Game Theoretic Approach for Cost-Benefit Analysis of Malware Proliferation Prevention*

Theodoros Spyridopoulos, George Oikonomou, Theo Tryfonas, and Mengmeng Ge

Cryptography Group, University of Bristol
Merchant Venturers Building, Woodland Road, Clifton BS8 1UB, UK
{th.spyridopoulos,g.oikonomou,theo.tryfonas}@bristol.ac.uk,
gemengmeng.2011@my.bristol.ac.uk

Abstract. Many existing research efforts in the field of malware proliferation aim at modelling and analysing its spread dynamics. Many malware dissemination models are based on the characteristics of biological disease spread in human populations. In this work, we utilise game theory in order to extend two very commonly used malware spread models (SIS and SIR) by incorporating defence strategies against malware proliferation. We consider three different security mechanisms, "patch", "removal" and "patch and removal" on which our model is based. We also propose a cost-benefit model that describes optimal strategies the defender could follow when cost is taken into account. Lastly, as a way of illustration, we apply our models on the well studied Code-Red worm.

1 Introduction

With the ever growing importance of networked computing, malicious software, known as malware, has been a considerable threat to the realm of interconnected computers. Often built by cyber criminals, malware aims to compromise target computers with the ultimate goal of stealing sensitive data or gaining access to private systems. Malware includes a variety of malicious software such as computer viruses, worms, trojan horses, key-loggers and many others.

Defence mechanisms such as firewalls and anti-viruses have been developed in order to defend against malicious software. Those mechanisms investigate the problem of malware at micro level by utilising experimental and heuristic findings, such as virus signatures, in order to prevent or detect and cure a computer's infection. Nevertheless, malware spread in a network of computers underlines the need for a network-level solution. The increasing number of malware which bases its function on new techniques that are difficult to detect and mitigate renders conventional defence mechanisms unsuitable. In light of these challenges epidemiological models which can describe the dynamics of malware proliferation over a computer network have been proposed.

Additionally, Game Theory has been introduced in a number of occasions across the fields of computer and network security (e.g. [1,2]) to describe the interactions between attacker and defender and the ways they may affect each other. As malware acts based

* This work has been kindly supported by the Faculty of Engineering's Systems Centre and its Industrial Partners.

L.J. Janczewski, H.B. Wolfe, and S. Shenoi (Eds.): SEC 2013, IFIP AICT 405, pp. 28–41, 2013.

on inscribed behaviour coded by cyber criminals, approaching it as a threat agent on its own right under the premise of game theory becomes a reasonable assumption.

Our work aims at combining well-known epidemiology models with a game theoretic framework that can describe the state of the system when the defender uses various strategies against the proliferation of a random-scanning worm. We develop a game between defender and malware, taking into account the spread dynamics, so that defenders manage to compute their optimal strategy by minimising the cost of security, on a cost-benefit basis.

The rest of the paper is structured as follows. In Section 2 we give a basic background on epidemiology models and game theory as it is applied in malware analysis. Special emphasis is given to the "FLIPIT" game. In Section 3 we present the models and methods that we have developed. In Section 4 we present an application of our approach to the well studied case of the Code-Red worm computing an optimal strategy for the defender. Finally, Section 5 discusses the conclusions drawn form this work.

2 Background

2.1 General Description of Epidemiology Models

The way that viruses and worms spread in a computer network shares common characteristics with the dissemination of biological diseases in human populations, in a way that the analysis of malware can benefit from investigating the behaviour of biological diseases. There are two kinds of models for analysing malware proliferation in epidemiology: stochastic models and deterministic models. Stochastic models are used to analyse small-scale networks, while deterministic models are used to analyse large-scale networks [3]. In our work, in order to study the effect of mass action, we consider malware spread in a large computer network, thus we utilise deterministic models.

In general, individuals in the epidemic population have several states, including susceptible, infected, recovered. A large fraction of the models used, rely on the transitions between those states. Among these, two models have been widely used, the Susceptible-Infected-Recovered (SIR) created by Kermack and McKendrick in 1927 [4,5,6] and a modified version of it, known as the Susceptible-Infected-Susceptible (SIS) model. Both models assume that all individuals within a closed population (i.e. no births and deaths) are susceptible to the malware in the initial phase and an individual may go through each state sequentially.

In the SIS model, the state transitions of an individual form a circulation. The individual may recover from the infection, but there is still a chance to be reinfected. In other words, an individual becomes again susceptible to the malware after its recovery. In the SIR model, the final state is described as the recovered state. An infected individual can recover from the infection and become immune to the malware. An immunised individual cannot be reinfected by the same malware.

Typically, disease spread depends on common shared characteristics of the individuals in a population. In a network of computers, malware exploits certain vulnerabilities in the system in order to infect a host. Common practice of malware is to exploit vulnerabilities in software that the victim-host has installed. Thus, in order for a host to be considered as susceptible to a certain malware, it has to have installed the specific

software version that bears the vulnerability that the malware can exploit. In case it doesn't then it cannot be infected and thus cannot be considered as susceptible. In the real world, not every host in a network carries the same vulnerabilities. However, in our work, in order to simplify our model, we have made the assumption that our network is homogeneous. In other words, in a single network architecture an individual can infect every other individual. Furthermore, the network is assumed to be symmetric so that no preferential direction of the malware proliferation exists.

2.2 Mathematical Specification of Standard Epidemiology Models

In this section we present the mathematical specification of two commonly used epidemiology models, SIR and SIS. In general, such models are formulated over a fixed-size network. Nodes represent individuals. Links or edges between nodes represent contacts between individuals. The infection spreads along direct links between nodes.

The SIR Model. In the SIR model [4,5,6], the total population is divided into three parts: i) susceptible nodes (denoted by S), ii) infected nodes (denoted by I) and iii) recovered nodes (denoted by R). The differential equations 1, 2 and 3 depict the rate of change of the susceptible nodes, infected nodes and recovered nodes respectively over time [7]. Here β denotes the probability of a susceptible node to be infected by another infected node when they come in contact in each time unit, also regarded as the infection rate; γ denotes the probability of an infected node to recover from an infection and become immune to the malware in each time unit, also regarded as the recovery rate. In our research a contact is considered as a network link between two nodes, and since all nodes are connected with each other, either directly or through a number of hops depending on the network's topology, they are always in contact with each other.

$$\frac{dS}{dt} = -\beta IS \tag{1}$$

$$\frac{dI}{dt} = \beta IS - \gamma I \tag{2}$$

$$\frac{dR}{dt} = \gamma I \tag{3}$$

The SIS Model. In the SIS model, the total population is divided into two parts, susceptible nodes (denoted by S) and infected nodes (denoted by I). Equations 4 and 5 model the rate of change of susceptible nodes and infected nodes respectively over time [8]. Again, β denotes the probability of a susceptible node to get infected by an infected node when they come in contact in each time unit, also regarded as the infection rate; γ denotes the probability of an infected node to recover from an infection and become susceptible again to the malware in each time unit, also regarded as the recovery rate. Even though the term "recovery rate" is used in both the SIR and the SIS model, it is used for different purposes.

$$\frac{dS}{dt} = -\beta IS + \gamma I \tag{4}$$

$$\frac{dI}{dt} = \beta IS - \gamma I \tag{5}$$

2.3 Brief Introduction to Game Theory

Game theory provides us with a set of analytical tools designed to describe and analyse the phenomena observed when two or more decision makers interact [9]. Decision makers are identified as unique players and the formal description of the interaction between them is denoted as a game [10]. The basic assumption of game theory is that every player acts rationally, aiming at the best possible outcome, and take into account other players' decisions. Solution to a game is the description of the strategies that each player has to follow in order to achieve the best possible outcome. *Nash equilibrium* is the solution of the game that describes a steady state, where each player gets the best possible payoff. A deviation from the Nash Equilibrium strategy always leads in lesser payoff. Games are divided in various categories based on the nature of their parts:

- *Cooperative - Non-cooperative Games.* In general, games are divided into cooperative and non-cooperative games based on the way that players interact with each other. In cooperative games, all players try to maximise the overall payoff, while in non-cooperative games, each player cares only about his own gain and cost. In the field of network security, the research falls under the category of non-cooperative games since there is no cooperation between the attacker and the defender [1].
- *Static - Dynamic Games.* Under the category of non-cooperation, games are divided into static and dynamic games. In static games, all players make their decisions simultaneously not knowing other player strategies. They are one-shot games where each player has a pre-computed move list, each move denoted as a strategy, from which he has to choose the best move in order to maximise his personal benefit. Benefit, also known as payoff, refers to a player's net gain when he choses to play a strategy, and is described by Equation 6. In dynamic games, a player can alter his move during the game. The game is played into stages in each of which each player has to choose his move. A strategy in such games is defined as the combination of sequential moves chosen by the player in order to maximise his total benefit. Each stage of a dynamic game can be considered as a static game leading to a structure of sequential static games.

$$Benefit = Gain - Cost \tag{6}$$

- *Perfect - Imperfect Games.* A game where the players choose their strategies simultaneously, without knowing the choices of the other players, is an imperfect information game. Contrary, in perfect information games every player knows exactly the strategies that other players have followed before his turn. Thus only games where players play sequentially can be considered as perfect information games.
- *Complete - Incomplete Information Games.* Complete information games indicate that players know the available strategies and payoffs of the other players but do not necessarily know the strategies that have been played by other players. On the contrary, in incomplete information games players may not have access to other players' available strategies and payoffs during the game.

- *Pure - Mixed Strategies.* Pure strategies refer to deterministic actions taken by a player in the game for every possible situation that s/he can face (for every other players' actions). In mixed strategies a player's move is not based on a deterministic action-decision, but involves a probabilistic combination of the available pure strategies [9].

2.4 Game Theory in Security and Malware Analysis

Traditional network security mechanisms such as Intrusion Detection and/or Prevention Systems (IDS/IPS) analyse malware at a level of specific technical detail. They focus on collecting, dissecting and recording its structure and behaviour. This allows them to respond to attacks that are based on well known techniques. For instance, IDS algorithms apply malware-signature identification or make use of heuristic algorithms to detect suspicious system behaviours that indicate possible infection. Nevertheless, since they mostly rely on such experimental findings, they are proved to be insufficient against sophisticated attacks which may utilise unknown techniques (e.g. zero-day attacks).

A shortcoming of the traditional network security solutions is that they lack a macro-level quantitative decision framework [1] and various researchers have focused their work on utilising game theory in order to provide a holistic solution [11,12,13,2]. The relationship between attacker and defender can be modelled as the interaction between two competing parts in a game theoretic scenario. The malware's goal is to spread widely, whereas the defender aims at protecting the network against the attack (minimising spread) whilst keeping costs as low as possible. Game theory can be used for studying decision making problems in multiplayer scenarios, to examine and evaluate all possible scenarios given the outcomes of each player's strategy and return the best one.

The "FLIPIT" Game: In order to develop our game we first devised a cost-benefit model to help us compute the gain of each strategy. The cost-benefit model was originally based on another game theoretic model known as "The FLIPIT Game" [2]. In FLIPIT, the authors have developed a model that describes the situation in which an attacker periodically takes over a system and is not immediately detected by the defender. There are two players, the attacker and the defender, and a shared resource. The two opponents compete to control the shared resource. The attacker tries to put the resource into a bad state, while the defender puts the resource into a good state. The objective of each player is to control the resource for the largest possible fraction of time and minimise at the same time their total cost. Players do not know the current situation of the game when other players make a move; they learn that only when they make a move. Making a move incurs cost and taking over control gains benefit. Each player loses some points per move and gains some points per second when he is in control.

The mathematical description of the game is provided below. Here we assume that the defender is player 0 and the attacker is player 1. Player i pays k_i points per move and gains one point per second when the source is under his control.

$$\gamma_i(t) = \frac{G_i(t)}{t} \tag{7}$$

The total period of time t is the time the resource is controlled by the defender plus the time controlled by the attacker as shown in Equation 8.

$$G_0(t) + G_1(t) = t \tag{8}$$

Thus, for each player, the gain rate $\gamma_i(t)$ is equal to the fraction of time that player i has the shared resource under control, as shown in Equation 9.

$$\gamma_0(t) + \gamma_1(t) = 1 \tag{9}$$

Equation 10 calculates the benefit of a strategy, which is denoted as the gain minus the total cost. The aim of each player in the game is to maximise the value of benefit.

$$B_i(t) = G_i(t) - k_i \cdot N_i(t) \tag{10}$$

The generic description of a shared resource taken under control by an attacker is suitable to describe the situation of a computer network under attack from a worm. In our work we view the network as the shared resource which both attacker and defender try to take under control. However, in this context the shared resource cannot be instantly fully taken over, since a worm spreads in a fraction of the total population in each time step rather the whole population. Hence, only a fraction of the shared resource can be taken over by the attacker.

3 Models and Methods

3.1 Game Theoretical Models of Malware Proliferation

Worms have the ability to self-replicate and spread without human intervention in a network [14], resembling human viruses. In the human virus spread example this could mean that all individuals are always in contact with each other. In a network it means that an infected node can infect every other node in the network since all nodes are linked with one anoother. Random-scan worms have the ability to spread without topology constrains since they rely on random IP scans. On the other hand, worms spreading via emails have specific routes according to the email list of each infected computer. In our work we model random-scan worms.

There are three security mitigation practices against worm dissemination: i) Remove, ii) Patch and iii) both Patch and Remove. Under the SIR and SIS models, a susceptible node can either be patched against the certain worm and become immune to it or stay in the susceptible state. If a susceptible node is infected then it can either stay infected and consequently spread the worm or it can use the removal tool (e.g. an antivirus) in order to remove the worm. However, the removal tool does not encompass immunisation functionality. Thus, when an infected node removes the worm it returns back to the susceptible state where it can subsequently be reinfected. However, if an infected node uses both the remove tool and the patch against the worm then it moves to recovery state where it is immune against the specific worm. For each of the three security strategies we set up differential mathematical expressions, as in SIR and SIS models, which describe the dynamics of the system.

Patch Strategy: When the Patch Strategy is used, susceptible nodes can become immune to the worm, but infected nodes cannot recover from the infection. In this case, the worm and the defender seem to take part in a race. If the worm spreads very fast, it will infect most computers in a short time before defenders notice it; if people in the network can patch their computers much faster than the worm proliferation, the wide-range infection can be avoided. The model is depicted in Figure 1.

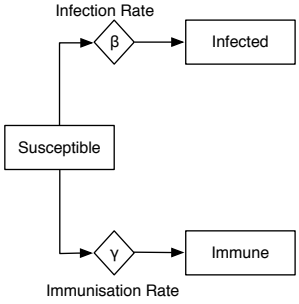

Fig. 1. Patch Strategy Model

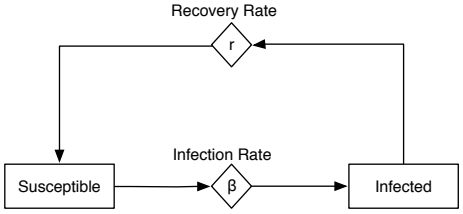

Fig. 2. Removal Strategy Model

The mathematical specification of the Patch Strategy is given in Equations 11,12 and 13, where S is the susceptible population, I is the infected population and R is the immune population. β is the probability that a susceptible node gets infected in each time unit, also regarded as infection fraction, and γ is the immunisation rate.

$$\frac{dS}{dt} = -\beta IS - \gamma S \tag{11}$$

$$\frac{dI}{dt} = \beta IS \tag{12}$$

$$\frac{dR}{dt} = \gamma S \tag{13}$$

Removal Strategy: When Removal Strategy is used, infected nodes can recover from the infection when the worm is detected and removed. However, nodes that have recovered from an infection are still susceptible to the specific worm since no immunisation against it is included. In this case the model is transformed into a SIS model where the system reaches an equilibrium where the number of infected nodes and the number of susceptible nodes stay almost constant. The model is depicted in Figure 2.

The mathematical specification of Removal Strategy is given in Equations 14 and 15. Again, S refers to the susceptible population and I refers to the infected population. β is the probability that a susceptible node gets infected and r is the removal or recovery rate. As seen, no recovered population is found in the system.

$$\frac{dS}{dt} = -\beta IS + \gamma I \tag{14}$$

$$\frac{dI}{dt} = \beta IS - \gamma I \tag{15}$$

Patch and Removal Strategy: The last strategy devised is the Patch and Removal. In this strategy both moves of patch and removal are available. A susceptible node can become immune to the worm when patch is used. Furthermore, an infected node can recover from the infection if the worm is removed and then become immune to the worm by using the patch. This is the most efficient, yet costly, way to eliminate malware spread. Eventually, all nodes in the network will be immune against the specific worm. The strategy model is shown in Figure 3.

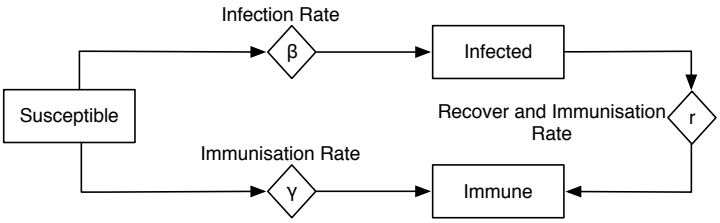

Fig. 3. Patch and Removal Strategy Model

The differential equations that describe the dynamics of the model are shown in Equations 16, 17, 18 and 19. S refers to the susceptible population, I refers to the infected population, R is used for the recovered and immunised population and Q for the population that becomes immune to the malware. As before, β is the probability that a susceptible node gets infected, γ refers to the immunisation rate when a susceptible node uses the specific patch and λ is the "removal and patch" rate.

$$\frac{dS}{dt} = -\beta IS - \gamma S \tag{16}$$

$$\frac{dI}{dt} = \beta IS - \lambda I \tag{17}$$

$$\frac{dR}{dt} = \lambda I \tag{18}$$

$$\frac{dQ}{dt} = \gamma S + \lambda I \tag{19}$$

3.2 Cost-Benefit Analysis

In "FLIPIT", two opponents compete in order to gain full control of a shared resource and gain is defined by the time the resource is under one's control. In our epidemiology model, the shared resource is the node population of the network. In each time unit

the two opponents (attacker and defender) perform actions to take under their control a part of the population (a number of neighbouring nodes). For instance, the attacker takes under control I nodes in each time unit, and I changes according to the equations presented above. Hence, gain is defined by the average fraction of node population under one's control. Therefore, by considering player 0 as defender and player 1 as attacker we define $G_i(t)$ the gain of player i and calculate it as shown in Equation 20, where $P_i(t)$ is the fraction of population under control by player i over time and t_k is the total time for which our model is running.

$$G_i(t) = \frac{1}{t_k} \int_0^{t_k} P_i(t)dt \tag{20}$$

Since there are only two fractions of populations, one under the control of the defender and one under the control of the attacker, we can say that $P_0(t) = 1 - P_1(t)$. Hence: $G_0(t) + G_1(t) = 1$.

For player 0, we define cost ($C_0(t)$) as the total number of moves made by player 0 (number of times that has used the security tool) ($n_0(t)$), multiplied by each move's cost (k_0) (Equation 21).

$$C_0(t) = n_0(t)k_0 \tag{21}$$

We define as cost for player 1 the perceived complexity of the algorithm that their malware implements.

Each player's benefit is equal to the player's total gain minus the cost (Eq. 22).

$$B_i(t) = G_i(t) - C_i(t) \tag{22}$$

In order to compute cost, we utilise quantitative tables of operational complexity. A strategy by either player (e.g. Patch Strategy for the defender or Code-Red worm for the attacker) may encompass several actions, with each action characterised by a complexity level. We set up empirically three levels of perceived complexity, low, medium and high, and assign a score to each of them, 1, 2 and 3 respectively. Therefore, the cost of a move for player 0 or the total cost of player 1 is equal to the sum of the costs of the actions it involves.

4 Application of the Model to Code-Red's Parameters

In this section we apply our game theoretic model to a real case of malware proliferation, the well known Code-Red worm. Albeit old, we chose Code-Red because it is a random-scanning worm with no topology constraints and so its characteristics fit well into the generic nature of our abstraction. It is self-activated by exploiting a vulnerability which exists in the host operating system. Other worms, such as Conficker, utilise various spread methods, e.g. through email, which would warrant specialisation of the differential equations describing the proliferation and mitigation strategies.

4.1 Code-Red

Code-Red was discovered in July 2001. It exploits a buffer-overflow vulnerability in Microsoft IIS Web Server [15]. It produces a list of random IP addresses and launches 99 threads to search each computer in the list in order to infect as many vulnerable computers as possible. It has two versions: Code-Red v1 and Code-Red v2. Code-Red v1 spreads slowly because it generates an identical list by using a static seed. Code-Red v2 is the variant of Code-Red v1. It can infect new nodes by using a random seed for its pseudo-random generator. Therefore, the latter version has a higher spread speed. The greatest damage caused by the worm is that it could launch a massive DoS attack. Finally, this worm is memory resident. Thus a reboot can clear the worm from a host node. However, in order to prevent the infection or reinfection, nodes have to use a specific patch [15].

We make the assumption of a network with a population of 10,000 susceptible nodes and 1,000 nodes immune to the worm due to not every node running susceptible software - a Microsoft ISS Web Server. We set the maximum time of worm spread period at one week or $t_k = 168$ hours. According to [16], an infected node infects other nodes with rate 1.62 nodes per hour. Thus the probability of a susceptible node in our network to get infected by an infected node in each second is equal to $1.62/N$ where N is the total population. Hence, $\beta = 1.47 \cdot 10^{-4}$. The costs for each player are shown in Tables 1 and 2. Since the attacker uses a specific malware that is not able of changing behaviour, there is no reason in computing an optimal strategy since this has already been chosen (the Code-Red worm's inscribed behaviour). We analytically compute the defender's gain according to Equation 20, however the calculations are not shown in this paper due to space restrictions.

Table 1. The cost for Code-Red worm

Actions	Complexity Low:1 Medium:2 High:3	Total
Exploit the buffer vulnerability	2	
Generate random IP addresses	1	4
Launch 99 threads with generated IP addresses	1	

Using Vensim as a simulation environment, we set up three simulations for the three security strategies (Patch, Remove, Patch and Remove) respectively, according to our models, in order to find the best strategy that the defender can follow. In our case study we assume that the defender has already chosen a patch rate or a removal rate so that he has only to chose the security strategy that he will follow. In an alternative scenario, the defender could also utilise our model in order to find both the security strategy and the rates that could give him the best possible benefit.

4.2 Patch Strategy

For the Patch Strategy the state of our system is shown in Figure 4. It can be seen that at the early hours of the worm spread the number of infected nodes increases

Table 2. The cost of each move for the defender

Actions		Complexity Low:1 Medium:2 High:3	Total
Patch	Detection	2	4
	Patch	2	
Removal	Detection	2	3
	Reboot	1	
Patch and Removal	Detection	2	5
	Reboot	1	
	Patch	2	

sharply while on the other hand the number of susceptible nodes decreases. Since infected nodes cannot recover and patched nodes cannot be infected, the system reaches soon its equilibrium where the Infected population is near 8,351 nodes and the immunised population around 1,649 nodes. If the defender used a much larger patch rate then they might be able to compete with the attacker in this race. However, this would not be easy since the number of infected nodes increases exponentially, whereas the number of immunised nodes linearly. Furthermore, an increase in the security rate would increase the total security cost. Based on the results of the simulation, the cost (using the Table 2) and the gain (the average number of non-infected nodes) of the Patch strategy are given by Equation 23. Thus the Benefit for the Patch strategy is equal to $B_0 = G_0 - C_0 = -4,652$.

$$\begin{aligned} C_0 &= number\ of\ patches \times cost\ of\ patch = 1649 \cdot 4 = 6596 \\ G_0 &= 1944 \end{aligned} \tag{23}$$

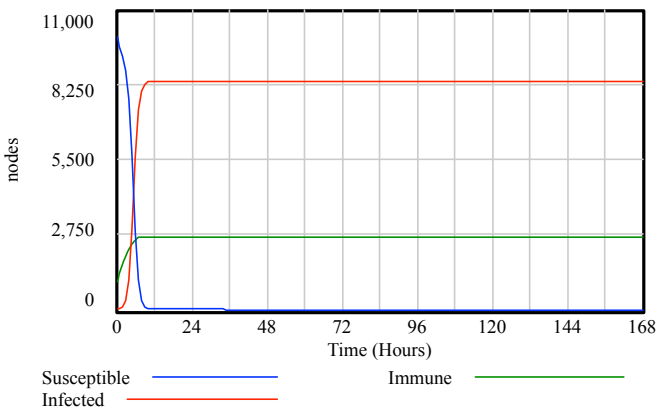

Fig. 4. Patch Strategy

4.3 Removal Strategy

For the Removal Strategy the state of our system is shown in Figure 5. It can be seen that after a period of time the system reaches an equilibrium where the populations of susceptible and infected nodes remain constant, 222 and 9,778 respectively. Again, the results would be better if the defender used a larger recovery rate. Based on simulation results, the cost and the gain (the average number of non-infected nodes) of the Removal strategy are given in Equation 24. Thus the Benefit for the Removal strategy is equal to $B_0 = G_0 - C_0 \simeq -156,152$.

$$\begin{gathered} C_0 = number\ of\ removals \times cost\ of\ removal = 52236 \cdot 3 = 156708 \\ G_0 = 556 \end{gathered} \tag{24}$$

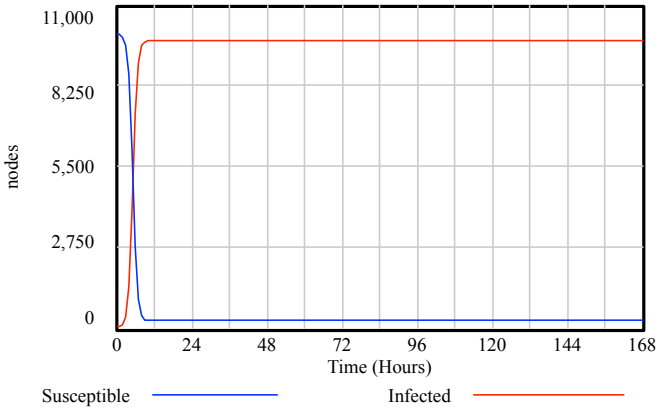

Fig. 5. Removal Strategy

4.4 Patch and Removal Strategy

The state of our system in time is shown in Figure 6. At the early hours of dissemination, the worm spreads exponentially into the network. However, as time passes more and more infected nodes recover form the infection and get immunised against the worm. Furthermore, the number of susceptible that are patched also increases. Thus, eventually every node in the network will be patched and thereby immunised against the worm. The dissemination slowly fades. Based on the results of the simulation, the cost and the gain of the Patch and Removal strategy are given in Equation 25. Hence, the Benefit for the defender's Patch and Removal strategy is equal to $B_0 = G_0 - C_0 \simeq -39,519$.

$$\begin{gathered} C_0 = number\ of\ patches\ \times\ cost\ of\ patch\ + number\ of\ removals\ \times\ cost\ of \\ patch\ and\ removal = 1771 \cdot 4 + 8188 \cdot 5 = 48024 \\ G_0 = 8505 \end{gathered} \tag{25}$$

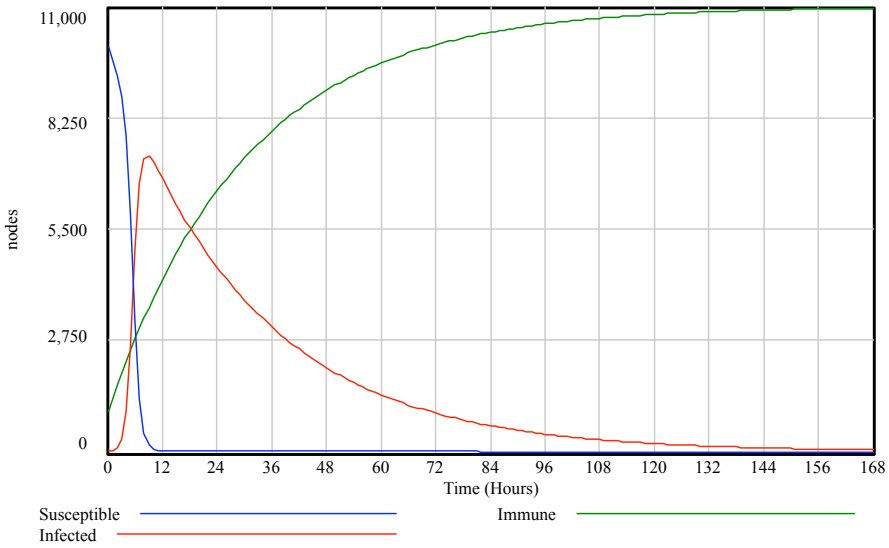

Fig. 6. Patch and Removal Strategy

4.5 Discussion

Our analysis indicates that the Patch and Removal strategy is the most efficient defence against worm dissemination, since this is the only one eventually leading to zero infected nodes left in the network. However, it is not the one that would give the best benefit to the defender. That is because, when the Patch strategy is used, infected nodes cannot be patched. Thus the total number of patches is significantly smaller and hence the total cost for the defender is lower. Therefore, although Patch and Removal is the most efficient strategy, when there are cost restrictions Patch strategy can also be used. An interesting approach could be the usage of a mixed strategy based on them.

5 Conclusions

In this paper we have integrated premises of game theory with malware proliferation models and developed a cost-benefit game-theoretic approach to evaluate defence strategies that mitigate malware proliferation. We applied our approach to a case study where defender could choose between three strategies i) “patch”, ii) “removal” and iii) “patch and removal” and discuss our results for both the spread and the cost-benefit strategy selection. Our model can be extended by introducing more options for the defender, such as the ability to change the security (i.e. patch and removal) rate; as well as the attacker, e.g. change the manifested behaviour of the worm deployed.

As mentioned, we kept the rates of patch, removal and the infection rate constant in our simulations. However, defender can vary their security rates in order to achieve better results. Furthermore, an even more complicated approach would be to give the

attacker the option to choose among strategies, in other words vary the malware's behaviour (e.g. metamorphic viruses). Thereby, we could establish a game where both players try to find their optimal strategies.

References

1. Roy, S., Ellis, C., Shiva, S., Dasgupta, D., Shandilya, V., Wu, Q.: A survey of game theory as applied to network security. In: 2010 43rd Hawaii International Conference on System Sciences (HICSS), pp. 1–10 (January 2010)
2. van Dijk, M., Juels, A., Oprea, A., Rivest, R.L.: FlipIt: the game of "Stealthy takeover". Technical Report 103 (2012)
3. Andersson, H., Britton, T.: Stochastic Epidemic Models and Their Statistical Analysis. Springer (July 2000)
4. Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. Proceedings of the Royal Society of London. Series A 115(772), 700–721 (1927)
5. Kermack, W.O., McKendrick, A.G.: Contributions to the mathematical theory of epidemics. II. the problem of endemicity. Proceedings of the Royal Society of London. Series A 138(834), 55–83 (1932)
6. Kermack, W.O., McKendrick, A.G.: Contributions to the mathematical theory of epidemics. III. further studies of the problem of endemicity. Proceedings of the Royal Society of London. Series A 141(843), 94–122 (1933)
7. Capasso, V., Serio, G.: A generalization of the kermack-McKendrick deterministic epidemic model. Mathematical Biosciences 42(12), 43–61 (1978)
8. Van der Molen, H.: Math on malware. ISACA Journal 3, 40–47 (2011)
9. Osborne, M.J., Rubinstein, A.: A course in game theory. MIT Press, Cambridge (1996)
10. Turocy, T.: Texas a&m university. Bernhard von Stengel, London School of Economics "Game Theory" CDAM Research Report (October 2001)
11. Lin, J.C., Chen, J.M., Chen, C.C., Chien, Y.S.: A game theoretic approach to decision and analysis in strategies of attack and defense. In: Proceedings of the 2009 Third IEEE International Conference on Secure Software Integration and Reliability Improvement, SSIRI 2009, pp. 75–81. IEEE Computer Society, Washington, DC (2009)
12. Wu, Q., Shiva, S., Roy, S., Ellis, C., Datla, V.: On modeling and simulation of game theory-based defense mechanisms against DoS and DDoS attacks. In: Proceedings of the 2010 Spring Simulation Multiconference, SpringSim 2010, pp. 159:1–159:8. Society for Computer Simulation International, San Diego (2010)
13. Khouzani, M., Sarkar, S., Altman, E.: A dynamic game solution to malware attack. In: 2011 Proceedings IEEE INFOCOM, pp. 2138–2146. IEEE (2011)
14. Saudi, M., Tamil, E., Nor, S., Idris, M., Seman, K.: Edowa worm classification. In: Proceedings of the World Congress on Engineering, vol. 1 (2008)
15. Moore, D., Shannon, C., Brown, J.: Code-Red: a case study on the spread and victims of an Internet worm. In: ACM SIGCOMM/USENIX Internet Measurement Workshop (IMW), Marseille, France, pp. 273–284 (November 2002)
16. Vojnovic, M., Ganesh, A.: On the race of worms, alerts, and patches. IEEE/ACM Transactions on Networking (TON) 16(5), 1066–1079 (2008)