Intrusion Detection Architecture for Distributed Systems using Game Theory Approach

Shahid Naseem¹, Waqar Ali¹, Fahad Ahmad², Qura-Tul-Ein², Tahir Alyas²

¹UCEST, Lahore Leads University, Lahore Pakistan. ²NCBA&E, Lahore, Pakistan
Email: Shahid.naseem@gmail.com

Abstract
Purpose---The purpose of this paper is to develop an “Intrusion Detection Architecture for Distributed Systems using Game Theory Approach” for interaction between nodes (normal or malicious) and IDS as a repeated game.
Methodology--- We inject a small sample of simulated attacks into the network traffic and use the system response to these attacks to define the game structure and utility functions. N players play initially cooperative and then non-cooperative game at each stage of the game, where players of the game are an IDS and nodes.
Findings-- It is found that Intrusion Detection Architecture for distributed systems by using the game theory approach provides the results more than direct optimization methods. It verifies real-world attacks performed on the monitored network by using both Max-Min and Nash equilibrium. The advantage of this approach is not only in its security, but also in better model characteristics in terms of strategy space coverage (unfrequented, but critical attacks are covered).
Research limitations-- Effective policing and a right balance between tradeoffs have always been an issue. Tradeoffs could include IDS sensitivity versus false alarm rates. Traditional intrusion detection systems are not evolved enough for complex detection and response strategy.
Practical Implementation-- Our approach is suited for realistic environments, where typically lack any ground truth information regarding traffic legitimacy/maliciousness and where the significant portion of system inputs may be shaped by the nodes in order to render the system ineffective.

Keywords-- Potential threats, distributed systems, Game theory approach, self-monitoring capability, realistic environment.

1. INTRODUCTION

Intrusion detection systems have long been utilized in detection and response strategy to potential attacks. Specifically in the area of networking to prevent or detect an increasing amount of intrusions, intrusion detection systems (IDSs) have been utilized. An IDS has to not only gather various information, but also process it and apply actions based upon the given policies. Intrusion detection systems (IDSs) monitor various events in a networked system and analyze them for signs of security compromises [1]. By extending the information security paradigm beyond traditional protective (e.g. firewalls) and reactive measures (e.g. virus and malware detection), they increase the ability of the system administrator to control the system, and help him or her better manage its security [2].
To count the increasing number of intrusions into a given system, companies are deploying intrusion detection systems. Each solution has its own advantages and disadvantages. As network-based computer systems play vital roles in modern society. They simply detect all traffic and discern malicious traffic from legitimate used. They have become the target of intrusions by our enemies and criminals. Modern and more evolved intrusion detection systems now have the ability to take responsive or pre-emptive actions. In addition to intrusion prevention techniques, such as user authentication and authorization, encryption and defensive programming, intrusion detection is often used as another wall to protect computer systems. Traditionally, there have been two main classes of intrusion detection systems:

- Host-based Systems
- Network-based systems

2.1 HOST-BASED SYSTEMS

Host-based systems monitor the detailed activity of a particular host in real-time. The system call traces produced by an auditing mechanism such as the Solaris Basic Security Module (BSM) typically provides the IDSs with the data needed to search for attack signatures. When an analysis of the BSM data shows signs of an intrusion, the IDSs alert the system administrator of an attack. Proceeding host-based IDSs was the development of network-based IDSs.

2.2 NETWORK-BASED SYSTEMS

Some network-based systems focus on a single host; most typically monitor a network of computers and other devices (i.e. routers, gateways) that are subject to attacks [3]. Network-based systems, as well as host-based systems, can be further classified by different methods of protection:

- Misuse detection approach
- Anomaly detection approach

2.2.1 MISUSE DETECTION APPROACH

Misuse detection systems simply rely on pattern or signature recognition wherein each pattern or signature represents a malice activity (intrusion). Therefore, each attack is also represented in a form of a pattern or signature. Depicted in fig 1 below is a typical misuse detection system. The misuse detection systems do, however, have many challenges. One such challenge is to controlling false positives. False positives are activities flagged by a system as intrusive when in fact they are legitimate uses. A false positive is usually attained when a signature representing an intrusive activity also matches a non-intrusive activity. A high false positive rate can render the entire intrusion detection system useless as an effect of repeatedly.

![Fig 1: Misuse Detection](image_url)
2.2.2 ANOMALY DETECTION APPROACH

Anomaly detection systems usually detect intrusive behavior by defining all intrusive activities as anomalous. In order to determine what an anomaly is, the system consults a profile engine. The profile engine typically contains a normal activity profile among other things. This normal activity profile contains all legitimate activities; the given system can perform. Thus, all activities are checked with the profile engine to determine, whether they match a certain activity in the normal profile. If they do match an activity within the profile, they are considered to be normal activities and subsequently legitimate use. However, if an activity is detected that does not match an activity within the normal activities profile, it is flagged as an intrusive activity and appropriate measures are taken as specified by the administrator [4]. Activities flagged as false negatives are much more dangerous than those flagged as false positive.

Fig 2: Anomaly Detection

3. TRADEOFFS AND DRAWBACK

Whether one has deployed a misuse detection system or an anomaly detection system, effective and well-balanced policing is paramount to the success of the intrusion detection technique. One of the main tradeoffs is that of convenience. An administrator must balance the security risks with the ease of accessing the system. Decisions in an intrusion detection technology must be made as quickly as possible [5].

4. PROPOSED MODEL

Game theory provides a rich set of tools to study problems with different objectives interact and compete with each other on the same system. Therefore, game theory is a strong candidate to provide the much needed mathematical framework for analysis, modeling, decision, and control processes for information security and intrusion detection. Game theory provides a way of formulating the decision making process of policy establishment and execution. We present an architecture that will integrate the game theory approach into IDS with self-monitoring capability, in order to simulate the worst case, optimally informed nodes and to optimize the system behavior against such nodes, where the interaction between malicious nodes and the IDS is modeled using 2-player static and repeated cooperative and non-cooperative games.

In this setting, the node(s) who attempt to gain unauthorized access to a networked system are represented by one player and the second player is the IDS, which allocates system resources to collect information for detection and decides on a response. The hypothetical attackers with full access to system parameters could dynamically identify the best strategy to play against the system. The game model integrates the preferences and strategies of two players (nodes and IDS). Their strategy sets are defined as a selection of IDS configurations for the IDS (defender) and the selection of a particular attack type (e.g. buffer over flow, password brute force, scan...) for the nodes (attacker). The main difference of the utility functions is the relaxation of the requirement on the identical nodes gain/defender loss and the proportionality of associated costs (alarm processing, monitoring etc.) with the gain/loss value.

The actual utility function values of both players depend principally on the sensitivity of the system using defender’s strategies with respect to individual attacker’s strategies (\(\alpha_{ij}\)), and the associated rate of false positives (\(\beta_i\)) for each configuration. \(\alpha_{ij}\) denotes the probability that the j-th attack strategy is detected by the IDS, when the defender plays...
the i-th defence strategy and $\beta_i$ denotes the probability that the i-th defender’s strategy will result in a false alert [6]. Both players simultaneously select their strategies from the set S and the combination of these strategies determines the payoffs to nodes and IDS, as defined by their respective utility functions by using the Max-Min and Nash equilibrium. There are two existing approaches to integration of the game model with an IDS:

4.1 OFF-LINE INTEGRATION

When the game is defined in design time, solved analytically, using a priori knowledge about expected impacts and success likelihood of the attacks, and the system parameters are fixed to resulting strategies according to game results. Game theory ensures that the system parameters are set to force the adversary into the selection of less damaging (or more rational) strategies. It is sufficient for systems deployed in stable environments, but most IDS need to cope with dynamic environments, where the background traffic and other factors change frequently. In such environments, the static strategies perform poorly.

4.2 ON-LINE INTEGRATION

When the game uses presumed adversary actions in the observed network traffic to define the game is the opposite approach. The game is being defined by the actual actions of real-world nodes executed against the monitored system, elegantly solving the relevance problem. Motivated nodes can easily mislead the IDS by insertion of a sequence of attacks that are orthogonal to its actual plan to target its utility. Indirect Online Integration provides interesting security properties desirable for real-world deployment. The solution uses the concept of challenges to mix a controlled sample of legitimate and adversarial behavior with actually observed network traffic and is a compromise between the above approaches.

Fig 3: Inline variant of Game/IDS Integration

In this model, the real traffic background (including any possible attacks) is processed in conjunction with simulated hypothetical attacks within the system. We measure the system response to these challenges, drawn from the realistic attack classes and use them to estimate the system response to the real-world samples from the same classes. The challenges are then mixed with the real traffic on IDS input and the system response to them is used as an input for game definition, measuring/estimating the current values of $\alpha_{i,j}$ and $\beta_j$. The major advantage is higher robustness w.r.t. strategic attacks on adaptation algorithms, and lower system configuration predictability by the
5. GAME FORMULATION OF THE PROPOSED ARCHITECTURE

Consider a game $G$, which will be called the stage game. Let the players set to be $I = \{1, \ldots, N\}$ and refer to a node’s stage game choices as actions. So each node has an action $A_i$. If it is a malicious node then sometimes its action is dropping of the incoming packets. Let $a^t_i$ refers to the action of the stage game $G$ which node $i$ executes in period $t$. The action profile played in period $t$ is just the $n$-tuple of individuals’ stage game actions $a^t = (a^t_1, \ldots, a^t_n)$. Each node $i$ has a Von Neumann-Morgenstern utility function defined over the outcomes of the stage game $G$, as $u_i: A \rightarrow \mathbb{R}$, where $A$ is the space of action profiles. Let $G$ be played several times and let us award each node a payoff which is the sum of the payoffs it got in each period from playing $G$. Then this sequence of stage games is itself a game, called a repeated game.

We now define the players’ payoff functions for the repeated game. When studying repeated games, we are concerned about a player who receives a payoff in each of many periods. A common assumption is that the player wants to maximize a weighted sum of its per-period payoffs, where it weights in later periods less than earlier periods [8]. Now we need to specify the strategies for each of these players. Each node makes the decision whether to (1) accept a packet and forwards it to improve its own reputation in the network; we call this action “Normal”. (2) do not cooperate and save battery life and stay selfish, we call this action “Malicious”. On the other hand, IDS always wants to catch a malicious node but it depends on how well it can detect an intrusion. Thus the output of IDs actions are either “Catch” a node as malicious or “Miss” it.

![Fig 4: Possible cases of intrusion between IDS and nodes](image)

Fig 4 depicts that in case of false positives and false negatives, payoff of one player is the maximum when it is the minimum for the other player. The most important case (rewarding for IDS) is when a node acts maliciously and IDS is able to catch it. IDS has different utility values based on which we would like to give different weights to false positives and false negatives detections. For simplicity, we assume $U(\text{Miss, Normal}) = v'$, $U(\text{Catch, Normal}) = v''$, $U(\text{Miss, Malicious}) = v'''$ and $U(\text{Catch, Malicious}) = v''''$. We define the utility of IDS as:

$$U_{IDS} = \gamma_1 v''' - \gamma_2 v'' - \gamma_3 v''$$

(1)

Where $\gamma_i$ represents the number of occurrences of case $i$. We consider the different retaliation strategy for IDS. In the initial period every node plays cooperatively and so IDS does not catch anyone, in later periods, IDS does not catch if the node has always played normal. However, if a node acts maliciously, then the IDS catches it for the remainder of the game. More formally, the IDS has the following strategy:

$$S_{IDS} = \begin{cases} 
\text{Miss} & \text{if } t=0 \\
\text{Miss} & \text{if } a^{t-1}_i = \text{Normal} \\
\text{Catch} & \text{otherwise.} 
\end{cases}$$

(2)
Each node in the initial period plays normally and so IDS does not catch anyone, in later periods, a node does not act maliciously if the IDs has missed it. However, if the IDS catches a node, then the node acts maliciously for the remainder of the game. More formally for a node $i$, we have the following strategy:

$$S_i = \begin{cases} 
\text{Normal} & \text{if } t=0 \\
\text{Normal} & \text{if } a_{i}^{t-1} = \text{Miss} \\
\text{Malicious} & \text{otherwise.}
\end{cases} \quad (3)$$

First, we show that the above strategies reach to Nash-equilibrium of the repeated game. Both players (intruder and defender) play cooperatively at $t=0$. Therefore, at $t=1$, the $U(\text{Miss, Normal})$; so they both play cooperatively again. Therefore, at $t=2$, the $U((\text{Miss, Normal}), \text{Miss, Normal})$ and so on. The repeated game payoff to each player corresponding to this path is trivial to calculate.

6. CONCLUSION

In this paper, we acknowledge that IDS, being a misuse detection system detects attacks for which it knows a signature. Detecting malicious activity by way of signature pattern, however, has been a widely supported in antivirus utilities. It decreases the number of false-positives in the detection process, which is very important for performance. Intrusion detection systems have proven their value to network and system security. The intrusion detection architecture for distributed systems using game theory provides the results more than the alternative direct optimization methods, as we have verified on inserted challenges and real-world attacks performed on the monitored network. These methods provide robust performance and reliably converge when using both Max-Min and Nash equilibrium. Infinite repetition can be the key for obtaining behavior in the stage game which could not be equilibrium behavior if the game was played once or a known finite number of times.

REFERENCES


