. The system reaches steady state when both infection rates are zero. This occurs when all worm-x infected machines are re-infected by worm-y.

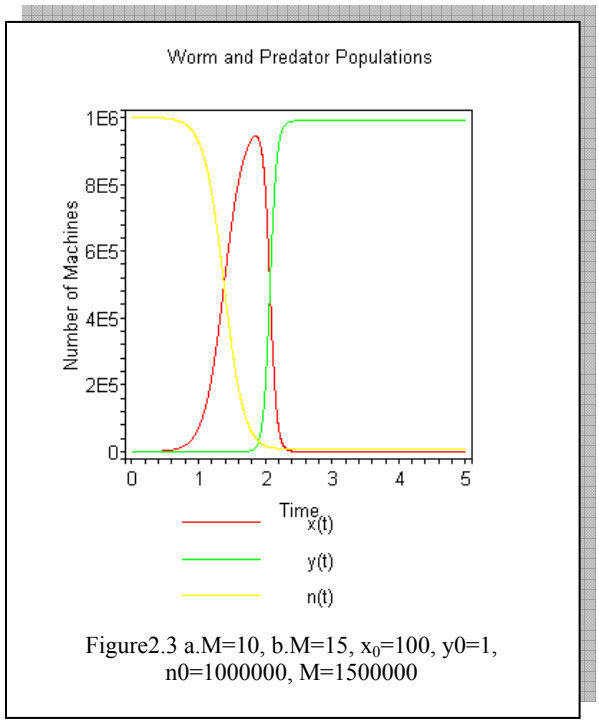


Figure2.3 a.M=10, b.M=15, x_0=100, y0=1, n0=1000000, M=1500000

3. Predator and System Patching

A system patch is some binary code provided by software provider to fix some security hole in that software. When a system patch is applied to some clean machine it will immunize it against possible worm infection. In addition, some system patches can remove an existing worm infection from a machine as it is patched. Usually a software provider supply such system patches available as Internet download or as CD. However, users may be unable to download such system patches if their machines are infected; due worm's network activity that limits available bandwidth or due to Denial of Service attack on the download website. Another way to patch a machine can be done by manually fixing the security hole, such as terminating the vulnerable on the machine. This method doesn't however remove a worm infection if existed.

Contrary to previous sections, we assume the number of removed machines on the network is variable $R(t)$. The reason for this is simple, a susceptible machine in set-S that is patched will become removed; joins set-R. We assume the rate of patching machines in set-S is constant k, regardless of machine status or patching method. We also make the following assumptions:

- When patching a clean machine is any mean it become immune (or removed).
- k_{cd} of machines in set-S are patched successfully using system patch CD. Those, regardless of their status will be removed as a result.
- k_{net} of users try to patch their machines by downloading system patch form the Internet. This method succeeds if the susceptible machine is clean and fails otherwise.
- k_{man} of machines in set-S are patched manually. If a machine is in set-n then after this manual patching it will be move to set-R. If the machine in set-x or set-y, it will still be infectious after such patch application, however immune to other infections. This case is interesting since a patched infectious machine will still be able to scan other machines.
- Of course the sum of k_{cd}, k_{net}, and k_{man} is k, as in equation 3.1

$$k = k_{cd} + k_{net} + k_{man} \tag{3.1}$$

$$\frac{dx}{dt} = axn - k_{cd}x - bxy(1 - k_{man}) \tag{3.2}$$

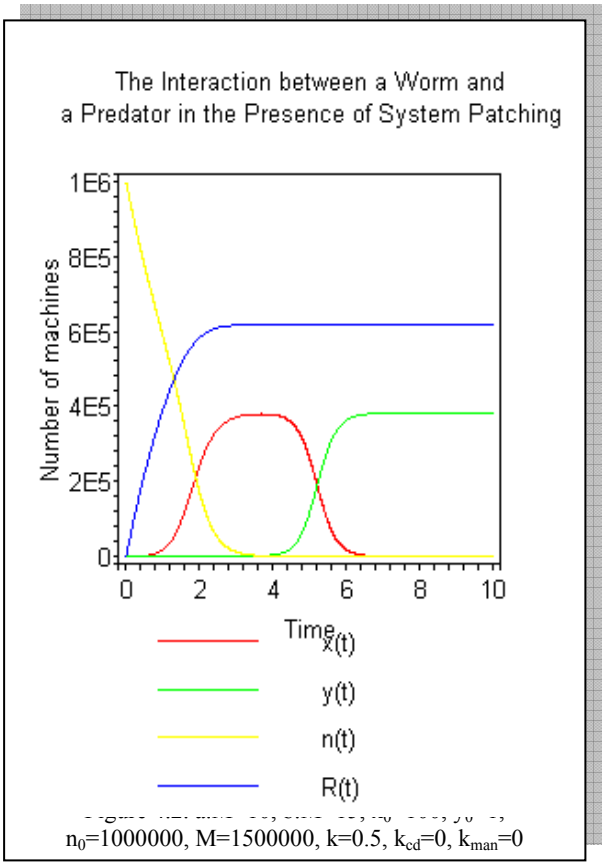
$$\frac{dy}{dt} = -k_{cd}y + bxy(1 - k_{man}) \tag{3.3}$$

$$\frac{dn}{dt} = -axn - kn \tag{3.4}$$

$$\frac{dR}{dt} = kn + k_{cd}x + k_{cd}y \tag{3.5}$$

Published in the Proceeding (533) of the
International Conference of Communications, Internet, and Information 2006 (ICCII06)
US Virgin Islands, USA, December 2006, pp 533-809.

$$x(0) = x_0, y_0(0) = y_0$$
$$R(0) = R_0, n(0) = n_0 \qquad (3.6)$$



n_0=1000000, M=1500000, k=0.5, k_{cd}=0, k_{man}=0

Figure 3.1 describes the machines transitions between the different sets as a result of interaction between the worm and the predator in the presence of system patch. Set-n will lose members to set-R proportional to the number of clean machines and the value of k. Both set-x and set-y will have k_{cd} of their population lost to the removed machines. Worm-x instances that are patched manually can infect more clean machines from set-n. However, this portion of worm-x population, which is manually patched, cannot be captured by predator instances, which explains the factor $(1-k_{man})$ in the transition rate between set-x and set-y. The model of this scenario is given in equations 3.2 through 3.6.

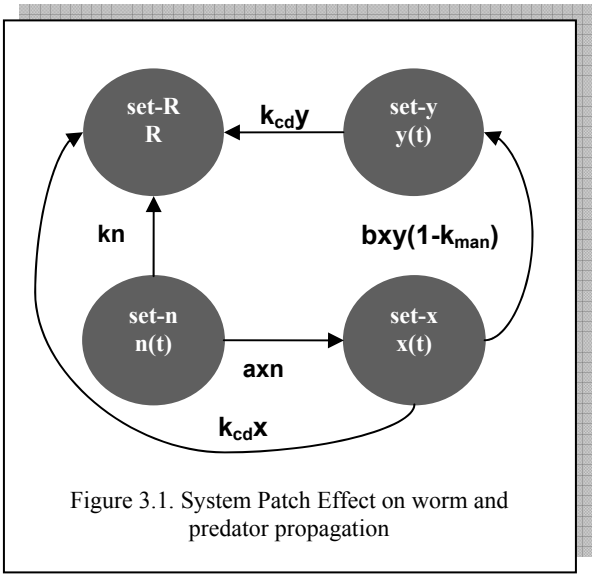


Figure 3.1. System Patch Effect on worm and predator propagation

In figure 3.2 we study the effect of k separately, thus both k_{cd} and k_{man} are zeros. It is obvious that system patching will limit the growth of both the worm and the predator populations. The $R(t)$ curve grows steadily the curve of $n(t)$ declines leaving limited number of clean machines to be infected by worm-x. In our case we assume only patching of clean machines, and thus the curve of $R(t)$ will become constant eventually. However the population of worm-x is consumed by the predator population. The maximum of the predator population is equal to the maximum worm population.

In figure 3.3 we turn to the study of the effect of k_{cd} and leaving k_{man} as zero. The figure shows the effect the system patch in lowering the maximum population of both the worm and the predator. Although the patching has a negative effect on the predator population it does contaminate the worm population even faster. The maximum value of curve $R(t)$ in this case is the environment capacity n_0.

Published in the Proceeding (533) of the
International Conference of Communications, Internet, and Information 2006 (ICCII06)
US Virgin Islands, USA, December 2006, pp 533-809.

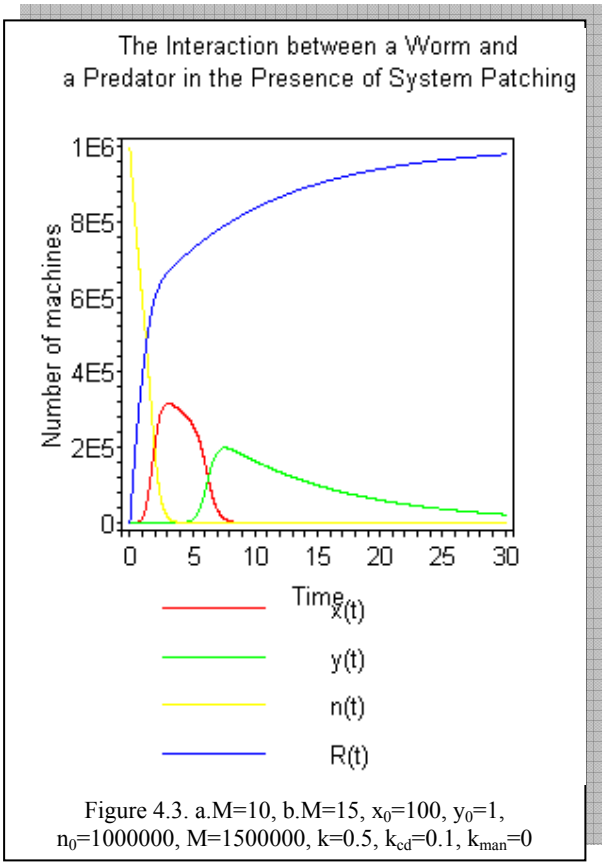


Figure 4.3. a.M=10, b.M=15, x_0=100, y_0=1, n_0=1000000, M=1500000, k=0.5, k_{cd}=0.1, k_{man}=0

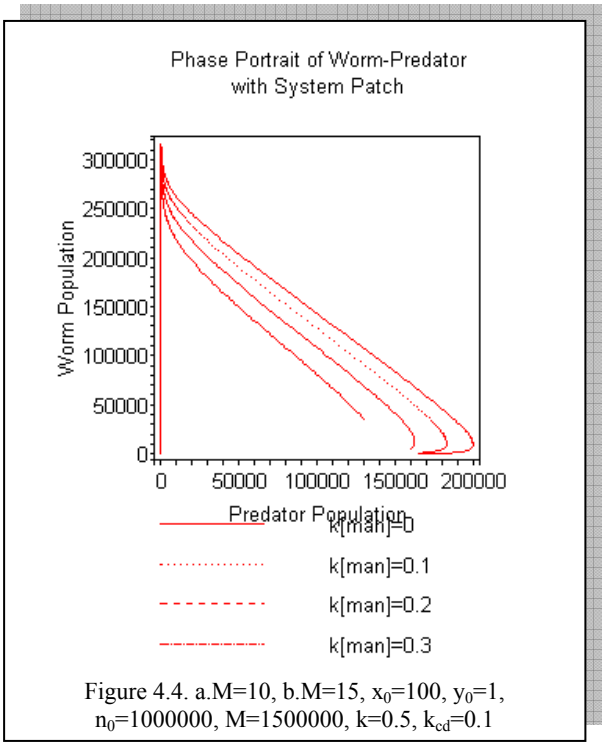


Figure 4.4. a.M=10, b.M=15, x_0=100, y_0=1, n_0=1000000, M=1500000, k=0.5, k_{cd}=0.1

In figure 3.4 the value of k_{man} is varied while keeping the values of k and k_{cd} fixed. It can be noticed the larger the value k_{man} the longer it takes the predator to take out the worm population. In one of the curves where $k_{man} = 0.3$ the predator wasn't able to kill all worm's instances, and thus the equilibrium point not zero. The reason why manual patching have such negative results is because it slows the spread of the predator by factor $(1-k_{man})$ which the fraction of worm instances that are immunized by manual patching.

4. Backdoor Blocker Worms

BBW is a worm that tries to protect itself from a predator by closing the backdoor through which it has penetrated the system. Although all BBW instances have such capability incorporated in their code, not all instances will succeed in closing the backdoor. This is due to configuration of the local victim machine or network; e.g. administrator account privileges might be needed. We don't consider a scenario where the predator is backdoor blocker predator; because predators don't need to protect themselves. Actually, some predators depend on a worm to step into the "trap" by mistakenly scanning a machine where a predator worm resides.

We classify BBW instances into:

- **set-x_p**: set of BBW instances that succeed to close the backdoor successfully. The set cardinality is $x_p(t)$
- **set-x_n**: set of BBW instances that fail to close the backdoor. The set cardinality is $x_n(t)$.

It's important to note that instances of BBW in set-x_p can breed both successful and unsuccessful BBW instances. Similarly instances in set-x_n can breed both successful and unsuccessful instants. Of course, set-x_p and set-x_n are complements of each others, see equation 4.1, and the ratios of $x_p(t)$ and $x_n(t)$ to $x(t)$ are p and q, respectively.

$$x(t) = x_p(t) + x_n(t) \quad (4.1)$$

Figure 4.1 describe the interplay between BBW and predator. The clean machines in set-n lose members to both set-x_p and set-x_n, at rates proportional to their ratios p and q, respectively. On the other hand, only set-x_n would loss members to predators (set-y).

Published in the Proceeding (533) of the
International Conference of Communications, Internet, and Information 2006 (ICCII06)
US Virgin Islands, USA, December 2006, pp 533-809.

Equations 4.2 through 4.6 describe the dynamics of the system.

$$\frac{dx_n}{dt} = qan(x_n + x_p) - bx_n y \tag{4.2}$$

$$\frac{dx_p}{dt} = pan(x_n + x_p) \tag{4.3}$$

$$\frac{dy}{dt} = bx_n y \tag{4.4}$$

$$\frac{dn}{dt} = -an(x_n + x_p) \tag{4.5}$$

$$x_n(0) = x_{n0}, x_p(0) = x_{p0}$$
$$y(0) = y_0, n(0) = n_0 \tag{4.6}$$

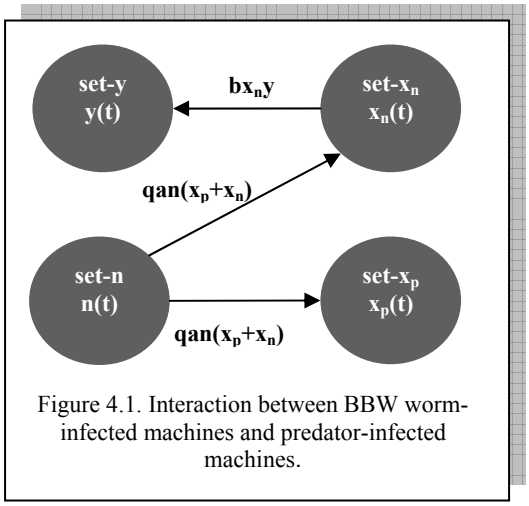


Figure 4.1. Interaction between BBW worm-infected machines and predator-infected machines.

This scenario is an example of a mutant worm, where a group of the worm population gains some extra feature that makes them more resilient. In this case, the featured members can close the backdoor and thus resist the predator worm.

As figure 4.2 shows the worm's population will grow exponentially. Once the predator population starts accelerating, the worm population will hit its maximum and start declining. In the figure, the maximum worm population is equal to the initial number of clean machines (also called environment capacity). As the predator population continues to increase the worm population will continue to decrease. However, a portion of the worm population $x_p(t)$ has instances that successfully blocked the backdoor of their host, and thus are immune to predator attack. That's why the curve of the worm will stop declining and become constant at value $p.\max(x(t))$ and that portion of the population will survive and persist. The predator population, on the other hand, will infect only the unsuccessful BBW instances. Thus the maximum predator's population is $q.\max(x(t))$ that is the complement of the worm maximum population.

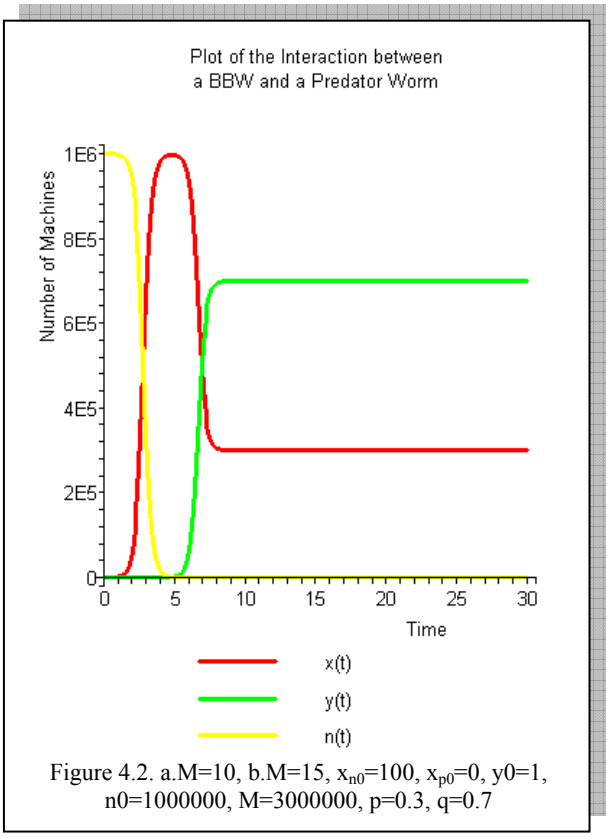


Figure 4.2. a.M=10, b.M=15, x_{n0}=100, x_{p0}=0, y0=1, n0=1000000, M=3000000, p=0.3, q=0.7

Published in the Proceeding (533) of the
International Conference of Communications, Internet, and Information 2006 (ICCII06)
US Virgin Islands, USA, December 2006, pp 533-809.

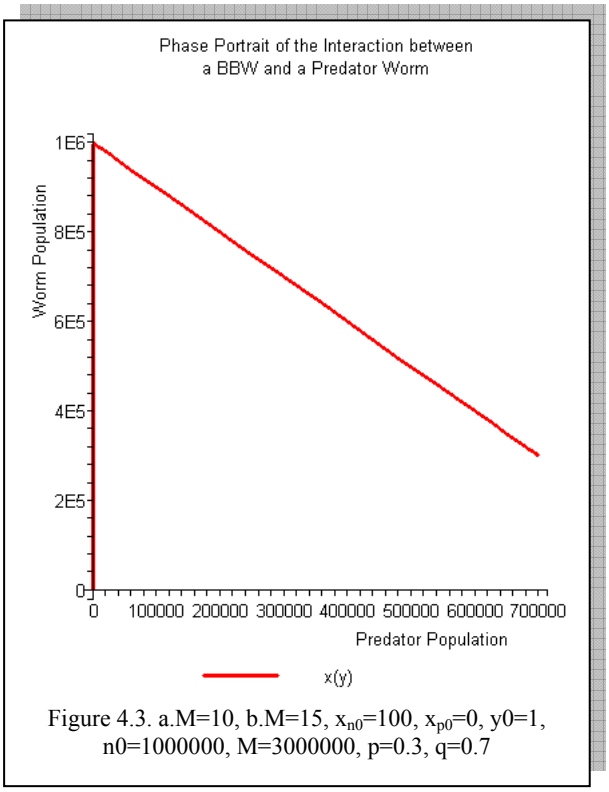


Figure 4.3. a.M=10, b.M=15, x_{n0}=100, x_{p0}=0, y0=1, n0=1000000, M=3000000, p=0.3, q=0.7

In figure 4.3 a plot of the curve $x(y)$ is shown. The equilibrium point of the system is when $\frac{dx}{dt} = \frac{dy}{dt} = \frac{dn}{dt} = 0$. In the figure, this equilibrium point is $(700000,300000)$, meaning eventually when the system becomes stable there will be left 300,000 worm instances. We can see that the success of the predator is dependent on p or in other words, how many BBW are successful.

5. CONCLUSIONS

In this paper we have presented two models of fighting a malicious worm using a predator worm. We investigated the effect of fixing the security hole used by the predator on its effectiveness. In one scenario, we have modeled the interplay between a predator and a backdoor blocker worm (BBW). We concluded that a BBW will hinder the predator functionality. In another scenario, we modeled the worm and predator interaction in the presence of system patching. In this case, we found out that supporting a predator worm with system patching can be effective as long as such patching is applied correctly and manual patching is avoided. However these conclusions are not to be taken in a rigid way. We consider the presented models as means to calculate and predict the outcome of releasing a predator.

References

[1] Z. Tamimi, J. I. Khan, Model-Based Analysis of Two Fighting Worms. *Proc. of ICCCE'06*, Kuala Lumpur, Malaysia, May 2006, 157-163

[2] A. Gupta and D. C. DuVarney, Using predators to combat worms and viruses: a simulation-based study. *Proc. of Computer Security Applications Conference*, Tucson, Arizona, USA, December 2004, 116- 125

[3] F. Castaneda, E. C. Sezer, J. Xu, WORM vs. WORM: Preliminary Study of an Active Counter-Attack Mechanism. *Proc. of WORM'04*, Washington DC, USA, October 2004, 83–93.

[4] H. Kim, I. Kang, On the functional validity of the worm-killing worm. *IEEE Communications*, Paris, France, June 2004, 1902-1906

[5] D. M. Nicol, M. Liljenstam, Models of Active Worm Defenses. *IPSI Conference*, Studenica, Serbia, June 2004

[6] H. Toyoizumi, A. Kara, Predators: Good Will Mobile Codes Combat against Computer Viruses. *Proc. of New Security Paradigms Workshop*. Virginia Beach, USA. September 2002, 13-21.