

# USING GAME THEORY TO ASSESS THE STRENGTH OF AN AV SYSTEM AGAINST EVOLVING OFFENCES

*Bin Mai*

Northwestern State University, Natchitoches,  
Louisiana, USA

*Anshuman Singh, Andrew Walenstein,  
Arun Lakhotia*

University of Louisiana at Lafayette, Lafayette,  
Louisiana, USA

Email maib@nsula.edu,  
{axs6222, walenste}@cacs.louisiana.edu,  
arun@louisiana.edu

## ABSTRACT

Many AV scanners are heterogeneous compositions of components, with each component specially tuned to work on a certain class of threat. Each component may have individually tunable parameters and different performance characteristics. No general theory is known for composing such components and assigning their individual parameters in order to ensure optimal resistance to attack. A particularly important question is posed by the possibility of attackers using anti-AV techniques like stealth, which may fool the system into using different components. This paper introduces a framework for modelling composite AV systems as classifiers wired together using selectors. It then uses game theory to analyse possible attacks. According to the game analysis, using a selector is beneficial only when the cost of developing an anti-AV technique to game it is above a certain threshold. Further, the AV system is always better off by configuring its detection components so as to deter attackers from deploying anti-AV techniques, and this can be achieved by decreasing the detection rate of the classifier designed specifically for that class of malware, and increasing the detection rate for the classifier designed for clean files.

## 1. INTRODUCTION

Many computer defence systems rely on multiple components that are compiled into a single system that, in combination, is used to defend against attacks. For example, a mail server may pass incoming mail to multiple AV products from different vendors before letting the mail through. And, at a finer level of granularity, a single AV product may also be composed of several discernible detector components. For example, a single product may include a component for matching cryptographic checksums, for ordinary signatures, for so-called ‘x-ray’ scanning, for static behaviour-based patterns, an emulation-based behaviour matcher, and a run-time behaviour matcher based on monitoring hooked system calls [1].

There are several important rationales for constructing heterogeneous AV systems. First, it is often simply good software engineering practice to decompose large systems into smaller, well-defined components. Second, it may be the case that certain detector components are applicable only to certain inputs. For example, an AV system’s static behaviour pattern

matcher may be known to work well only on specific types of malicious files. Third, there may be important performance reasons for dividing the work up between components; in particular, certain components may incur much higher computational costs than others, so it is important to ensure that they are used only in those situations in which they are most likely to be needed, rather than on all files. Whatever the reasons for using multiple components, they must be wired together in such a way that they work in coordination to perform the detection. The logic used to select the inputs that the various components will work on is an essential element, so that the computational costs are kept low and the components are used only on the appropriate inputs. Another issue to consider is the settings of the components’ tunable parameters.

Of critical concern is whether the system, as a whole, is made more resilient to attack by virtue of its combination of components and connection logic. A specific problem is caused by the possibility of using various anti-AV techniques, such as stealth, to game the selection of different classifiers. In particular, anti-AV techniques may be used to fool a single detector component into making the wrong decision but, with selection logic added to the system, new anti-AV attacks are possible directly on the selection logic. Thus new questions arise as to whether the compositions are more resistant than the individual components, and as to how to assign any detector parameters that are tunable. No generic framework or analysis method is known for answering such questions.

This paper proposes a modelling framework and analysis technique that can help begin to answer such critical questions. The framework for modelling heterogeneous AV systems treats them as a combination of *classifiers* connected together using probabilistic *selectors*. From such models, defence construction (AV system setup) and attacks on them are treated as a game. For example, the virus-antivirus coevolution described by Nachenberg [2] can be modelled as a game in this framework. A game-theoretic analysis can then be performed that can expose potential attack weaknesses. By setting up a game using variables in the models instead of actual constants, an abstract game model can be constructed, that is, equations for player payoffs can be extracted.

Using a sample game with a two-component AV system, the paper shows that interesting general characteristics of composite AV systems can be extracted. Specifically, it characterizes the conditions in which the AV system as a whole is made *weaker* by the addition of a selector and specific classifier. More specifically, we found first that within our model setting, augmenting detection with a selector would not always benefit the AV system. The selector’s value can be fully realized only when the cost of developing anti-AV techniques is above a certain threshold. Secondly, the AV system is always better off by configuring its classifiers so as to deter malware authors from deploying anti-AV techniques, and this can be achieved by decreasing the detection rate of the classifier designed for malware and increasing the detection rate of the classifier designed for the normal files. This implies that when the anti-AV development cost is low and selection accuracy is high, the difference in detection rates of the classifiers should be low for optimal performance of the AV system.

The rest of the paper is organized as follows: we describe our model of the detection game in Section 2. We analyse the impact of selector-classifier architecture in Section 3.

In Section 4, we discuss the implication of our results and conclude the paper.

## 2. MODELLING HETEROGENEOUS AV SYSTEMS

### 2.1 Simple AV system vs. composite AV system

A simple AV system with a single detection component can be thought of, abstractly, as a *classifier* that classifies its inputs into one of possibly several categories. In this case, the inputs are potentially malicious programs, and the output classes might, for example, be *clean*, *suspicious*, and *dirty*. Classifiers such as these may be connected together in parallel so that for any given input all classifiers are run, and the outputs are combined in some manner. An example is shown in Figure 1(a). In such configurations, well-understood analysis methods such as *boosting* can be found in the classifier literature [3].

For composite AV systems such configurations are not desirable since not only is it too costly to run all classifiers on all inputs, it is frequently the case that certain classifiers are specialized to work only on certain subsets of the input space. For example, consider the case of a *normalizing* detector for metamorphic malware similar to the one defined by Walenstein *et al.* [4]. Although the algorithm used is more efficient than semantics-based static normalization approaches, the normalization is likely most helpful only for a small number of files, so for performance reasons the normalizer is likely to be combined with a selector component that can quickly filter out the files that are highly unlikely to need the normalization. For example, a fast statistical selector [5] might be used in combination with the normalizer, as in Figure 1(b). It selects whether the incoming file is likely to be metamorphic and in need of going through the normalization process [4]. In this way, the majority of files need not be scrutinized by the more heavyweight normalizer. This is a classic instance of a specialized classifier approach. For these, a different model for classifier combination must be used as compared to models such as Figure 1(a).

### 2.2 Game-theoretical model of an AV system

Game theory, a branch of applied mathematics, attempts mathematically to capture behaviour in strategic situations, in which an individual agent's success in making choices

depends on the choices of other agents. Applications of game theory attempt to find equilibria in these games – the combination of the strategies for each agent in which none of the agents have incentive to change their strategy. This is an analytical tool that is especially valuable in analysing situations where there are strategic interactions among multiple agents and each agent's behaviour and consequences are related intricately to the others'.

The following steps are involved in developing a game-theoretical model for capturing the strategic interactions between multiple agents:

1. Identify the various agents, where each agent typically represents a role in the interaction.
2. Identify various parameters of the game, which include the payoffs for each agent for various outcomes of the game and the probabilities for making various decisions.
3. Develop a tree representing the strategic interactions between the agents.
4. Analyse the tree to compute the expected payoffs to obtain the optimal strategies for each agent.

The remainder of this section describes steps 1 to 3 of using game theory to model the strategic interaction between the agents involved in a security scenario. The final step, analysis using the tree, is described in the section that follows.

The agents involved in a security scenario may be modelled in three classes: a normal user (NU), a malware author (MA), and a security analyst (SA). NU sends normal files through the AV system to derive positive payoff to NU as well as to SA. MA tries to develop malware to attack the AV system to cause damage to SA and maximize MA's expected total payoffs. Meanwhile, SA attempts to devise an optimal configuration of the AV system to detect MA's malware and thus minimize the AV system's expected total cost.

When NU sends normal files to the AV system, NU obtains a payoff of  $\mu_n$ , while SA derives a positive utility of  $v$ . We assume that if SA successfully detects the malware, the AV system avoids any loss and MA gets  $\mu_l$  payoff ( $\mu_l > 0$  such that SA always has incentive to try to block an attack). If SA fails to detect the malware, the AV system incurs a damage of  $d$  and MA obtains a payoff of  $\mu_h$  ( $\mu_h > \mu_l$ ).

A classifier (or an AV system) can never detect MA's malware with complete accuracy. Let  $p_D$  denote the detection rate (true positive rate) – that is, the probability that the classifier detects MA's malware correctly. Classifiers may also raise false alarms when scanning through a normal user's files. We denote the false alarm rate (false positive rate) by  $p_F$ . A classifier can be configured to operate at a specific combination of  $(p_D, p_F)$  values on its Receiver Operating Characteristics (ROC) curve, which specifies the permissible combinations for the device [6]. A ROC curve represents  $p_D$  as an increasing concave function of  $p_F$ . We assume that the ROC curve is

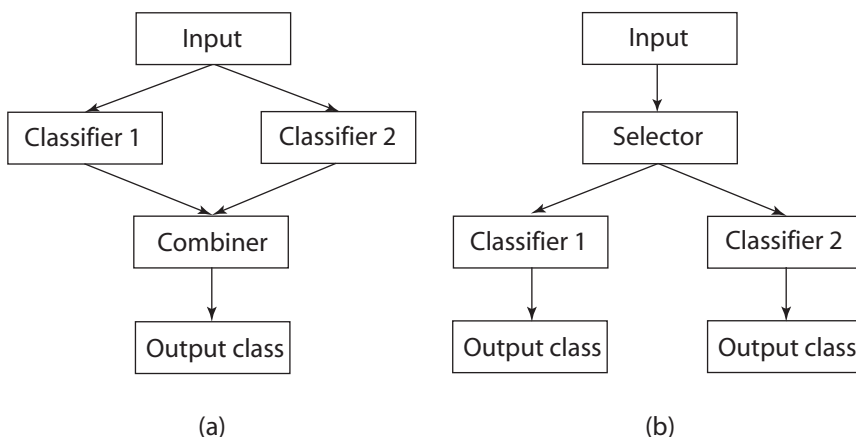


Figure 1: Two methods of composing multiple classifiers in an AV system.

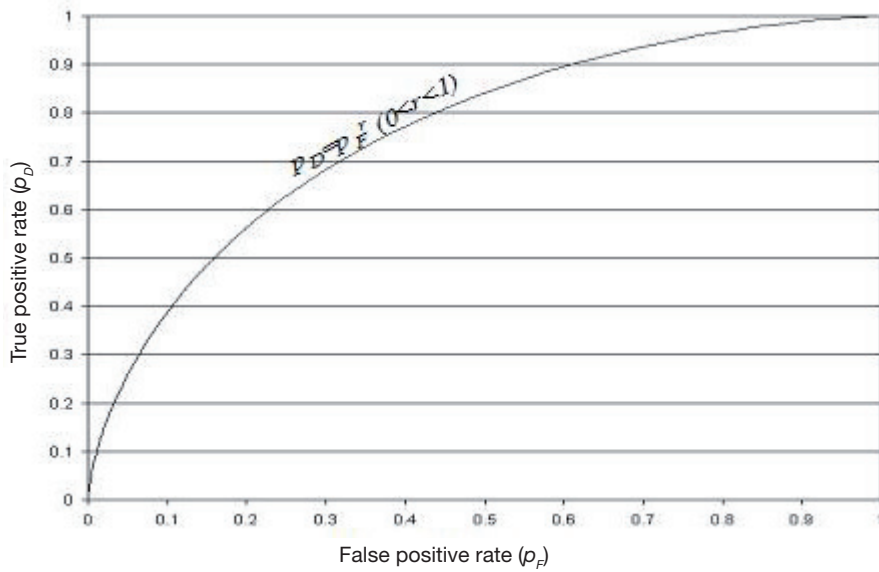


Figure 2: A ROC curve.

given by the power function  $p_D = p_F^r$ , with  $0 < r < 1$ . Figure 2 shows a sample ROC curve.

Once the AV system generates a positive alarm, regardless of whether it is a true positive or false positive, we assume SA would incur a cost of  $c$  for processing the alarm. When the AV system generates a false alarm towards NU's file, we assume NU incurs a cost of  $\beta$ . No alarm-related cost would be incurred to either NU or SA if no alarm is generated.

We now develop the game to represent interaction between the various agents. We do so using two scenarios for constructing an AV system:

1. *The baseline architecture.* The baseline scenario consists of an AV system with a single method for classifying a file as malicious or non-malicious. This baseline is useful for showcasing the use of the game-theoretic model.
2. *The selector-classifier architecture.* In the selector-classifier architecture we model a sample system containing two classifiers, one of which classifies normal files (white list) and the other

classifies malicious files (black list). The choice of which classifier to use is made by a selector, as in Figure 1(b).

Since the baseline architecture has only one component, without any choices, there is no opportunity for MA to exploit. For this architecture Figure 3 shows the payoffs for the three agents when the classifier correctly and incorrectly classifies a file.

The leftmost node shows the incoming file, which may be a file sent by NU or MA. We assume that  $\lambda$  portion of all the incoming files to the system are malware. As mentioned earlier, the classifier is configured to operate at the  $(p_D, p_F)$  values on its ROC. Figure 3 shows that the classifier raises an alarm for a normal file with

a probability of  $p_F$ , the false alarm rate of the classifier, and does not (correctly) raise an alarm with a probability of  $(1 - p_F)$ . Similarly, if the file is malicious, the classifier correctly raises an alarm with a probability of  $p_D$ , the true-positive rate of the classifier. The right column of Figure 3 shows the payoffs for the three agents for each outcome.

The game tree for the selector-classifier architecture, shown in Figure 4, is a bit more complex. It models MA's attempt at gaming the system. When MA notices that SA uses a selector for pre-screening to choose between a classifier for normal files and a classifier for malware, MA may attempt to use a stealth technique to cause the selector to send its file to the classifier with a lower detection rate with respect to detecting malicious files. This requires MA to incur an additional cost of  $\Delta$ , to develop the stealth. Thus, the game tree for selector-classifier architecture, Figure 4, has an additional node in the tree when the input is a malicious file. This node, 'using stealth' represents the choice made by MA.

In Figure 4, the accuracy of the selector is denoted as  $t_i, i \in \{NU, MA\}$ , which is the probability of correctly classifying type  $i$  incoming files. Similarly, the operating

characteristics of the normal and malware classifiers are represented by the pairs  $(p_D^{NU}, p_F^{NU})$  and  $(p_D^{MA}, p_F^{MA})$ , respectively.

In summary, Figure 4 represents the SA-MA interactions as a game that proceeds along the following timeline:

1. SA decides on the selection and configuration of classifiers (i.e. how many classifiers to use, and the detection rate configured for each classifier).

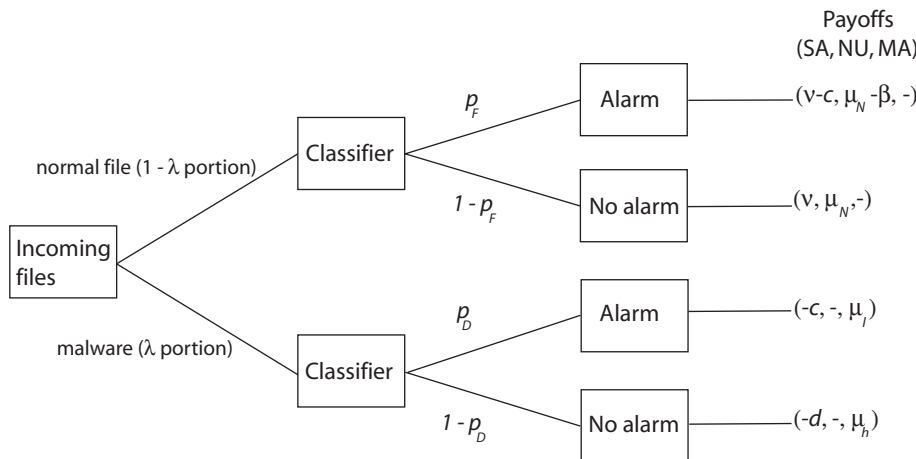


Figure 3: Game tree for the scenario where SA uses one classifier.

2. MA decides on the anti-AV technique to use.
3. The payoffs/losses are realized.

This is a general game because variables and formulae replace specific constants and calculations. Thus, it applies for a whole class of similar composite detectors that share the same basic composite structure. However it is a *sample* general analysis because it applies only to one specific composition graph. In general, the overall framework could be used to analyse other composition graphs, but we will only look at this one and then analyse the properties.

### 3. GAME-THEORETIC ANALYSIS OF OPTIMAL STRATEGIES

We now demonstrate how the game trees representing the interaction between agents may be analysed to compute the expected payoffs for obtaining the optimal strategies for each agent. We follow the reverse order of the timeline of the game to determine the optimal strategy for SA, i.e. we first determine the optimal strategy for MA given SA's configuration and classification decisions, followed by SA's decisions. We normalize the total number of malware to one for our analysis.

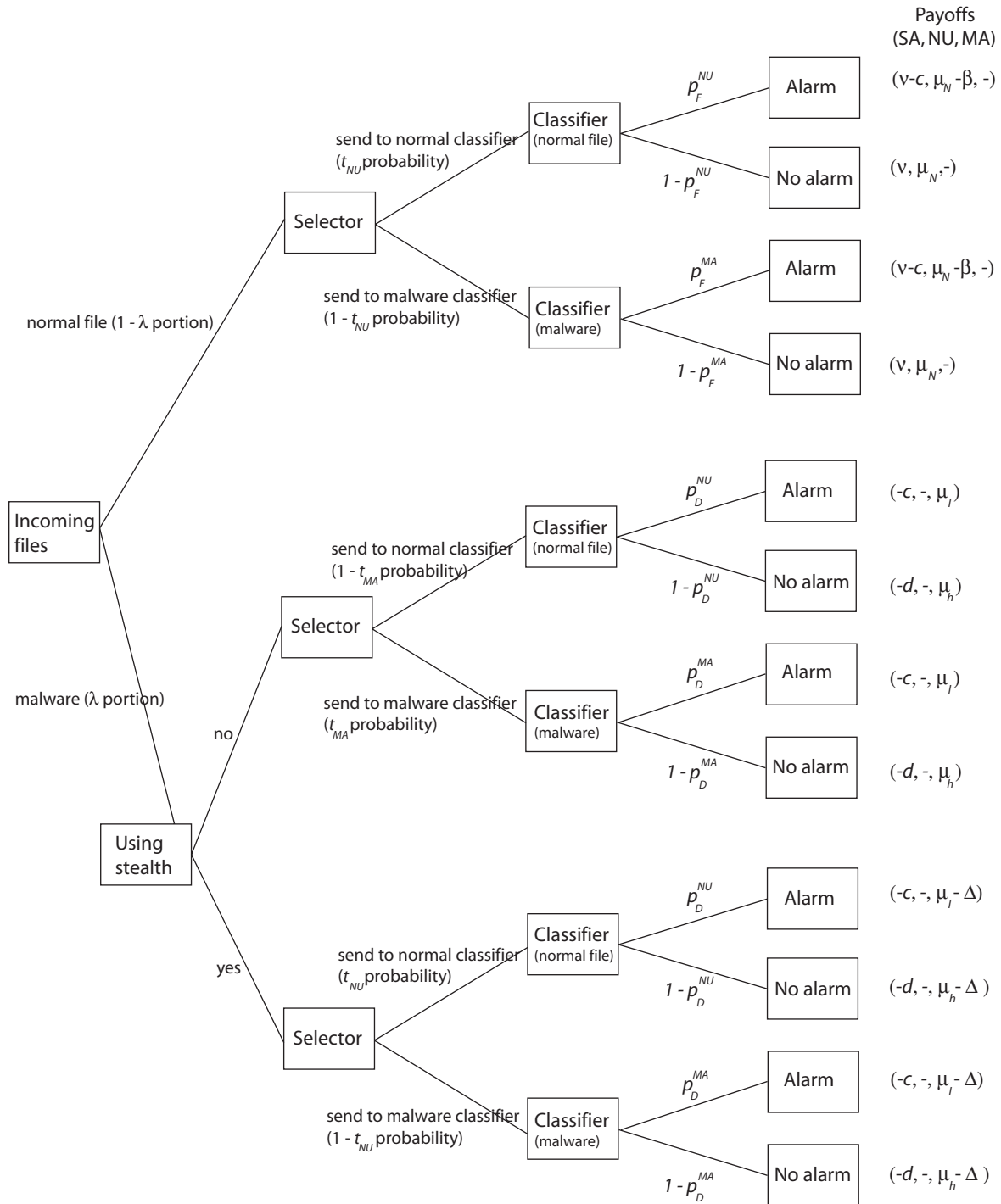


Figure 4: Game tree for the scenario where SA uses a selector and two classifiers.

### 3.1 The baseline scenario

In the baseline scenario, SA chooses to use only one classifier and does not pre-screen incoming files, thus every file is sent through the same classifier device. Figure 3 shows the game tree for this scenario.

The expected payoff for each agent in the game is obtained by adding the weighted sum of all payoffs for that agent for all the cases in the game tree [7]. For example, to calculate the expected payoff for NU, from Figure 3, we multiply NU's payoff in each end node with the probability that the game reaches that node, and sum up to obtain the overall expected payoff. As mentioned before, NU get a payoff of  $\mu_N$  for using the system, and incurs a cost of  $\beta$  for a false alarm. We do not model a payoff for NU for a malicious file. Thus, NU's payoff for 'alarm', i.e. false alarm, is  $(\mu_N - \beta)$  and for 'no alarm' is  $\mu_N$ . The probability of receiving a false alarm is  $p_F$  and no false alarm  $(1 - p_F)$ . This leads to NU's payoff to be  $(\mu_N - \beta)p_F + \mu_N(1 - p_F)$ , which simplifies to  $\mu_N - \beta p_F$ .

The expected payoffs for all agents are thus given by the following:

$$\begin{aligned} \text{NU: } & \mu_N - \beta p_F \\ \text{MA: } & \mu_h(I - p_D) + \mu_l p_D \\ \text{SA: } & v - (d + v)\lambda - c(I - \lambda)p_F + (d - c)\lambda p_D \end{aligned} \quad (1)$$

Since in this baseline scenario SA chooses not to use a selector, MA's anti-AV technique targeted at the selector becomes a non-issue: all the incoming files will be sent to a single classifier. Therefore, SA's decision is to configure the optimal value of  $p_D$  to maximize equation (1). This yields the optimal solution for SA in the baseline scenario:

$$p_D = \left( \frac{c(1 - \lambda)}{(d - c)\lambda r} \right)^{\frac{r}{r-1}}$$

### 3.2 The selector-classifier scenario

In this scenario, SA decides to implement two classifiers, one of which targets the files from NU, while the other targets the files from MA. SA uses a selector to pre-screen the incoming files and decide to which classifier each file should be sent.

Figure 4 shows the game tree for this scenario. With the selector scenario, the expected total payoff to all the agents in the game would be as follows:

$$\text{NU: } t_{NU}((\mu_N - \beta)p_F^{NU} + \mu_N(1 - p_F^{NU})) + (1 - t_{NU})(p_F^{MA}(\mu_N - \beta) + (1 - p_F^{MA})\mu_N)$$

MA if not using anti-AV:

$$(1 - t_{MA})(\mu_p p_D^{NU} + \mu_h(1 - p_D^{NU})) + t_{MA}(\mu_l p_D^{MA} + \mu_h(1 - p_D^{MA}))$$

MA if using anti-AV:

$$(1 - t_{NU})(\mu_l - \Delta)p_D^{MA} + (\mu_h - \Delta)(1 - p_D^{MA}) + t_{NU}((\mu_l - \Delta)p_D^{NU} + (\mu_h - \Delta)(1 - p_D^{NU}))$$

SA:  $p(\text{selected as normal, alarm}) * [p(\text{normal file} \mid \text{selected as normal, alarm}) * v - c] + p(\text{selected as normal, no alarm}) * [p(\text{normal file} \mid \text{selected as normal, no alarm}) * v - p(\text{malware} \mid \text{selected as normal, no alarm}) * d] + p(\text{selected as malware, alarm}) * [p(\text{normal file} \mid \text{selected as malware, alarm}) * v - c] + p(\text{selected as malware, no alarm}) * [p(\text{normal file} \mid \text{selected as malware, no alarm}) * v - p(\text{malware} \mid \text{selected as malware, no alarm}) * d]^1$

<sup>1</sup> The detailed mathematical expression for SAs can easily be derived, but is too complex to present.

Maximizing SA's expected payoffs, we can obtain the optimal configuration of the two classifiers, which can be summarized as the following:

If

$$\Delta \geq (t_{MA} + t_{NU} - 1) \left[ \left( \frac{c(1 - \lambda)(1 - t_{NU})}{(d - c)\lambda r t_{MA}} \right)^{\frac{r}{r-1}} - \left( \frac{c(1 - \lambda)t_{MA}}{(d - c)\lambda r(1 - t_{MA})} \right)^{\frac{r}{r-1}} \right] (\mu_h - \mu_l)$$

the optimal configuration of the two classifiers would be:

$$p_D^{MA} = \left( \frac{c(1 - \lambda)(1 - t_{NU})}{(d - c)\lambda r t_{MA}} \right)^{\frac{r}{r-1}}$$

and

$$p_D^{NU} = \left( \frac{c(1 - \lambda)t_{MA}}{(d - c)\lambda r(1 - t_{MA})} \right)^{\frac{r}{r-1}}$$

If

$$\Delta < (t_{MA} + t_{NU} - 1) \left[ \left( \frac{c(1 - \lambda)(1 - t_{NU})}{(d - c)\lambda r t_{MA}} \right)^{\frac{r}{r-1}} - \left( \frac{c(1 - \lambda)t_{MA}}{(d - c)\lambda r(1 - t_{MA})} \right)^{\frac{r}{r-1}} \right] (\mu_h - \mu_l)$$

the optimal configuration of the two classifiers would satisfy the following equations:

$$p_D^{MA} - p_D^{NU} = \frac{\Delta}{(\mu_h - \mu_l)(t_{MA} + t_{NU} - 1)}$$

and

$$(1 - t_{NU})(p_D^{MA})^{\frac{1-r}{r}} + t_{NU}(p_D^{NU})^{\frac{1-r}{r}} = \frac{r\lambda(d - c)}{(1 - \lambda)c}$$

Algebraic manipulation of the mathematical expression of the optimal configurations shown above allows us to obtain the following propositions:

#### Proposition 1:

For a given cost for anti-AV development (fixed  $\Delta$ ),  $p_D^{MA} - p_D^{NU}$  increases in  $t_{MA}$  and  $t_{NU}$  if

$$\Delta \geq (t_{MA} + t_{NU} - 1) \left[ \left( \frac{c(1 - \lambda)(1 - t_{NU})}{(d - c)\lambda r t_{MA}} \right)^{\frac{r}{r-1}} - \left( \frac{c(1 - \lambda)t_{MA}}{(d - c)\lambda r(1 - t_{MA})} \right)^{\frac{r}{r-1}} \right] (\mu_h - \mu_l)$$

and decreases in  $t_{MA}$  and  $t_{NU}$  otherwise.

#### Proposition 2:

For a given selection accuracy (i.e. fixed  $t_{MA}$  and  $t_{NU}$ ),  $p_D^{MA} - p_D^{NU}$  increases in the anti-AV development cost as long as the cost remains lower than

$$(t_{MA} + t_{NU} - 1) \left[ \left( \frac{c(1 - \lambda)(1 - t_{NU})}{(d - c)\lambda r t_{MA}} \right)^{\frac{r}{r-1}} - \left( \frac{c(1 - \lambda)t_{MA}}{(d - c)\lambda r(1 - t_{MA})} \right)^{\frac{r}{r-1}} \right] (\mu_h - \mu_l)$$

and is constant, otherwise.

If the anti-AV technique is relatively costly (high  $\Delta$ ), MA would not choose to develop such techniques, and hence, as the selection accuracy improves, SA designs a less and less stringent classifier for those files pre-screened as normal and a more and more stringent classifier for those files pre-screened as malware. Thus, as conventional wisdom would suggest, the differentiation between the detection rates for the two types of files increases when SA becomes better at discriminating

the two types. However, if the cost of anti-AV techniques is sufficiently low, MA will use anti-AV techniques to beat the selector. In this case SA must deter MA from using anti-AV techniques by making the detection system less stringent for those pre-screened as malware and more stringent for those classified as normal when classification accuracy improves. Thus, SA deters the faking of anti-AV techniques by subjecting malware to a more lenient detection system and normal files to a more stringent detection system.

With further algebraic manipulation of the threshold level of the anti-AV technique cost shown in the propositions above, we obtain the following result:

**Proposition 3:**

*The threshold anti-AV cost required to deter MA from using anti-AV techniques will increase if the selector accuracy increases.*

This proposition implies that with the increase of selector accuracy, MA has more incentive to devise anti-AV techniques. To deter such behaviour from MA, it is important for SA also to increase the cost for devising anti-AV techniques. Therefore, to fully realize the benefits of having a selector to pre-screen incoming files, it is important to make sure it would be sufficiently costly to deter MA from devising anti-AV techniques.

#### 4. CONCLUSION

In this paper, we have constructed a stylized game-theoretic model to analyse the optimal configuration of a heterogeneous AV system with a selector component that pre-screens incoming files. Our model incorporates one crucial aspect of the AV scheme: the strategic behaviour of malware authors who would invest in developing techniques to try to beat the selector component of the AV system. Based on the analysis of our model, we obtain the following implications for the design and configurations of a detection system:

When the cost of anti-AV development for a malware author is sufficiently low, the difference between the optimal detection rates configured for the two classifiers will be decrease with selection accuracy. This implies that when the selector increases its accuracy, it is optimal for a security analyst to maintain a less stringent classifier for malware and a more stringent classifier for normal files.

A security analyst may deter a malware author from developing anti-AV techniques by limiting the difference of the detection rates of the two classifiers. Thus when the anti-AV development cost increases, the detection rates can be more different. Therefore, using a highly sophisticated (i.e. high detection rate) classifier is optimal only if it is coupled with increased anti-AV development costs.

The minimal cost for developing anti-AV techniques that would be sufficient to deter a malware author from developing those techniques increases with the accuracy of the selector. Therefore, the cost of developing such stealth techniques should be considered as an important factor, in addition to the discriminatory power of detection, when designing an AV system.

#### REFERENCES

[1] Ször, P. The Art of Computer Virus Research and Defense. Addison-Wesley, 2005.

[2] Nachenberg, C. Computer virus-antivirus coevolution. *Communications of the ACM*, Volume 40, Issue 1, pp. 46–51, January 1997.

[3] Han, J.; Kamber, M. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2006.

[4] Walenstein, A.; Mathur, R.; Chouchane, M.R.; Lakhotia, A. Normalizing Metamorphic Malware Using Term Rewriting. *Proceedings of the Sixth IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 2006)*, pp.75–84, 2006.

[5] Chouchane, M.R.; Walenstein, A.; Lakhotia, A. Statistical signatures for fast filtering of instruction-substituting metamorphic malware. *Proceedings of the 2007 ACM Workshop on Recurring Malcode (WORM 2007)*, pp. 31–37, 2007.

[6] Egan, J.P. *Signal Detection Theory and ROC Analysis*. Academic Press, 1975.

[7] Osborne, M.J.; Rubinstein, A. *A Course in Game Theory*. MIT Press 1994.