Π Α Ν Ε Π Ι Σ Τ Η Μ Ι Ο Α Ι Γ Α Ι Ο Υ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Ιωάννα Καντζάβελου

Ανίχνευση Εισβολών στην Ασφάλεια της Τεχνολογίας της Πληροφορίας

(Intrusion Detection in Information Technology Security)

ΔΙΑΤΡΙΒΗ
για την απόκτηση Διδακτορικού Διπλώματος
του Τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων
του Πανεπιστημίου Αιγαίου

Ιούνιος 2011

Π Α Ν Ε Π Ι Σ Τ Η Μ Ι Ο Α Ι Γ Α Ι Ο Υ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΑΤΡΙΒΗ

για την απόκτηση Διδακτορικού Διπλώματος
του Τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων
του Πανεπιστημίου Αιγαίου

Ιωάννας Καντζάβελου

Ανίχνευση Εισβολών στην Ασφάλεια της Τεχνολογίας της Πληροφορίας (Intrusion Detection in Information Technology Security)

Συμβουλευτική Επιτροπή:

Πρόεδρος:

Σπύρος Κοκολάκης
Επίκουρος Καθηγητής
Πανεπιστημίου Αιγαίου

Μέλη:

Σωκράτης Κ. Κάτσικας
Καθηγητής
Πανεπιστημίου Πειραιώς

Διομήδης Σπινέλλης
Καθηγητής
Οικονομικού Πανεπιστημίου Αθηνών

Εξεταστική Επιτροπή:

Πρόεδρος:

Σπύρος Κοκολάκης
Επίκουρος Καθηγητής
Πανεπιστημίου Αιγαίου

Μέλη:

Σωκράτης Κ. Κάτσικας
Καθηγητής
Πανεπιστημίου Πειραιώς

Διομήδης Σπινέλλης
Καθηγητής
Οικονομικού Πανεπιστημίου Αθηνών

Κωνσταντίνος Λαμπρινουδάκης
Επίκουρος Καθηγητής
Πανεπιστημίου Πειραιώς

Χρήστος Ξενάκης
Επίκουρος Καθηγητής
Πανεπιστημίου Πειραιώς

Ιωάννης Μαριάς
Λέκτορας
Οικονομικού Πανεπιστημίου Αθηνών

Γεώργιος Καμπουράκης
Λέκτορας
Πανεπιστημίου Αιγαίου

Intrusion Detection in IT Security

Ioanna Kantzavelou

University of the Aegean, Samos, Greece

Abstract

Intrusion Detection in IT Security

by

Ioanna Kantzavelou

Doctor of Philosophy in Computer Science

University of the Aegean, Samos, Greece

Our aim is to expand Intrusion Detection to the area of Game Theory, to borrow concepts required to study the interactions between users and Intrusion Detection Systems (IDSs). The results offer a better understanding of what directions should be followed by an IDS and what motivations drive a user towards attacking activity. It is an inter-disciplinary research work, between Computer Science and Game Theory that falls in the new emerging area of Algorithmic Game Theory.

We argue that an insider's behavior might be predicted and when it gives characteristics of attacking activity, the system can take the analogous countermeasures by deciding proper strategies to confront the potential attacker. The use of the logit QRE, as a method to calculate the likelihood an action will be chosen, enhances the effectiveness of a classical IDS. This is provided through an implementation scheme that proposes a Detection Mechanism, which operates on an algorithm that combines the output of an IDS with the output of the QRE algorithm in order to ensure more precise system reactions. In another implementation scheme, we propose a game-based ID model that will play the games between users and an IDS in such a way that an IDS will be able to defend the Target System by choosing proper actions to avoid attacks.

Περίληψη

Στόχος μας είναι η επέκταση της Ανίχνευσης Εισβολών προς την περιοχή της Θεωρίας των Παιγνίων, με σκοπό να δανειστούμε τα εννοιολογικά εκείνα στοιχεία που απαιτούνται για τη μελέτη των αλληλεπιδράσεων μεταξύ χρηστών και Συστημάτων Ανίχνευσης Εισβολών (Intrusion Detection Systems - IDSs). Τα αποτελέσματα προσφέρουν καλύτερη κατανόηση ως προς τις κατευθύνσεις που πρέπει να ακολουθηθούν από ένα Σύστημα Ανίχνευσης Εισβολών και ως προς τα κίνητρα που οδηγούν ένα χρήστη προς την επιθετική δραστηριότητα. Πρόκειται για έρευνα μεταξύ δύο διαφορετικών επιστημονικών κλάδων, της Επιστήμης των Υπολογιστών και της Θεωρίας των Παιγνίων, που εμπίπτει στη νεοεμφανιζόμενη περιοχή της Αλγοριθμικής Θεωρίας των Παιγνίων (Algorithmic Game Theory).

Υποστηρίζουμε ότι η συμπεριφορά ενός εσωτερικού επιτιθέμενου μπορεί να προβλεφθεί και όταν αυτή φανερώνει χαρακτηριστικά επιθετικής δραστηριότητας, το σύστημα μπορεί να λάβει τα ανάλογα αντίμετρα, επιλέγοντας κατάλληλες στρατηγικές για την αντιμετώπιση του δυνητικού επιτιθέμενου. Η χρήση του logit QRE, ως μέθοδος για τον υπολογισμό της πιθανότητας να επιλεγεί μία ενέργεια, ενισχύει την αποτελεσματικότητα ενός κλασικού Συστήματος Ανίχνευσης Εισβολών. Αυτό παρέχεται μέσα από ένα σχήμα υλοποίησης που προτείνει ένα Μηχανισμό Ανίχνευσης, ο οποίος λειτουργεί με έναν αλγόριθμο που συνδυάζει την έξοδο ενός Συστήματος Ανίχνευσης Εισβολών με την έξοδο του αλγόριθμου του QRE, έτσι ώστε να διασφαλιστούν περισσότερο ακριβείς ενέργειες αντίδρασης από την πλευρά του συστήματος. Σε ένα άλλο σχήμα υλοποίησης, προτείνουμε ένα μοντέλο Ανίχνευσης Εισβολών βασισμένο στα Παίγνια και τη θεωρία τους, το οποίο θα παίζει τα παίγνια μεταξύ των χρηστών και ενός Συστήματος Ανίχνευσης Εισβολών, με τέτοιο τρόπο, ώστε ένα Σύστημα Ανίχνευσης Εισβολών θα είναι σε θέση να αμυνθεί για την προστασία του συστήματος στόχου (Target System), επιλέγοντας τις κατάλληλες ενέργειες ώστε να αποφύγει επιθέσεις.

To my son Dimitris-Michael,

and to my husband Stavros

Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgments

I am indebted to Sokratis Katsikas for believing in me and my skills to achieve this tough goal. Thank you Sokratis for inspiring my efforts and stimulating my imagination to reach the end, up to the required standard. It is not the first time Sokratis guided me since I first met him in 1989. I am grateful to him for his support all the way through the last two decades in whatever I tried or I gained. He is a guru.

I also want to thank my dissertation committee members. I thank Diomidis Spinellis for giving me the first directions towards this long, difficult task. I learned LaTeX and I saved my time, I became member of the ACM and the IEEE and I became up to date, and finally, I followed his work through these years and I found out how to focus on my goals. I thank Stefanos Gritzalis for his continuous interest and support all these years and for encouraging me in teaching the IT Security course in TEI of Athens.

I am grateful to a very special teacher of mine, Yannis Keklikoglou, who has spent very much time to advice me for all the moves and changes I have made in my life, since 1985 I first met him. He is always so right... I want to thank Vassilis Tsotras for sending me the first bibliographical list as the starting point of my research, and for his guidance and concern on achieving this goal. I also thank Dimitris Gritzalis for encouraging me to improve my skills as a reviewer.

It is a pleasure to thank all those from the Technological Educational Institution of Athens and the University of the Aegean, who supported me in any respect all these years. I have also benefitted greatly from my students for the discussions that have led to new ideas and research directions.

Ultimately, I am really grateful to my son Dimitris-Michael for understanding since his born that what I was doing was extremely time consuming and for my long days and nights at the computer. I also thank him for spending hundreds of hours with his father and his grand parents. I am also grateful to my husband Stavros for being so patient and supporting

me to finish this PhD, for taking care of our son while I was working, and proofreading all the draft papers and the draft of this dissertation. It is also gratefully acknowledged all the support provided by my father, my mother and also by my parents-in-law, especially the last years. I also thank my sister for being a great godmother for my son that allowed me to concentrate on my objectives.

Chapter 1

Introduction

> If we knew what it was we were doing, it would not be called research, would it?
>
> Albert Einstein (1879 - 1955)

The chapter begins with the required background concepts from the main disciplines this dissertation engages, the *Intrusion Detection in IT Security* and the *Theory of Games*. We start with a brief summary for the area of Intrusion Detection. We show how Intrusion Detection Systems (IDSs) classify a system's events, and we employ Euler diagrams to illustrate the reasoning that connects the various classes of events an IDS recognizes and produces as output. In the sequel, we decompose an attack into steps, to clarify terms useful in subsequent discussions. Limitations and problems in Intrusion Detection are also described, with emphasis on *reliability* and *accuracy* problems.

For the Theory of Games, we answer a series of questions in order to provide a concise exposition of the subject, necessary for our research work. We describe games, players, strategies, information, the different types of the Theory of Games, applications, solution concepts, and some of the problems that limit its successes.

Then, we justify the reasons we decided to merge Intrusion Detection with the Theory

of Games, and we present the emerging area of *Algorithmic Game Theory*, which shares the same spirit as the present work. We conclude with the motivation of our decisions, the thesis statement, and finally, we present our results to signify the contributions of this dissertation.

1.1 Intrusion Detection in IT Security

The area of Intrusion Detection (ID)[118, 98, 82] in Information Technology (IT) Security includes monitoring and decision. Intrusion Detection is the monitoring of a system's events and the decision whether an event is *normal* or *abnormal*. The word *normal* defines every event that is consistent with the security policy applied to the system, and the word *abnormal* defines any event that threatens the security status of the system. *Intrusion Detection Systems* (IDSs)[78, 45, 47] are the practical representation of ID. The system an IDS monitors and protects is called *Target System* (TS).

Intrusion Detection Systems (IDSs) have been developed to implement the objectives designated by the area of ID, as summarized in the following list:

- To keep records of every event that takes place on a Target System.
- To detect and prevent an attack before its completion, in *real time*.
- To determine the way a Target System was breached or attacked.
- To identify the person who is responsible for a system's breach or attack.
- To counteract in order to prevent further damage and similar breaches in the future.

Depending on the method used to build the detection part of an IDS, three Intrusion Detection techniques ([110, 177, 129]) have gained favor to date; the *anomaly detection* ([145, 36]), the *misuse detection* ([97, 105, 19]), and the *specification-based detection* ([91,

175, 59]) technique. In the sequel, we outline each of the three ID techniques in turn, discussing also the associated advantages and disadvantages.

Anomaly Detection: Also known as *behavior-based* detection technique. For every subject of the system, a profile is created to reflect his behavioral attitudes regarding the system use. This profile includes several characteristics that give evidence of a certain behavior, related to the corresponding subject. Any event that deviates from the expected behavioral attributes, specified for a subject in his profile, indicates an anomaly and raises an attack alarm. There are a number of reasons this detection technique raises numerous false positive alarms and fails to detect several known attacks. The IDES [111], the first IDS implemented in 1988 by the SRI International, which is based on the seminal model proposed in 1987 by Dorothy Denning [50], initially had employed only the anomaly detection technique. Other IDSs with solely an anomaly detection engine that followed the IDES are the Wisdom & Sense (W&S) system [176] and the Computer Watch tool [52].

Misuse Detection: Also known as *signature-based* detection technique. Attack signatures are recorded in a base, for those attacks which have a known *pattern*, also called *signature*. A series of events are checked against the stored attack signatures, and if they match any of them, then an attack alarm is raised. It is an accurate technique for the included attack signatures, but it fails to detect unknown attacks causing many false negative alarms. IDSs that use the misuse detection technique are the State Transition Analysis Toolkit (STAT) [76], the IDIOT that has incorporated the Colored Petri Automata (CPA) [97], and the Bro [146].

Specification-based Detection: It is the most recent detection technique with the least implemented systems. For every subject of the system (program, module, the system itself) the specifications of its proper and correct functioning are determined.

While monitoring a subject's operation, it is checked whether it is consistent with the corresponding specifications, and if not, then an attack alarm is produced. The inherent implementation difficulties of this technique have prevented its widespread use. The most representative IDS that implements this technique, established later by Ko, Ruschitzka and Levitt [91], is in [90].

Only early implementations of these intrusion detection techniques came along exclusively in IDS products, distinguishing *anomaly-based* ([161]) from *signature-based* ([97]) IDSs. Especially for the anomaly detection and the misuse detection techniques, it was almost the beginning when realized that the disadvantages of the one technique could be covered by the advantages of the other, and vice versa. Therefore, the large majority of products are *hybrid* systems which incorporate both detection techniques. Among the hybrid systems, were the Haystack[168], the MIDAS [160], the Network Security Monitor (NSM) [72], the Next Generation Intrusion Detection System (NIDES) [79] successor of the IDES, and the EMERALD [149] successor of the NIDES.

An additional categorization derives from the source of an IDS's audit records. The *host-based* IDSs [179] receive data from one host, and the detection aims at attacks against this host only. The *network-based* IDSs [178] receive data from the network traffic and examine packets to detect network attacks, like Network Security Monitor (NSM) [72], or Bro [146]. Likewise, the *distributed-based* IDSs ([169, 19]) receive data from different hosts, and probably, from the network as well that connects them.

In Intrusion Detection several tools, methods, and approaches have been used so far. Expert systems ([77, 76]), neural networks [46], Colored Petri Nets [97], graphs [172], rule-based systems ([62, 76, 89, 85]), data mining approaches [99], estimation theory ([17, 18]), soft computing ([38, 1, 2, 173]), alert correlation [96], etc.

Methods and approaches in the direction of generic intrusion detection models have been attempted in the past ([131, 130]), or at least endeavors to enable IDSs to exchange

information, communicate, and cooperate by employing a common architecture, as it is the Common Intrusion Detection Framework (CIDF) [148].

In the past two decades, a number of surveys have reported on the progress, the trends, and the limitations in the area of Intrusion Detection ([81, 118, 82, 98]), in the Intrusion Detection techniques ([110, 177, 129, 36]) and in the Intrusion Detection Systems ([78, 14, 16, 45, 47]) developed on the basis of the proposed approaches and models. They provide valuable information necessary for future research.

1.1.1 Events Classification

The *events* that take place on a system are *attacks* mixed together with *non-attacks*. A proportion of these *attacks* will correctly be *detected* by an IDS, whereas the rest will be *missed*. Likewise, an IDS will characterize some of the *non-attacks* correctly as *discarded*, while for others it will raise *false alarms*. Figure 1.1 depicts this classification task that an IDS accomplishes for a system's events.

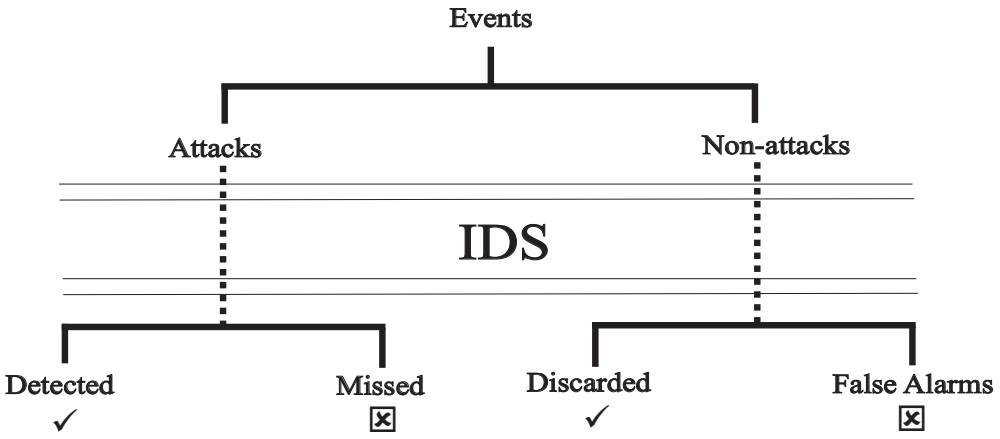

Figure 1.1: Events Classification by an Intrusion Detection System

Considering all the events that occur on a system, we distinguish between *security*

relevant and *security irrelevant events*. Another categorization separates *security relevant events* into *true attacks* and *normal events*. For those events that look as if they were normal but cause damage to the system - although no attack plan has been used (unintentional activity including mistakes) - an additional mixture between *true attacks* and *normal events* is necessary.

1.1.2 The output of an IDS

When an IDS makes decisions examining *security relevant events* only, the output of the classification task previously described in Fig. 1.1 is the set of events *detected as attacks* (Fig. 1.2). It consists of a number of *detected true attacks* (*true positive alarms* - TP) and a number of *false positive alarms* (FP) for those non-attacks detected as attacks. The number of *false negative alarms* (FN), in other words the number of missed real attacks, should be added to complete the set of all the *true attacks* occurred on a system. *False negative alarms* have been derived from *true attacks* plus *normal events* that cause damage. Because the number of *true attacks* are not known for a real world system, the *false negative alarms* are also unspecified. The *detected as attacks* events require further classification, perhaps with human intervention, to distinguish between *detected true attacks* and *false positive alarms*.

The following premises summarize what explained in detail above regarding the output of an IDS.

- Some *Events* are *Security Relevant Events*
- *Security Relevant Events* consist of *True Attacks* and *Normal Events*
- All *Events* that are not *Security Relevant Events* are *Security Irrelevant Events*
- Some *Normal Events* are *True Attacks*
- Some *True Attacks* are *Detected as Attacks by an IDS* (*True Positive Alarms*)

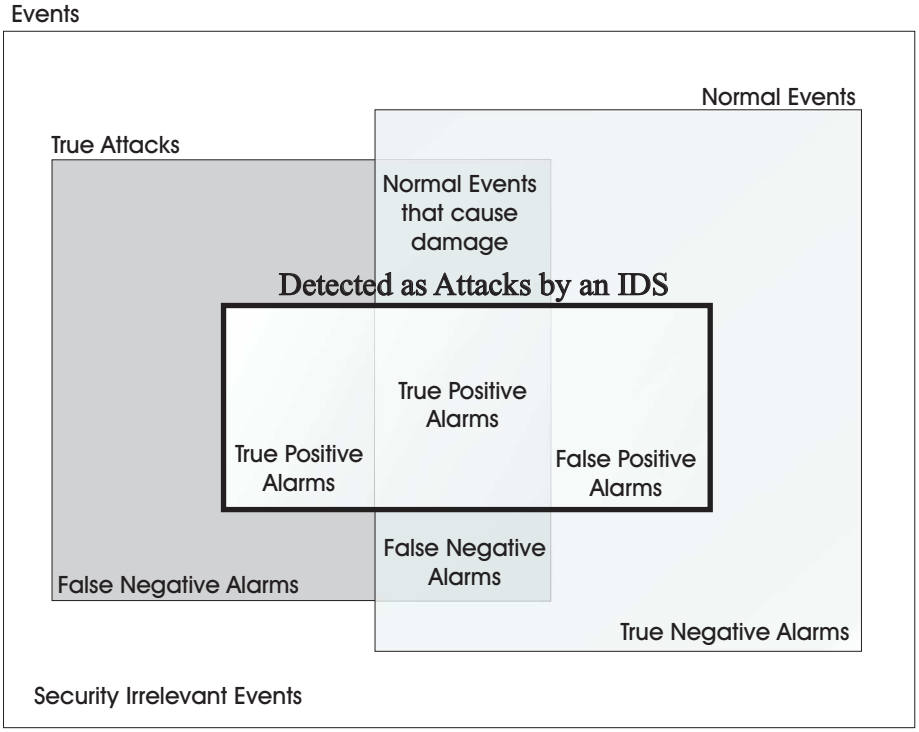

Figure 1.2: Euler diagram for the output of an IDS

- Some *Normal Events* are *Detected as Attacks by an IDS* (*False Positive Alarms*)

- Some *Normal Events* that are *True Attacks* are *Detected as Attacks by an IDS* (*True Positive Alarms*)

We use the Euler diagrams ([57],[163]) as a visual technique to check the validity of these arguments[125]. Figure 1.2 illustrates this syllogistic reasoning behind the blend of events an IDS recognizes and produces as output of its process. Shading has been applied to the rectangles that represent the *true attacks* and the *normal events*, to enhance their overlapping discussed previously, although shading is not a convention in Euler diagrams, as it is in Venn diagrams.

1.1.3 Attack partitioning

It is essential to delineate the meaning of the term *true attack*. An attack consists of many steps. Following a similar partitioning that has been applied in penetration testing[20], we specify that an attack is comprised of three phases. The initial steps of an attack are considered to be the *pre-attack phase*. Next, the steps that carry out and complete the attack form the *actual attack phase*. Finally, at the end of a successful attack, the *post-attack phase* indicates any potential action taken by the attacker, as an attempt to cover his traces, or to bring back the system to its original state. Figure 1.3 draws the described phases of an attack in a sketch that shows their sequential timing.

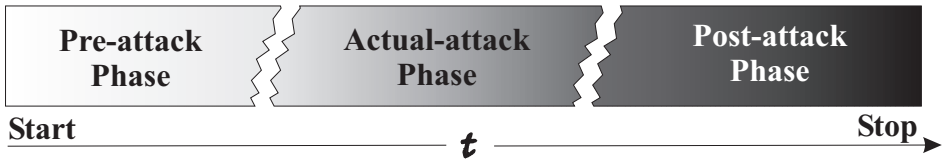

Figure 1.3: Partitioning an attack into phases across time

By referring to an attempt of an attack that does not lead to a successful completed attack, it is implied that only one phase exists, the *pre-attack phase*. It is the *pre-attack phase* that an attacker might intercept information necessary to compromise a system (e.g. to get a backup password file). Because interception is one of the threats a system faces, we assume that even attempts of attacks have bad consequences for a system, and therefore, an IDS must deter them promptly to prevent an attacker from damaging the system. This initial attacker's activity has also been established as *reconnaissance attack*[159].

1.1.4 Limitations and Problems

Intrusion Detection Systems (IDSs) are the practical representation of Intrusion Detection. Therefore, limitations and problems appeared in IDSs reflect corresponding limitations and problems in the area of Intrusion Detection. In the following list, we summarize general limitations in current Intrusion Detection Systems, as presented in [147]:

- *the lack of generic development methodology*, which causes significant high costs;
- *efficiency*, because many IDSs have been designed in a complicated way, in order to detect as many types of attacks as possible, but this generates complexity and increases the runtime overhead of the system;
- *portability*, because most of the IDSs are operating system dependent, and those which have been designed and implemented as platform independent, might be proved inefficient with limited capabilities;
- *upgradability*, because a substantial effort and cost are required to upgrade an existing IDS, mainly due to the lack of a generic development methodology;
- *maintainability*, because IDSs require skillful experts in fields other than security, to configure an IDS, to maintain the rule base of an IDS, or to adjust the statistical metrics of the corresponding component of an IDS; and
- *benchmarking*, because testing an IDS using attack scenarios is a difficult phase of its development, data on IDS benchmarks hardly exist in the literature, and only a few data on the performance of IDSs have been published.

Axelsson [15] also poses some unanswered questions regarding the problems encountered in ID. Moreover, the reasoning described in Section 1.1.2 extends the list of limitations with two significant problems, which derive from the use of Intrusion Detection Systems; the problem of *reliability* [84] and the problem of *accuracy*.

Reliability An IDS is reliable if the number of detected true attacks is equal to the total number of attacks that take place against a system. The reliability of an IDS is defined as the ratio of the number of detected true attacks to the total number of attacks that occurred, and expresses the *detection rate* of an IDS:

$$IDS_{Reliability} = \frac{number\,of\,detected\,true\,attacks}{total\,number\,of\,attacks}$$

This ratio should be as close as possible to 1, with a maximum rate of 1. As long as the ratio departs from 1, the IDS becomes less reliable, because it fails to hit all the attacks. The missed attacks generate the number of *false negative alarms*, also known as *false negatives (FN).*

It is reported [53] that IDSs are not as reliable as required by the frequency of attacks, and thus the number of undetected attacks remains extremely high, up to the 96%. This problem is directly connected with the limitation of efficiency. IDSs become more and more complex to achieve the objective of detecting every attack. Although this seems impractical and causes additional problems, it is a principal requirement for the development of an IDS. The larger the ratio, the more reliable an IDS.

The calculation of an IDS detection rate becomes more complicated, because as mentioned above, the number of *true attacks* are not known for a real world system. It means that systems and system administrators know a proportion of attacks that take place and not all of them. As a result, the reliability of an IDS can be computed at its testing phase using training data, rather than for real cases.

Accuracy An IDS is accurate if it raises a minimum number of false alarms, if not none. The accuracy of an IDS may be expressed as the ratio of the number of detected true attacks to the number of the detected as attacks (see Section 1.1.2):

$$IDS_{Accuracy} = \frac{number\,of\,detected\,true\,attacks}{number\,of\,detected\,as\,attacks}$$

This ratio should be as close as possible to 1, with a maximum rate of 1. The larger the ratio the more accurate an IDS. At the installation time of an IDS, configuration and tuning are required to keep the number of false alarms at a minimum level and make the IDS more accurate [152]. Recent IDSs, however, have failed to minimize this ratio [53], producing significant numbers of alarms for legitimate events detected as attacks. These are the *false positive alarms*, also called *false positives (FP)*. Furthermore, the IDS and consequently the Target System overhead increase when processing irrelevant activities, detecting falsely attacks, and counteracting normal events. These wrong decisions make also an IDS less effective. Debar and Morin [48] state that the accuracy of IDSs is closely related with the success and usability Intrusion Detection technology can reach.

To quantify the described measures of reliability and accuracy, we consider the following example (Example 1) for an IDS that monitors a system.

Example 1. Among a number of 100 attacks that reach successfully a system, its IDS examines an enormous number of events and detects 90 events as attacks. But the detected true attacks are 85, whereas the other 5 are false positive alarms. According to the measures specified above, this IDS has 85% *reliability* (detection rate or True Positive rate - TP) and $85/90 = 0.94\overline{44}$ that is $94.\overline{44}\%$ *accuracy*. Expanding the calculations to cover the reasoning of the arguments established in Section 1.1.2, the False Positive rate is $FP = 5/90 = 0.0\overline{55} = 5.\overline{55}\%$ and the False Negative rate is $FN = 15/100 = 0.15 = 15\%$.

Intrusion Detection Systems, however, aim at detecting as many attacks as possible raising as less false alarms as possible. This is also the requirement of *effectiveness* [15] for the IDSs, which is actually comprised of the reliability and the accuracy requirements.

Unfortunately, the goal of the highest possible detection rate fails to meet simultaneously the goal of the least possible false alarms. As the IDS design becomes more complex to cover as many instances as possible, i.e. to detect a greater number of attacks, the false alarm rates increase rather than decrease, causing worse problems for the Target System. As a result, an additional problem is appended to the list discussed previously in this section, regarding the effectiveness of IDSs. Axelsson examined this problem under the label the *base-rate fallacy* in Intrusion Detection [15], according to which he tights the goal of a high detection rate with the goal of the low false alarm rate and requires both to be achieved in order to have great performance in an IDS. In the next example (Example 2), we reproduce an analogous scenario to present the base-rate fallacy of Intrusion Detection, as addressed by Axelsson [15].

> **Example 2.** During the testing phase with training data, an IDS examines 100 attack events and detects 88 of them as attacks, and when it examines 100 normal events it characterizes 88 of them as normal and 12 as attacks. Consequently, this IDS has 88% confidence (certainty) when it decides whether an event is attack or normal.
>
> Under real circumstances, the same IDS examines a random event and concludes that it is an attack. Presuming that the Target System of this case has been appropriately protected by a number of security mechanisms, only one attack event appears every 10000 event records, i.e. the attack rate is $1/10000 = 10^{-4}$. Then, what is the probability this positive alarm to be true?
>
> To answer this question, first we denote an attack as A, and a normal event as $\neg A$ (not A). Let D be a detected event as an attack (alarm) and $\neg D$ an undetected event (no alarm). Then, given the above information, the probability this IDS to detect an attack event is $P(D \mid A) = 88\% = 0.88$, the probability a normal event to remain undetected is $P(\neg D \mid \neg A) = 88\% = 0.88$, and the probability an event to be an attack is $P(A) = 10^{-4}$. To measure the validity of a positive

alarm, that is, the probability $P(A \mid D)$ of a detected event to be a true attack, we will apply the Bayes' theorem and calculate the following:

$$P(A \mid D) = \frac{P(A) \cdot P(D \mid A)}{P(A) \cdot P(D \mid A) + P(\neg A) \cdot P(D \mid \neg A)} \tag{1.1}$$

We substitute the values given above and calculate the $P(D \mid \neg A)$ which is the $1 - P(\neg D \mid \neg A)$ and equals to $1 - 88\% = 0.12\%$. Then, Equation (1.1) gives:

$$P(A \mid D) = \frac{10^{-4} \cdot 0.88}{10^{-4} \cdot 0.88 + (1 - 10^{-4}) \cdot 0.12} = \frac{0.000088}{0.120076} = 0.000732869 \simeq 0.073\% \tag{1.2}$$

Therefore, the probability a positive alarm to be true is surprisingly low, only 0.073%, although the IDS's confidence is 88%. This is caused by the fact that in this scenario attacks are rare enough, only one attack event appears every 10000 event records. In addition, from Equation (1.2) we conclude that there is an analogy between the factor that controls the $P(D \mid A)$ (the detection rate) with the corresponding that controls the $P(D \mid \neg A)$ (the false alarm rate). As a consequence, in order an IDS to be effective, it should have both, a high detection rate and a low false alarm rate.

In spite of the above shortcomings, the majority of organizations have realized that Intrusion Detection has a central role in their IT security. Thus, Intrusion Detection should be on the market. Governments and firms have an interest in computer science and would like to know how much money to invest and in which area. Is Intrusion Detection an area that is worth of funds, which will return analogous security solutions? Then, the point-blank question generated is why Intrusion Detection commercial solutions are limited.

Unfortunately, Intrusion Detection Systems fail to balance to an acceptable level the ratios between detected real attacks and false positives and false negatives respectively, and

to overcome the limitations discussed previously. The commercial world is interested in the reduction of false alarms for a number of reasons, but also because these increase the system runtime overhead. If IDSs succeed commercially, then they can reduce the recovery cost of attacked systems in the industry. Therefore, the research question motivated by these problems is how can we improve Intrusion Detection in IT security, by optimizing IDSs and making them less problematic commercial systems. Consequently, there is a need for new types of Intrusion Detection approaches in IT security, which will employ tools and techniques from other established fields, to address its problems and overcome its limitations.

In the next section, we investigate the *Theory of Games*, a discipline that shares many commonalities with the area of Intrusion Detection, and in Section 1.5 we justify the reasons we strove our research work towards this discipline.

1.2 The Theory of Games

> To be literate in modern age, you need to have a general understanding of Game Theory.
>
> Paul Samuelson, 1991

Games are played between persons, or more generally, between organizations or firms controlled by persons, in cases the latter interact. In these interactive situations a participant of the game, called *player*, is interested in predicting others' future actions, and also, in how others interpret his own actions. Camerer describes a game as *a mathematical x-ray of the crucial features of these situations* [33], while others describe it as a set of specifications of what actions players are able to take, but not what actions players do take [135].

Some games are played once, the so called *one-shot games*, while most are played repeatedly, finite or infinite number of times, specified as *repeated games*. In repeated games a base

game, the *stage game*, is reiterated. Players organize their actions in *strategies* considering also other players' actions, information and beliefs.

Osborne and Rubinstein [135] set three different *dimensions* on which three divisions of game theoretic models are based, as described in the sequel:

- The first is the *player*. If a player is an individual, then the type of game theoretic model is *non-cooperative*, whereas, if a player is a group of individuals, the type of game theoretic model is *cooperative*.

- The second is the plan of *actions* a player chooses. It is a two folded dimension that includes both *action and time*, because action has also the notion of move. If a player chooses his plan of actions once at the beginning of the game and he is not aware of the plan of actions of any other player, then this is a model of a *strategic game* and decisions are made *simultaneously* for all players. At the opposite side of this model type, an *extensive game* allows each player to think about his plan of actions whenever he plays, formulating interactions with *sequential* moves.

- The third is *information*. When the players of a game are fully informed about each others' moves, then the model of the game is defined as a game with *perfect information*. On the contrary, when players are not very well informed, then the game is characterized as a game with *imperfect information*.

In the following paragraphs, we provide an overview of the Theory of Games by answering a series of questions, directly related to most of the game theoretic aspects discussed in this thesis, which are not familiar to an IT Security specialist reader.

▷ *What is the history of Game Theory?*

In order to present the Theory of Games as a very old piece, in 1985 Aumann and Maschler [13] used an early example by formulating and solving a marriage contract problem,

which was defined as a coalitional game by the Babylonian Talmud, about 2000 years ago, anticipating the modern theory of cooperative games. The Theory of Games effectively serves the Theory of Economics since its start.

A brief but comprehensive historical synopsis of Game Theory is provided in [167]. It starts with Leibniz[1], probably the first who expressed in 1704 the idea that later motivated the Theory of Games. Again before 1900, philosophers, like Hobbes, Hume, Rousseau, and Smith, gave game theoretic descriptions when thinking of social interactions. Zermelo (1913), Borel (1921 and 1934), and von Neumann (1928) at his early attempts, they all gave mathematical analyses to some problems that fall in the area of Game Theory. Finally, in 1944, the Theory of Games appeared as a new field of study, when John von Neumann and Oscar Morgenstern published their work in the book *Theory of Games and Economic Behavior* [180]. They believed that people have the same reasoning whether they play a game, like chess, or they get involved in any social interaction.

The contributions in the Theory of Games led to Nobel prizes in Economic Sciences to be awarded. Among them, the Nobel prize 1994 in Economic Sciences was awarded jointly to John C. Harsanyi, John F. Nash Jr. and Reinhard Selten *for their pioneering analysis of equilibria in the theory of non-cooperative games* [133]. A few years later, the Nobel prize 2005 in Economic Sciences was awarded jointly to Robert J. Aumann and Thomas C. Schelling *for having enhanced our understanding of conflict and cooperation through game-theory analysis* [134]. Sylvia Nasar has written a biography of John Nash, also a great reference for the study of the Theory of Games [127].

▷ *What is the Theory of Games?*

Under various circumstances, people interact with others in several ways. Each interaction implies that while a person is thinking about a situation, another person is also

[1]Leibniz accepted that people play games and spend time and effort thinking about which strategy to choose, and therefore he concluded, philosophers should examine players' reasoning when they play a game.

thinking about it, and both of them aim at deciding what action to take to deal with this situation. Dixit and Skeath define that *Game Theory is the analysis or science of such interactive decision making* [51]. But, because people think carefully before deciding upon an action, i.e. they are aware of their objectives or preferences, they clearly specify limitations and constraints, and they set criteria before they choose, it is said that people are *behaving rationally.* For that reason, Dixit and Skeath continue that *Game Theory is the science of rational behavior in interactive situations.*

According to Skyrms and Vanderschraaf *Game Theory is that branch of decision theory which deals with the case in which decision problems interact* [167]. They also argue that what von Neumann and Morgestern conceived of as a scientific theory for social interactions is far from what has been formulated until now, because it is a much more general theory that extends over many other disciplines and expands continuously and rapidly.

▷ *What the Theory of Games is good for?*

When asking professionals whether the Theory of Games is able to predict what people do, or if it is suitable for advice to players, their answer is none of these, but it is a set of answers to mathematical questions regarding what players with ranging rationality will do in the future [33].

▷ *What are the types of the Theory of Games?*

In Game Theory, the construction of a game is the first step toward answering the following question; how the players choose their strategies in real play? There are three different approaches to address this question, which distinguish three different types of game theory; *normative*, *descriptive*, and *prescriptive* [42]. In *normative* game theory, one examines the consequences of choices made by super-rational players. *Descriptive* game theory usually requires experimentation, to examine how players actually play, and the strategies they

choose in reality. Social science and psychology are involved in this examination. When one engages in *prescriptive* game theory, one examines theoretically a constructed game, to determine how players should play it, and to recommend strategies. As a result, it is possible to give advice that helps players to make better decisions. A triumph of *prescriptive* game theory is in the design of auctions. Game theorists have successfully designed auctions for among other things, radio-wave frequencies [122] and cellular telephony [22]. The three approaches overlap, for example when we ask another question: will the players choose a recommended strategy or not?

▷ *Where the Theory of Games is applicable?*

The Theory of Games can be applied in any discipline except those dealing with completely inactive objects [51]. Many disciplines have incorporated game theoretic techniques, including Economics, Law, Biology, Psychology and Political Philosophy. It has also been used in agency models, in models of search, and in studies of price formation under various institutional circumstances [95]. Recently, Game Theory has been applied in Computer Science, and the derived results present a promising match between the two disciplines (see Sections 1.3 and 2.1).

▷ *What is a strategy?*

A *strategy* of a game is a specification of how to play the game taking into account any event that might occur [101]. It is a player's option, an alternative among those available for him to choose.

▷ *What is a game in normal form?*

Normal form games are more suitable for two-player games. They are depicted in matrices, in which rows represent the actions taken by one player and columns represent

the actions taken by the other player. For each cell there is a pair of numbers, the *payoffs* of the two players. *Normal form games are descriptions of games that specify the strategies and the payoffs*, i.e. *a map from strategy profiles to payoffs*, also called *strategic form* games [101]. It is important that in *normal form* games players choose *independently*, without observing each others' strategies or influence another player's strategy, and thus, players play as if they were choosing *simultaneously* ([167, 95]).

▷ *Is there a difference between a strategic game and a decision problem?*

In *decision problems* each player is only concerned about his own actions, but in *strategic games* each player is also concerned about the actions taken by the other players. Osborne and Rubinstein [135] refer to an action profile $a = (a_j)_{j\epsilon N}$, one action for each player, as an outcome, and denote the set $x_{j\epsilon N}A_j$ of outcomes by A. Based on this, they clarify that what distinguishes a decision problem from a strategic game is the requirement that each player i's preferences are defined over A instead of A_i.

▷ *What is utility?*

The notion of utility has conceptual and practical difficulties, especially when someone attempts to describe it as a number[180]. When the utility is numerical, then it is the *payoff* at every outcome of the game that reflects players' preferences for the various outcomes of the game [167].

▷ *What is a game in extensive form?*

Extensive form games are more suitable to model dynamic interactions. Kreps [95] describes that they are depicted in trees with at least an arrow that points out from each node and at most one arrow that points at each node. In addition, when tracing the tree using backward induction, the initial node should be reached only once, that is, it should be the end of the back tracing and should not be included in circles.

▷ *How can we solve a game?*

Solving a game is an old philosophical attempt to answer the question: Why rational people choose a specific behavior when interacting with others? Hobbes and Hume are among those social philosophers who contributed towards an answer. While John Nash presented the equilibrium as a solution of a game, centuries before Hume simply referred to an equilibrium play between interacting individuals. He argued that *individuals are rational to follow certain conventions, provided that they expect others to follow these conventions* [167].

A Nash equilibrium of a game is a set of players' decisions that results in an outcome, such that, no player has any reason to deviate from his choices, given that all the players do the same. John Nash proved that every noncooperative game has at least one Nash equilibrium (NE) [128, 74]. In games with more than one NE, the problem of multiple NE of which one to choose appears (discussed in the subsequent question on the problems of Game Theory). In noncooperative game theory, the NE is the most commonly used solution concept.

Other solution concepts are the backward induction, the subgame perfect NE, the Bayesian NE, the Perfect Bayesian NE, the Trembling hand, the Correlated equilibrium, the Sequential equilibrium, the Pareto efficient, the Quantal Response Equilibrium (QRE) (see Chapter 7), etc. In cooperative Game Theory, the Shapley value and the core are the two most commonly used solution concepts.

▷ *What are the problems of Game Theory?*

Studying the Theory of Games as a tool to model economic phenomena, Kreps [95] summarizes four main problems that unfortunately weaken a number of strengths associated with them. The *first* problem is the requirement for clear and distinct rules of a game. The *second* is the problem of multiple Nash equilibria. In games with many Nash equilibria, Game Theory has no systematic method to check whether any one is the actual solution

of the game, and if so, to indicate which one. The backward induction, although used for games with multiple NE, is not sufficient to solve this problem. The *third* problem of Game Theory is called *equilibrium refinements* and derives from the second problem. To address the problem of multiple NE, one of the approaches is to choose among them but having a more sound notion of equilibrium. According to this, NE that reveal unbelievable choices should not be selected. Finally, the *fourth* problem is located in the rules of a game. As Kreps states, game theoretic analyses take the rules of the game seriously into account without knowing their origin. Moreover, the rules might be influenced by the outcomes of the game and this is not under consideration either.

1.3 The Merge: Intrusion Detection and the Theory of Games

The Internet, which deservedly represents our digital world, includes interactions with conflicting interests and strategies planned to maximize profits between involved parties. Others are competitors and others agree on a policy of cooperation. Single users in front of stand alone workstations have irreversibly passed. To handle the special requirements caused by these new circumstances, it seems that we need to borrow tools from other disciplines, like Game Theory that acts as a set of tools for interactive situations, and Economics that analyzes diverse interests in the product lines of goods or services.

In the past several years, Game Theory, Computer Science, and Economics intersected to create a new field, *Algorithmic Game Theory* [132]. But Algorithmic Game Theory has not only emerged as a simple interdisciplinary application where Computer Science benefits from the Theory of Games or Economics. It has drawn new directions of thinking and new fundamental conceptions from different perspectives. The reasons lie in this mixture itself that generates new questions. Computer Science questions players rationality and makes Economics to abandon the assumption that people are fully rational and to examine classical situations under bounded rationality. When Computer Science meets the Theory of

Games, then new computational problems arise in the presence of algorithmic complexities. Likewise, classical game theoretic concepts, such as the Nash Equilibrium, are proved infeasible to be computed even in two player games [43, 44]. Such considerations establish new lines of research and reasoning that hopefully will significantly expand the collaborative areas and their influences.

In a similar way with the one applied in the previous section, we question the area of Intrusion Detection as if it were a game, to discover the most fundamental aspects that would direct the construction of such a game.

▷ *Is Intrusion Detection a strategic game or a decision for action situation?*

Dixit and Skeath explain that strategic games are *interactions between mutually aware players*, whereas, decisions are *action situations where each person can choose without concern for reaction or response from others* [51]. Intrusion Detection is an interactive situation in which mutual awareness of the cross-effects of actions is taking place. An Intrusion Detection System is affected by attackers' actions directly, and takes the proper countermeasures to prevent further damage of the Target System. An attacker is also affected by IDSs' actions, when his purpose is to carry out a successful undetected attack. Therefore, Intrusion Detection in IT Security is a strategic game and not a one person decision problem.

▷ *Is the Intrusion Detection game being played once or repeatedly and with the same or changing opponents?*

An Intrusion Detection game has an established IDS player, which operates in a certain way and starts playing a unique game with each user, at the time the latter enters the Target System. If a user penetrates the TS and uses it only once, then the game with the IDS is a *one-shot game*. At the opposite side, a user who uses the TS often or just more than once, plays a game *repeatedly*, and he is worried about the consequences and

implications on other future games he might have to play with the same IDS. The IDS, as a constant player, plays different games with changing opponents, probably with some similarities shared among them.

▷ *Is the Intrusion Detection game a zero-sum game or not?*

When the players of a game have interests in total conflict, then the game is a *zero-sum* game. But, when the players' interests have some commonalities, then they might both benefit from their interactions. Examining the preferences of IDSs and attackers respectively, one can lead to the conclusion that an Intrusion Detection game is a *zero-sum* game, because when an attacker carries out an undetected successful attack, which is the worst preference for an IDS, then the attacker wins and the IDS loses.

However, there are other instances where both win something while they lose something else, as when an IDS detects a successful attack. In that case, the attacker has achieved to attack the Target System successfully, although he has been detected, and the IDS has detected the attack but after its completion. In such a situation, the game is a *constant-sum* game, but the constant is not zero. Since it is possible a game to be *zero-sum* in the *short run*, but have scope for mutual benefit in the *long run* [51], it seems that the Intrusion Detection game is a *constant-sum* game that some times turns to a *zero-sum* game.

1.4 Motivation

Considering a user who is using a system, the user's activity is either legitimate or illegal. We can define the terms *legitimate activity* and *illegal activity* as follows.

Definition 1. A *legitimate activity* is an activity which preserves all the three principles of Information Technology Security, that is, Confidentiality, Integrity, and Availability.

Definition 2. An *illegal activity* is an activity which violates at least one of the three

principles of Information Technology Security, that is, either the Confidentiality, or the Integrity, or the Availability.

Consequently, it seems reasonable that an IDS, which is able to distinguish between an illegal and a legitimate activity, is an efficient IDS that can successfully detect attacks. So, the problem of intrusion detection turns into the problem of characterizing an activity of a user, either legitimate or illegal in the sense described above. But, this is the problem itself in the intrusion detection literature. Then, why we should adopt a different approach to solve the problem, and leave behind the three well known approaches (see Section 1.1), that have gained favor in recent years? Will a new approach solve the problem more effectively with a higher detection rate and less false alarms?

Detecting intrusions by characterizing an event as legitimate or illegal respectively is a posterior (ex post) detection, helpful only to avoid the attack's consequences and further damage to the Target System. Unfortunately, detecting this way is not so straightforward, mainly because a large number of false alarms degrades the IDS's detection rate (see Section 1.1.4). So, there is a drawback embodied in the traditional approaches.

The answer to the previous questions is that, we do not know the type of user who is acting illegally or legitimately, to conclude what might happen in the near future inside the Target System. We care about real-time detection in order to prevent any damage, because a user who is acting legitimately at a point of time, might act illegally the next moment, if he is an attacker. Also, a user who is acting legitimately for a while, suddenly acts illegally by accident, although he is a normal user. We comment on its possible modeling in Chapter 9 as a future research work. These cases are the main sources of false alarms.

To achieve this goal, we are interested in the type of source of any activity, legitimate or illegal, and not only in distinguishing activities between illegal and legitimate. This objective differs significantly from the corresponding of the anomaly detection technique, which is also a user-centric technique, like the proposed one. The anomaly detection technique

works with user profiles to detect deviations from normal activity, i.e. to detect masquerades, but is not able to infer what is the source of this anomaly, so it detects a normal user who is acting differently one day as an attacker, raising a false positive alarm. That is why the anomaly detection technique goes together with the misuse detection technique. The latter one merely detects attacks based on the corresponding signatures, that is, it distinguishes between known illegal activity that derives from an attacker and legitimate activity wherever it derives from.

Concluding, it is not always a normal user who is acting legitimately. A normal user is also acting illegally when he makes mistakes. Similarly, an attacker is not only acting illegally. In cases of bluffing, he is acting legitimately to confuse his opponent. As a result, the IDS does not know for sure the type of user interacting with it, to prevent any future illegal activity. For this reason, we need to model the user behavior to predict that an attack might happen in the near future, so that we can counteract to prevent it. Anomaly and misuse detection techniques detect intrusions as they happen and not just before.

1.5 Thesis Statement

Some of the research questions we initially stated are: Can we entirely solve any of the main problems of Intrusion Detection? Our discussion in Section 1.1.4 gives a negative answer because some of them are unsolvable (e.g. detection rate vs false alarms). Then, what can we do? We can try new approaches to extend and improve the field of Intrusion Detection and mitigate some of its limitations and problems. Therefore, we have decided to employ the Theory of Games and we justify our choice as explained below.

As IT is a human-computer interactive situation, Intrusion Detection in IT security is also an interactive situation. It is an interaction between a user (potential attacker) and the Intrusion Detection System (IDS) designed and implemented to make a Target System (TS) secure. The IDS is an active player, whereas the TS is a passive part of the process. The

discipline that studies interactive situations is the Theory of Games. Intrusion Detection in IT Security is a field that has the features of a game.

The similarities between the two fields of Intrusion Detection and Game Theory, deserve further study and research. In the past several years, interdisciplinary research between Computer Science, Game Theory, and Economic Theory has given directions to a new exciting area, *Algorithmic Game Theory* [132] (see Section 1.3). Our research work belongs to the field of Algorithmic Game Theory and addresses problems of Intrusion Detection in IT Security. It is based on answering the research questions stated in the following two thesis hypotheses:

Hypothesis 1: Suppose there is an IDS that decides whether a user's activity is malicious or not, with absolute certainty, i.e. 100% detection rate and 0% false alarms.

Question 1: What are the optimal (best) strategies the IDS should follow to counteract detected attacks?

Hypothesis 2: Suppose there is an IDS that decides accurately with probability < 1 whether a user's activity is malicious or not, i.e. with $< 100\%$ certainty, $< 100\%$ detection rate and $> 0\%$ false alarms.

Question 2: What are the optimal (best) strategies the IDS should follow to counteract detected attacks?

Our work belongs in the non-cooperative Game Theory and has incorporated the commonalities of the two types of Game Theory [141], the *descriptive* and the *prescriptive* as described in Section 1.2, by first constructing and solving a game, and then by proposing a detection mechanism as an exploitation vehicle of our results. Research reported in the present thesis is closer to the *prescriptive* than to the *descriptive* approach, since it aims at predicting attacker's potential deviations from the equilibrium path. Details on our results, contributions and their significance are given in the next section.

1.6 Results, Contributions, and Significance

Our first contribution is that we model *Intrusion Detection* as a *generic game*. Following a systematic way, we explore and specify, in a generic form, the players, their actions, the information each player has when he acts, the expected outcomes, and the players' preferences over these outcomes. The general formal description of the ID game and its formal definitions are determined to serve as the underlying foundation for the succeeded results. The repeated ID game model is constructed as well, in order to study real cases in which players play the game again and again. The key-concept behind the generic ID game model is to release ID from any platform, system dependency and detection technique and focus on the heart of the interactions that take place between players.

We subsequently concentrate in the insider threat, and we specifically model the ID game for *insiders* only. In this game, we define *players' preferences*, and so the corresponding *payoffs*, using the *von Neumann-Morgenstern utility function*, by employing an established method rather than defining arbitrary payoffs. This gives the flexibility for important interpretations of players' actions, responses, and beliefs, when studying various case scenarios.

We also provide another formulation of the ID game as a signaling game. The construction of such a game requires the same elements to be specified, but the game has a different method to be examined and solved. Our results show that the problem od multiple NE and which one to choose appears and prohibits us from giving clear conclusions on how the game will be played in the future.

Any attempt to solve the repeated ID game for insiders was also precluded by the known problem of multiple Nash equilibria (see second problem in Section 1.2). Therefore, it was difficult, if not impossible, to conclude on the strategy a player would choose, in response to an opponent's strategy. This forced our research toward other game theoretic solution concepts, beyond the classical Nash equilibrium. The *Quantal Response Equilibrium (QRE)*

is used in the repeated form of the ID game for insiders, and our results show that this achieves a *behavioral prediction* for future players' actions. The significance of this choice is stemmed from the fact that we can detect attempts of attacks if we look at users' intentions, plans, and possible strategies, instead of detecting nearly finished successful attacks by examining signals hidden in billions of event records.

This behavioral prediction can act as an assistant in Intrusion Detection. Instead of trying one more method to detect attempts of attacks, as those that aim at identifying signals received from a target system, we strive toward different directions outside the classical intrusion detection techniques discussed in Section 1.1. The main concept of the behavioral prediction is the intentions that drive a user's plans when he uses a system, and the diversity of these intentions, especially when the user is a potential internal attacker.

Finally, we present *two implementation schemes* suitable for the application of the proposed model, and we design in detail *an algorithm* for the one of these schemes. For the second implementation scheme, a complete Intrusion Detection model with a game-based detection engine is introduced, to demonstrate a new architectural approach that can be applied in ID, a *game theoretic ID architecture*.

1.7 Outline of the Dissertation

In Chapter 2, we review the literature linked with all topics related to our research problem. We start with a summary for the conjunction of the Theory of Games with the Computer Science and with the IT Security. The main part of this review follows and is devoted to others' work in Intrusion Detection with the use of Game Theory. Because the proposed approach is based on the intentions a user has, we conclude with a review on intention-based detection approaches.

In Chapter 3, we construct a generic game model that represents the area of Intrusion Detection. Therefore, a number of elements are defined, such as, the players and their

actions, information players possess when they act, outcomes of the game, and players' preferences over these outcomes. Then, we check whether this ID game is a finite, or a lose-lose game, and if it includes sequential or simultaneous moves, or both. The general formal description of the game follows, and then we check if the game adheres to the rules that define an extensive form game. We close this chapter with the formal definitions of Intrusion Detection as a game with *perfect* and *imperfect* information.

In Chapter 4, we formulate the ID game described in Chapter 3 as a repeated game. We examine two types of repetition, with perfect monitoring and with imperfect monitoring. We first establish the stage game, which is played repeatedly, with its players, their pure and mixed actions, and the action profiles. To cover the different kinds of games an IDS plays with different users, the preferences are separately discussed for normal users, for attackers, and for IDSs, and the corresponding utility functions are defined. Then, the repeated ID game model is explained when players are fully informed about other players' moves (perfect monitoring) and when players are not fully informed about other players' moves (imperfect monitoring). These two cases match the thesis hypotheses stated in Section 1.5.

In Chapter 5, the ID game is specifically constructed for insiders. For this special class of users, a part of this chapter is devoted to clarify how they act, what they think or believe before they act, how risky might be, and what elimination methods have been used in the past against their attacks. The ID game is reconstructed to be played with an insider, using four specific actions defined for insiders. The repeated form of the game is also considered to identify how long and complex it becomes when repeated infinitely. The ID game with an insider is then defined under perfect and imperfect information. The solution of the stage game gives answers on how the game can be played and what actions the players might choose. But the most interesting is the discussion on repeating the ID game with an insider, the folk theorem, the grim strategies, and different scenarios with their solutions, which aim at covering most of the cases that can be realized. In closing, the game is played with an

unconventional insider, who has different unusual preferences, to discover substantial or not differences with other similar games.

In Chapter 6, we construct the ID game as a signaling game and discuss possible Nash equilibria that solve the game. In particular, we apply the domination criterion, and we compute the game equilibria by locating NE in pure and mixed (behavioral) strategies. Our conclusions include the multiple NE problem that generates uncertainty and the need for new signals that will give more understanding in playing the ID signaling game.

In Chapter 7, we introduce and justify the use of another solution concept beyond the common Nash Equilibrium (NE), the Quantal Response Equilibrium (QRE). Solving the ID game with an insider in its repeated form with the use of QRE, we interpret the corresponding results in order to predict insider's future behavior, and we oppose them to the NE results.

In Chapter 8, two implementation schemes are proposed as appropriate to exploit all the results derived from our research work. According to the first one, a new game-based ID model is presented and its architecture is described to a great extent. The functionality of this model is totally different from other intrusion detection techniques presented in other approaches in the past. The second implementation scheme includes a Detection Mechanism that will jointly work with a classical IDS to improve its effectiveness with the QRE calculations. A detailed algorithm prescribes how this collaboration will be successful.

In Chapter 9, we describe our concluding remarks and related open questions that give the motivation for new directions of future works.

Part of the results of Chapters 3, 5, 7, and 8 have been published in joint works with Sokratis Katsikas [86, 87, 88].

Finally, the bibliography list has adopted the citation conventions described in the 2009-2010 citation guide ([71]), which is based on *The Chicago Manual of Style*, 15th ed.

1.8 Summary

Intrusion Detection has unsolvable problems when addressed by signal identification approaches. Nevertheless, the commonalities that shares with the Theory of Games, which is suitable for interactive situations, allows research directions for ID in the new area of Algorithmic Game Theory. Therefore, we decided to improve Intrusion Detection by examining attackers' intentions for future actions when playing with IDSs.

Chapter 2

Literature Review

> There is only one good, knowledge, and one evil, ignorance.
>
> Socrates (469 BC - 399 BC), from Diogenes Laertius, Lives of Eminent Philosophers

In this chapter we describe what we studied for the Theory of Games and its gradual inclusion first in Computer Science, then in IT Security, and finally in Intrusion Detection. In Section 2.1 we cover issues originated when Game Theory met Theoretical Computer Science. In Section 2.2 we specifically examine game theoretic approaches that have been applied in IT Security. The heart of the chapter is Section 2.3, where we present results derived from the survey we have carried out to reveal all related works that fall in the intersection between Game Theory and Intrusion Detection. Because our approach points towards behavioral detection of intruders and especially users' intentions for future actions, in the last section, Section 2.4, we examine research works that use intention-based approaches in ID.

2.1 Game Theory in Computer Science

Or, Computer Science in Game Theory. Which discipline did actually first need the other? Our study has identified both directions of reach, and surveys are conducted by game theorists and by computer scientists as well. Computer Science includes interactive situations and requires Game Theory to model them and solve related problems. Likewise, Game Theory faces an increasing number of computational problems in the digital world and expects Computer Science to provide solutions.

Nathan Linial [104] attempted in 1994 the first systematic study in the interface between Game Theory and Theoretical Computer Science. He mainly focused on protocols and he considered a number of outstanding issues and problems generated when the two disciplines come together.

Two years later, Ehud Kalai [83] recognized that Computer Science, Game Theory, and Operations Research have scientific interactions with significant implications in several fields of applications. In order to open new lines of research, he described a few examples, as the graphs in games, the multi-person operations research, the complexity of playing a game, the complexity of solving a game, and the modeling of bounded rational players.

Christos Papadimitriou [142], among the pioneers of Algorithmic Game Theory, came early across with the challenges and opportunities that Game Theory and Mathematical Economics can provide in Theoretical Computer Science and especially in the Internet, as tools. Yoav Shoham [165] found also that the Internet is another reason to merge Computer Science and Game Theory, because in its present form requires special system design for multiple different entities with conflicting interests that continuously interact to cover a large number of economic activities.

Joseph Halpern [64] had focused on the overlap between Computer Science and Game Theory at an early stage, from the standpoint of a computer scientist, although he is a mathematician. In a work for distributed computing, he concluded that a set of issues,

as part of the commonalities between the two disciplines, has to be really addressed by game theorists, including *fault tolerance*, the *representation of knowledge and uncertainty*, and the *difficulty in designing large mechanisms and games*. In that way, Game Theory can be expanded. But also suggests computer scientists to change the concept they design distributed protocols, to cover the game theoretic aspects involved, for example, when designing Internet agents. In closing, he underlines another significant area of commonality in which game theorists and computer scientists have to be concentrated; to change the way games are represented and find a more compact representation mode.

The first book that established the area of Algorithmic Game Theory (presented in Section 1.3) was published in 2007 [132], as an edited collection of the most representative survey works, which justify this scientific r-evolution. The book includes three main parts that cover *computing in games*, *Algorithmic Mechanism Design*, and *quantifying of the inefficiency of equilibria*.

In another textbook [166], Shoham and Leyton-Brown consider multiagent systems and address issues of algorithmic, game-theoretic, and logical fundamental aspects. But the most recent is again an edited collection [12] that serves as a complete introduction to game theory, discussing those areas relevant for application in computer science, as *program design*, *synthesis*, *verification*, *testing and design of multi-agent or distributed systems*.

In a subsequent survey work, Joseph Halpern [66] considers complexity in the intersection between Computer Science and the Theory of Games. His survey includes bounded rationality, problems in computing Nash Equilibrium, and Algorithmic Mechanism Design. But his main interest is on a game-theoretic problem caused by Computer Science, the *price of anarchy* [94]. It is the ratio between the optimal centralized solution and the worst equilibrium, and measures the inefficiency of equilibria.

Recent findings show that new computational problems arise in the presence of algorithmic complexities. Daskalakis [43], Goldberg, and Papadimitriou [44] concentrated on

classical game theoretic concepts, such as the Nash Equilibrium that have been proved infeasible to be computed even in two player games. In their research work, they examined the computational complexity of the Nash equilibrium and they classified the problem of computing a Nash equilibrium into the PPAD complete class, that is, it belongs to the Polynomial Parity Argument in Directed graphs complexity class. Following this, they studied the complexity of computing approximate Nash equilibria in the *anonymous games*, in which players are unaware regarding other players' identity, and they proposed a polynomial time approximation scheme for these games, when the number of strategies is bounded.

The specific topic of computing equilibria that belongs to the area of computational complexity is thoroughly reviewed in [154] from the view of a game theorist. Tim Roughgarden first distinguishes *easy* from *hard* problems, then examines several equilibrium computation problems and the efficiency of their algorithms, and finally discusses the implications of the under consideration work for computation, games, and behavior.

In another *brief and biased* survey, as Tim Roughgarden [153] himself characterizes, there are interesting findings and conclusions that summarize the past, the present, and the future of work carried out in fields where Game Theory and Theoretical Computer Science meet. His major concentrations are again problems in Algorithmic Mechanism Design, the price of anarchy, and the complexity in computing Equilibria.

2.2 Game Theory in IT Security

In an interview to Sergiu Hart [70], the 2005 nobelist in Economics Robert Aumann, who has played an important role in developing the Theory of Games, emphasized the application of Game Theory in protecting computers against hackers. He characterized it as *a very grim game* analogous to war.

Five years later, in a plenary talk at GameSec 2010 [75], Jean-Pierre Hubaux, a distinguished researcher at EPFL, enumerated at least five IT security problems that have been

addressed using Game Theoretic tools: Security of physical and MAC layers, anonymity and privacy, intrusion detection systems, security mechanisms, and cryptography. The scope of the talk was to reveal the ways Game Theory can help in designing security mechanisms. In conclusion, Hubaux argued that game theoretic modeling of security mechanisms can assist predicting the behavior of the interacting parties and influencing them when mechanism design is used.

One of the earlier papers is by Lye and Wing [112], who constructed a general-sum stochastic game between an attacker and the administrator of a Target System. The presented game-theoretic method aims at examining the security of computer networks. It models attackers' and defenders' interactions with game theory and locates strategies to enhance the network security. One of the players in the game is the administrator in the role of the defender. The game is an imperfect information game in extensive form and Bayesian updating is used to calculate transition probabilities when moving from one state to another. Three significant aspects are discussed regarding this approach. First, the difficulty in computing solutions in stochastic models. Second, the model is extremely large when all possible states are included and this causes difficulties in handling it. Third, further difficulties are met in the construction of the game model and the corresponding numerical representation of outcomes. This work has been completely presented in [113]. Its main drawback is the absence of sound justification for the applicability of the proposed model.

At the same time, a work on MANETs [123] employed Game Theory to model the interactions between nodes in mobile ad hoc networks. The authors designed a security mechanism, called CORE, based on reputation to enforce cooperation and avoid selfish behavior. They showed that when the network takes no countermeasures against non-cooperative nodes, then the whole network malfunctions. Among the conclusions was that the CORE mechanism was able to ensure that at least half of the nodes behave cooperatively.

Another major research concern continues to be the growing complexity caused by sys-

tem infrastructure, bugs and security flaws. Ho et al. develop in [73] the Fundamental matrix, a framework for examining the qualitative nature of decision making. Using this matrix, they explain in a qualitative way many theorems and known results about optimization, complexity, and security. In their view, two are the most significant results derived from the development of their matrix; the first is that, as long as complexity increases, the things that have the potential of planning decrease, and therefore, things that have not been subject to planning are likely to produce negative payoffs. The second result is important for security and states that, if we cannot foresee all conceivable attacks, then there may exist an attack that will defeat the Target System.

A two player adversary game has also been constructed in [11] to model the interactions between spam senders and e-mail users. The authors explore the game strategies when players repeat the game and conclude that predicting the strategies, which will be adopted, might be of some help in tuning anti-spam filters. Another model suitable for spam detection has formalized the problem of adversaries, who manipulate the data of classifiers, to generate false negative alarms in [41]. The authors extend the naive Bayes classifier to produce an optimal classifier and construct a game between a classifier and an adversary, who uses optimal strategies too. In a similar way, other works address specific types of attacks, like [93] for backoff attack by Jerzy Konorski. Similarly, Mehran Fallah [56] has modeled several sophisticated flooding-defense scenarios as two player infinitely repeated games.

The problem of evaluating the resilience of computer networks is addressed in [31] by Bursztein et al. The authors propose a framework based on model and temporal logic, which uses two layers to represent incidents and dynamic responses with their corresponding delays. The first layer is the dependencies between files and services, and the second layer is the timed automaton games. A variant of TATL has been used, the $TATL\diamond$ (Timed Alternating-time Temporal Logic), and an example with a simple redundant Web service has been presented to illustrate an anticipation game. The complete implementation of this

anticipation game is the NetQi tool, presented in [32].

In 1999, David Burke designed a game theory model to represent Information Warfare [30]. His model was based on the class of repeated games of incomplete information. He described that the game has two players, the attacker and the defender, and that is a non-zero sum game. Likewise, the role of the Theory of Games and application issues in the information warfare are discussed in [67] and [68]. A survey that summarizes the game theoretic contributions in Network Security and Privacy is provided in [116]. Another one for the application of Game Theory only to Network Security can be found in [155]. In a different study [80], Game Theory has been used to assess an IT security specialist's expertise and quantify how much it affects the overall network security. The results reverse the assumption that skillful users with special security knowledge ensure a more secure network, because they might act as traitors or free-riders. Furthermore, a Game Theory Inspired Defense Architecture (GIDA) is proposed in [164] according to which a game model is considered between attackers who use methods of attacks and the system administrator who uses defense mechanisms.

Machado et al. have reviewed the game theoretic approaches used in wireless sensor networks [114]. As for the Quantal Response Equilibrium (QRE) chosen as a solution concept for the proposed model, it has been used in [103] to update the sequential equilibrium strategies in signalling games for online phishing classification.

2.3 Game Theory in Intrusion Detection

The problem of detecting an intruding packet in a communication network has been considered by Kodialam and Lakshman in [92]. The described game theoretic framework has been formulated in such a way that, the intruder picks paths to minimize chances of detection, whereas the network operator chooses a sampling strategy - among the developed sampling schemes - to maximize the chances of detection. The results derive from the solution of this

minmax problem. The same problem is addressed in [138] by Otrok et. al using a similar approach.

An attempt to infer the Attacker's Intent, Objectives, and Strategies (AIOS) with a game theoretic approach has been presented by Liu P. and Zang in [107]. Specifically, a general intensive-based method is introduced to model AIOS. But, the authors explicitly distinguish AIOS modeling and inference from Intrusion Detection, as two different areas. They establish this over the aspect that Intrusion Detection is based on the characteristics of attacks, while AIOS modeling is based on the characteristics of attackers. An extended version with this model is presented in [108]. This work shares the same motivation with our proposed model, because we formulate a game for Intrusion Detection and we use QRE to predict players' behavior.

The problems of reliability and accuracy in ID have been addressed by Cavusoglu et al. in [35]. The authors attempt to compare decision theory and game theory results, over a model framework for the configuration of IDSs, when firms are faced with strategic hackers. The goal of this configuration is to achieve the optimal balance between detection rate and false positive and false negative rates, in order to minimize the firm's cost. Based on the fact that IDSs are not perfect, the use of manual configuration techniques, by a human expert, is required to support the optimization scenarios. But although game theoretic findings have not fully been incorporated in ID, this work gives credible and valuable results in support of this new merged area.

Alpcan and Başar suggest the use of game theoretic tools for the development of practical schemes that can be incorporated in existing IDSs [8]. Their work investigates the potential of using game theory in developing a *formal decision and control framework* in ID. To achieve these goals, they develop a generic model for distributed IDSs applied on a network of sensors. In particular, two different schemes are proposed. The one is based on cooperative game theory and the other one on non-cooperative game theory. In both proposed schemes,

the authors assume that a classical IDS is working and detects intrusions, using the anomaly and the misuse detection techniques. The use of game theory does not aim at solving detection problems, or directly detect intrusions, but to optimize some general network security tradeoffs.

An extension of this work, but specific to access control systems, is presented in [9]. In a similar vein, they formulate a security game between an attacker and an IDS using the two different branches of Game Theory, a finite strategic game and a coalitional game solved using the kernel solution concept. But it is not clear, how the proposed game theoretic approaches of ID will be applied in an access control system. Finally, a 2-player zero-sum stochastic (Markov) security game between attackers and IDSs has been formulated in [10]. It is actually another extension of [9] to a stochastic and dynamic game. In all these approaches players' preferences have not been considered and payoffs have been arbitrarily assigned to outcomes.

Patcha and Park have modeled the interactions between the nodes of an ad hoc network as an incomplete information game in [143] and [144]. From this standpoint, they formulate a signaling game between an attacker and a node, where an IDS is present to defend attacks. They use a different perspective in their approach, by assuming that a node might be either a *regular node* or a *malicious node/attacker*, and that's how they build a signaling game. Although their approach is interesting, the authors do not utilize repeated games, and thus it is not possible to get insights from the structure of players' behavior.

Another game-theoretic attempt that focuses in computing the expected behavior of attackers is presented by Sallhammar et al. in [157] . This work is an extension of others' related work, Lye and Wing in [113] and Liu P. and Zang in [107]. The authors examine the attackers rewards and costs before acting, and assume that an attacker weighs up his benefit vs his loss. A game model for the interactions between attackers and IDSs is constructed to predict attackers' behavior. The solution of the game is limited to the computation and

use of NE in a stochastic game.

The problem of reducing false positives in MANETs when IDSs cooperate is addressed by Otrok et al. in [137]. The Shapley value is used for the cooperation of nodes. In a different work [139], Otrok et al. use mechanism design, sometimes called reverse Game Theory, to increase the effectiveness of an IDS for a cluster of nodes in ad hoc networks. The mechanism employs the Vickrey, Clarke, and Groves (VCG) mechanism[1], to balance the resource consumption of the nodes. The framework combines cooperative and non-cooperative game theory with a catch and punish method, which considers that the leader is either honest or dishonest in a black and white approach. In an extension of this work [126], the authors present leader election algorithms and propose two application schemes the Cluster Dependent Leader Election (CDLE) and the Cluster Independent Leader Election (CILE), in order to receive optimal election results in low cost.

Yu Liu et al. model the interactions between potential malicious nodes and defending nodes within a game theoretic framework for wireless ad hoc networks in [109]. The authors examine static and dynamic interactions in complete and incomplete information games using Baysian updating regarding the belief whether a node is malicious or not. As for the defender, the use of a lightweight IDS for nodes and of a heavyweight IDS are proposed to update his belief on attackers' types.

Agah et al. worked on non-cooperative game theoretic models of Intrusion Detection for sensor networks ([6], [5], [3]) and in [4] they describe a repeated game for preventing DoS attacks in wireless sensor networks. This paper addresses the problem of security in sensor networks in a different way than the one for ad hoc networks. In particular, a game has been formulated to prevent DoS attacks, targeted to wireless sensor networks. To achieve this goal, a game theoretic protocol is proposed, to operate in between an intrusion detector and the nodes of a sensor network.

[1]The Vickrey, Clarke, and Groves Mechanism is discussed by Noam Nisan in Chapter 9 of "Algorithmic Game Theory" [132], and by Paul Milgrom in Chapter 2 of "Putting Auction Theory to Work" [124]

To the best of our knowledge, the only game-theoretic approach in ID for insiders has been carried out by Liu D. et al. in [106]. They model an insider game as a stochastic game between an insider and an administrator. The insider threat is recognized as a problem to be addressed. The accurate prediction of insider's moves is the motivation of this work. But, only equilibrium analysis has been used to capture insider's future actions, in order to respond in an appropriate manner. The game is a one-shot game, no discussion on repeated games has been given, and instead of an IDS, the role of defender is played by the system administrator.

In 2009, Otrok et al. revert to the problem of reducing false positives in MANETs [140]. They use a combination of a Bayesian approach and the Dempster-Shafer theory to determine the belief value whether a sender is misbehaving or not. Their approach is a hybrid model, which focuses on lowering the uncertainty in detecting attackers and thus decreasing false positives.

To conclude, most of the presented related works are on one-shot games instead of repeated games. Most games are between an attacker and an administrator, instead of an attacker and an IDS. The NE solution is the only one used in non-cooperative game theoretic approaches in ID. The Shapley value has been used, but it belongs in cooperative game theory, which is outside the scope of the present work. The QRE has not been employed in ID. Insiders are the subject of research only in [106] using a one-shot game.

In our approach, we construct a novel repeated game model, between an insider and an IDS, to determine how an insider will interact in the future, and how an IDS would react to protect the system. For that reason, we solve the game and we extend the NE notion to the Quantal Response Equilibrium (QRE), to capture players' bounded rationality. This work has incorporated the results of our previous works that showed the potential of implementing a framework within which an IDS and a user will safely interact, preserving their own interests. A generic ID game model introduced in [86] was examined to discover

some essential features as its repeated divisions, and it was validated by solving and trying out the game with an insider. Following this, the insider threat was presented in [87], and preliminary results of calculating the QRE, when playing games with insiders repeatedly, were given. To exploit QRE results in ID, we proposed the use of a detection mechanism in [88]. To present a possible implementation scheme of this detection mechanism, we have created an application model and a detailed game-based detection algorithm.

Game theoretic solutions in Intrusion Detection Systems are discussed in an early survey [7] when only a few works had appeared. Among the problems presented as limitations of Game Theory is the unrealistic assumption that players are rational, and the authors conclude that human behavior is still unpredictable. However, the intentions that triggered a human's behavior have been modeled in the past, as reviewed in the next section.

2.4 Intention-based and Behavioral Detection in ID

David Levine was among the first who attempted to justify human's altruistic behavior and spite in games of standard economic models, using a model of signaling of intentions [100]. In a late research work jointed with Charness, they find evidences that market players are not only motivated by monetary reward, but social preferences play significant roles in these types of interactions, such as altruism and reciprocity [37]. It is therefore the intentions rather than the outputs that under certain circumstances concern market players more.

An early attempt to introduce intention modeling for intrusion detection appeared in [171]. The authors described a theoretical approach, by employing Task Knowledge Structures (TKS) and Cognitive Task Modeling (CTM), to construct a User Intention Identification system, the UII. The UII is an anomaly-based intrusion detection system that was developed as an autonomous module within the SECURENET EU funded project ([49], [170]). It aims at detecting deviations from normal behavior based on reasoning of users' intentions. The module is an expert system that integrates an advanced intelligent mod-

ule, an Intent Specification Language (ISL), to further increase the effectiveness of an IDS. Known intention patterns specified as authorized user intentions are maintained. The SECURENET ISL is used to identify the intentions of a user by trying to match his behavior with any of the authorized patterns of intentions. In the case of a mismatch, an alarm is raised.

The motivation of this approach are malicious actions that derive from legitimate activity, as we have realized in our approach and described in Section 1.1.1 that normal events might cause damage. But because intention profiles are used, the use of an ISL to detect user's intention is tight to the traditional anomaly detection technique.

Recently, Burgoon et al. [29] focused on the role of deception in order to determine intention from behavior. The authors recognize the difficulty in detecting human intentions and link deception to observable behaviors and psychological reactions. By studying deception, they argue that it is possible to identify intentions which might cause damage. Among their conclusions is that future research directions will obvert from the psychological cues of deception to the investigation of strategic interactions revealed in human behaviors. This is a preamble of the use of Game Theory in Intrusion Detection and, more widely, in IT Security.

McCabe et al. [117] discuss an interesting point for intentional detection using Game Theory, regarding the representation form of the game. They argue that players behave differently in normal forms than in extensive forms (see Section 1.2 for the related descriptions). As even little children are able to infer other people's intentions by "reading" what they say or what they do, in the Theory of Games we make assumptions about what players believe, expect or know for each other, as an attempt to read opponent's intentions. In their observations, they found that players cooperate more frequently in extensive forms, and they conclude in that the game form matters when we attempt detection of intentionality.

2.5 Summary

Reviewing the literature to discover how Computer Science incorporated game theoretic aspects and how Game Theory has been influenced by Computer Science, we realize that IT Security has accepted a large part of this merge. Moreover, Intrusion Detection is a field with a significant number of research works that aim at addressing old problems with completely new approaches. It is also remarkable that no research work in the area of ID with the use of a game theoretic approach is based on any of the classical intrusion detection techniques. It seems that game theoretic models in ID abandon anomaly, misuse, and specification-based detection techniques and convert to completely different methods.

Chapter 3

A Generic Intrusion Detection Game Model

> The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work.
>
> John von Neumann (1903 - 1957)

In Chapter 3, we construct a generic Intrusion Detection game model, to present how an Intrusion Detection System (IDS) interacts with a user. To represent Intrusion Detection as a game, we define the players and their actions, what players know when they act, possible outcomes, and players' preferences over these outcomes. We identify that such a game is a finite, lose-lose game, with sequential and simultaneous moves. We give the general formal

description of the game and we check its extensive form if it complies with the rules that ensures it. We close the description of this generic model with the formal definitions of Intrusion Detection as a game with *perfect* and *imperfect* information.

3.1 Representing Intrusion Detection as a Game

Von Neumann and Morgenstern [180] point out that in order to use mathematics (e.g. Game Theory) in a certain field, the problems in this field must be formulated and stated clearly. Consequently, what follows in the rest of this chapter provides an analytical representation of Intrusion Detection as a game.

As mentioned in the introduction, Intrusion Detection (ID) is an interactive situation, adopting the interactive characteristics that hold in IT. Interactions occur between a user and an IDS in a dynamic way. Game theory models dynamic interactions using the extensive form representation [95]. To capture this dynamic nature of the ID interactions, we present the generic game model of Intrusion Detection as an extensive form game. Among the different ways a game can be described, we have chosen the one suggested by Watson [181]. According to this representation, we have to specify the following formal elements:

- the list of *players*,
- their possible *actions*,
- what the players *know* when they act,
- the *outcomes* of the players' actions, and
- the players' *preferences* over these outcomes.

Identifying these elements for the Intrusion Detection game, we establish the following:

Players The interactions in ID are between two players; a user (potential attacker) and an Intrusion Detection System (IDS). Because a user can be one of several different types, the game that is being played also differs. Accordingly, the IDS adjusts its operation to the user's type. An IDS plays many games at a time, each with a different user. Games between an IDS and a group of attackers that attacks a Target System (TS) is outside the scope of this research work.

Actions Players' possible actions are numerous. Generally speaking, if the user is an attacker, then he chooses a method of attack to harm the TS, or attacks the TS just to gain information for his own purposes (benefit). When an IDS detects an attack, it selects the proper counteraction to prevent damage.

Information At the beginning, no player has enough information, and thus they act under great uncertainty. As for the case of an attacker, he might observe a TS in order to gain the necessary information to perform an attack (reconnaissance attack [159]). Similarly, an IDS observes the protecting TS to detect attacks (logs).

Outcomes In an ID game, the winner is an attacker who successfully attacks a TS and remains undetected, at least until he achieves his goals. On the other side, the IDS of a TS, which detects an attempt of an attack and prevents attacker's further actions before any threat has been realized, is the winner of the game over this attacker.

Preferences In most cases, an attacker prefers achieving his goals over not achieving them, but without being detected. On the other side, an IDS prefers detecting an attempt of an attack over detecting a successful attack. Moreover, an IDS prefers detecting real attacks over causing false alarms. Table 3.1 shows these general preferences of attackers and IDSs.

Figure 3.1 illustrates the described formal elements in a player-centric diagram. In the subsequent subsections, we examine in detail these formal elements, in order to describe the

General Preferences		
	Successful Attack	
Detection	YES	NO
YES		IDS wins
NO	Attacker wins	

Table 3.1: General preferences of attackers and IDSs

ID game in a systematic way, to cover most of its aspects, and prepare the construction of the generic ID game model.

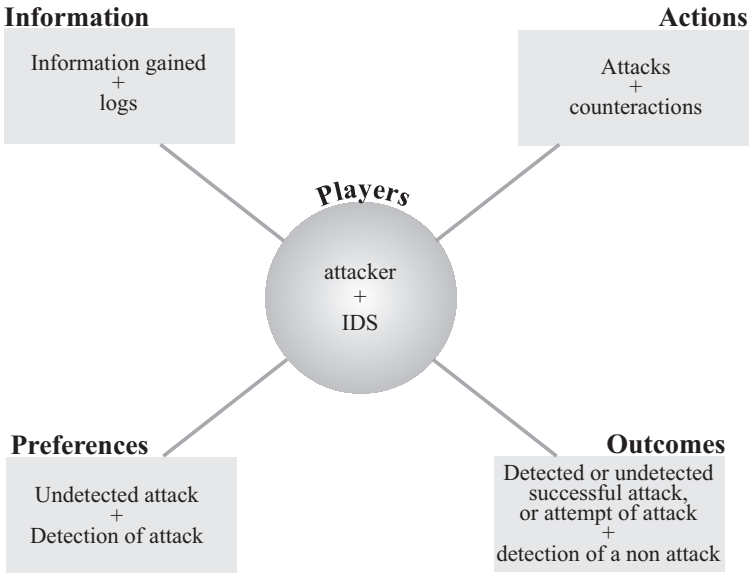

Figure 3.1: The formal elements of the ID game

3.1.1 Players and Actions

The game has two players, an IDS indicated by I and a user indicated by U. Player U might be a normal user or an attacker. But, even if he is a normal user, he might unintentionally harm the TS. Consequently, player U is considered as a general user, and no further categorization is needed before he acts.

Each player has a number of possible actions to choose from. Player I, examining player U's actions, allow player U to continue using the TS by choosing C, if player I comes to the conclusion that player U acts legitimately. Conversely, player I chooses P to prevent additional damage to the TS, if it decides that player U is committing illegal actions. In short, in this game player I has two choices; choice C to allow player U to continue using the TS, and choice P to prevent player U to attack or to further damage the TS. In real cases, this binary approach reflects that player U requests a service or a resource from the TS, and player I either accepts to fulfil the request (choice C) or refuses (choice P). Although other approaches might appear to include more than two choices, the interpretation is the same.

Similarly, player U has three possible actions; L when acting legitimately, A when acting illegally generating attacks, and E when he decides to exit the TS and he logs out. Comparing to player I's actions, player U has one more action to choose, that is, he has three actions. Regarding player U's activity, it is either legitimate or illegal, as defined in Section 1.4.

3.1.2 Information

What players know when they act is significant, because it implies the type of the game that will be played. Fully informed players play complete information games, while partially informed players play incomplete information games. In the game of ID, player U might know the existence of an IDS behind the TS. If he has this information and he is a common

user, then his actions are adjusted to normal behavior avoiding mistakes. But if he is an attacker, then he tries to cover his attack traces, in order to avoid detection by the IDS.

As for player I, the first time it monitors the actions of a new user, it has no information if he is a potential attacker. Besides, whenever it decides upon the actions of a user, whether they are normal or attacking, it is not 100% sure. It is the measured accuracy of an IDS discussed in Sec. 1.1.4, which gives the percentage of detected true attacks and the corresponding false alarms.

3.1.3 Outcomes

The outcome of the ID game might be one of the following listed below:

- an attempt of an attack is an unsuccessful attack at the real TS. Such an attack could be detected or left undetected by an IDS,
- an attempt of an attack at a fake TS (e.g. honeypots). Such an attack could be detected or left undetected by an IDS,
- a successful attack, which could be detected or undetected,
- a successful attack which is not detected in real time by an IDS, and the damage cannot be confirmed ex post (e.g. disclosure of information), and
- detection of a non attack (i.e. false positive alarm).

3.1.4 Preferences

In Intrusion Detection, the specification of players' preferences is not clearly implied by the rules of the game that is being played between a user and an IDS. It is well known that user's preferences depend on user's type. When a user is normal with no intentions to act illegally, then he has reverse preferences from a user with malicious plans to damage a TS. In addition, an attacker's preferences significantly differ from an IDS's preferences, because

his possible actions and their corresponding outcomes diverge and conflict. As the type of an attacker is our major interest, we specify preferences for an attacker rather than for a normal user. The preferences must satisfy the transitivity condition [135]. Generally, one can assume the following preferences for an attacker and an IDS.

Attacker's Preferences

An attacker prefers carrying out a successful attack avoiding detection over carrying out a successful attack and being detected by an IDS. Moreover, an attacker prefers an unsuccessful attack that left undetected over a detected unsuccessful attack.

Finally, an attacker might prefer a successful detected attack over failing to successfully attack the TS without being detected. Although this seems to be reasonable, in some cases an attacker might prefer exactly the opposite. Table 3.2 summarizes attacker's preferences arranged from the most preferred (A) to the least one (D).

Attacker's Preferences		
	Attack	
Detection	Successful	Unsuccessful
YES	B	D
NO	A	C

Table 3.2: Attacker's preferences in ID in IT Security

IDS's Preferences

An IDS prefers detecting an attempt of an attack, i.e. an unsuccessful attack, over detecting a successful attack. Similarly, an IDS prefers detecting over no detecting a successful attack that usually gives information of the Target System loopholes, its threats, etc.

Finally, an IDS prefers characterizing a normal action as an attack and causing a false alarm over no detecting a successful attack, which seems to be the worst case scenario. Table 4.4 summarizes IDS's preferences arranged from the most preferred (A) to the least one (D).

IDS's Preferences		
	Attack	
Detection	Successful	Unsuccessful
YES	B	A
NO	D	C

Table 3.3: IDS's preferences in ID in IT Security

Identifying the game as a win-win or a lose-lose game

Examining attacker's and IDS's preferences on the above tables to identify whether such a game is potentially a lose-lose game or a win-win game, it is noticeable that D preferences conflict and cannot be accomplished simultaneously, but C and B preferences, which are awful but not the worst case preferences, are common to both players. Therefore, a game between an attacker and an IDS may well be characterized as a lose-lose game and definitely not as a win-win game.

3.2 Sequential and Simultaneous Moves

Next, the key question to be addressed is how this game is being played, with simultaneous or with sequential moves. The crucial criterion to answer this question is to take into account first, that using a TS and requesting a service from it, the user waits for a response,

although he does not usually even realize it, and afterwards, the user makes another request to which the TS replies too, and so on. Thus, the kind of interaction formulated here is a sequential-move interaction, like the one taking place between two players in a chess game.

However, when the TS is protected by an IDS, the user is not only interacting with the TS, but he is also interacting with the IDS. In the latter kind of interaction, the user is acting and at the same time the IDS collects data related to this action, filters it and decides to counteract in the case of an attack. The IDS performs a counteraction ignoring the user's current action.

Elaborating the described interaction into the game theoretical approach [51], in Intrusion Detection an attacker plans his moves before he acts and calculates the *future consequences*. Up to this point, the game is a sequential-move game. But when the attacker starts applying this plan and confronts the existence of an IDS protecting the TS of his attack, then he is trying to discover what this IDS is going to do *right now*. This argument indicates that the game includes also simultaneous moves.

On the contrary, an IDS has been designed and implemented to incorporate one or more ID techniques, which lead to a predefined plan of its moves, and to calculate the future consequences to the TS that protects. Up to this point, the game again is a sequential-move game. But when a user enters the system, the IDS observes his moves to decide whether he is an attacker or not, and according to its design, to figure out what the attacker is going to do right now. The conclusion once more is that the game includes also simultaneous moves. Consequently, Intrusion Detection in IT Security is a game that combines both sequential and simultaneous moves.

3.3 General Formal Description of the Game

Consider the extensive form of the Intrusion Detection game depicted in Fig. 3.2. A user (player U) attempts to enter a TS protected by an IDS (player I). The user might

successfully login to the TS or not (e.g. typing in a wrong password). Even if he gains access to the TS, he might already be a member of a black list. Therefore, player I moves first at the initial node (the root) of this game denoted by an open circle, when player U attempts to enter the TS. Then, examining this attempt, player I has two choices; to allow the user continuing (choice C) or to prevent the user from using the TS (choice P) which ends the game. In the latter case, it is assumed that player I has achieved to detect a real potential attacker and has not caused a false alarm. Hence, the outcome at this point of the game is the vector *(detection, attempt of an attack)* for player I and player U respectively.

If choice C has been selected by player I, then player U has three choices; to perform legal actions (choice L), to attack the TS (choice A), or to exit from the TS (choice E). If player U exits the TS, then the game ends with outcomes *(no detection, attempt of an attack)*. The reason for these payoffs is first that the user has achieved to enter the TS and the IDS did not detect an attack even if he was a masquerade, so this is counted as penetration, and second that the user did not attack the TS although he might got the keys (the pair username and password) and checked them against the TS, to attack it another time in the future.

The game continues with player U selecting a legal action over the TS or attacking the TS. In both instances, player I analyzes afterwards the opponent's move, and decides whether he is acting legally or not. If player I chooses P, then the payoffs of the game totally diverge. In particular, if player U has committed an attack (choice A) then the payoffs are *(detection, successful attack)*, otherwise (choice L), the payoffs are *(detection, no attack)* which constitutes a false alarm.

Alternatively, when player I allows player U continuing working with the TS (choice C), while player U is acting legally, then player U might either proceed with legal actions (choice L), or with an attack (choice A), or he decides to exit the TS (choice E) terminating the game. This outcome of the game results in the payoffs *(no detection, no attack)*. The

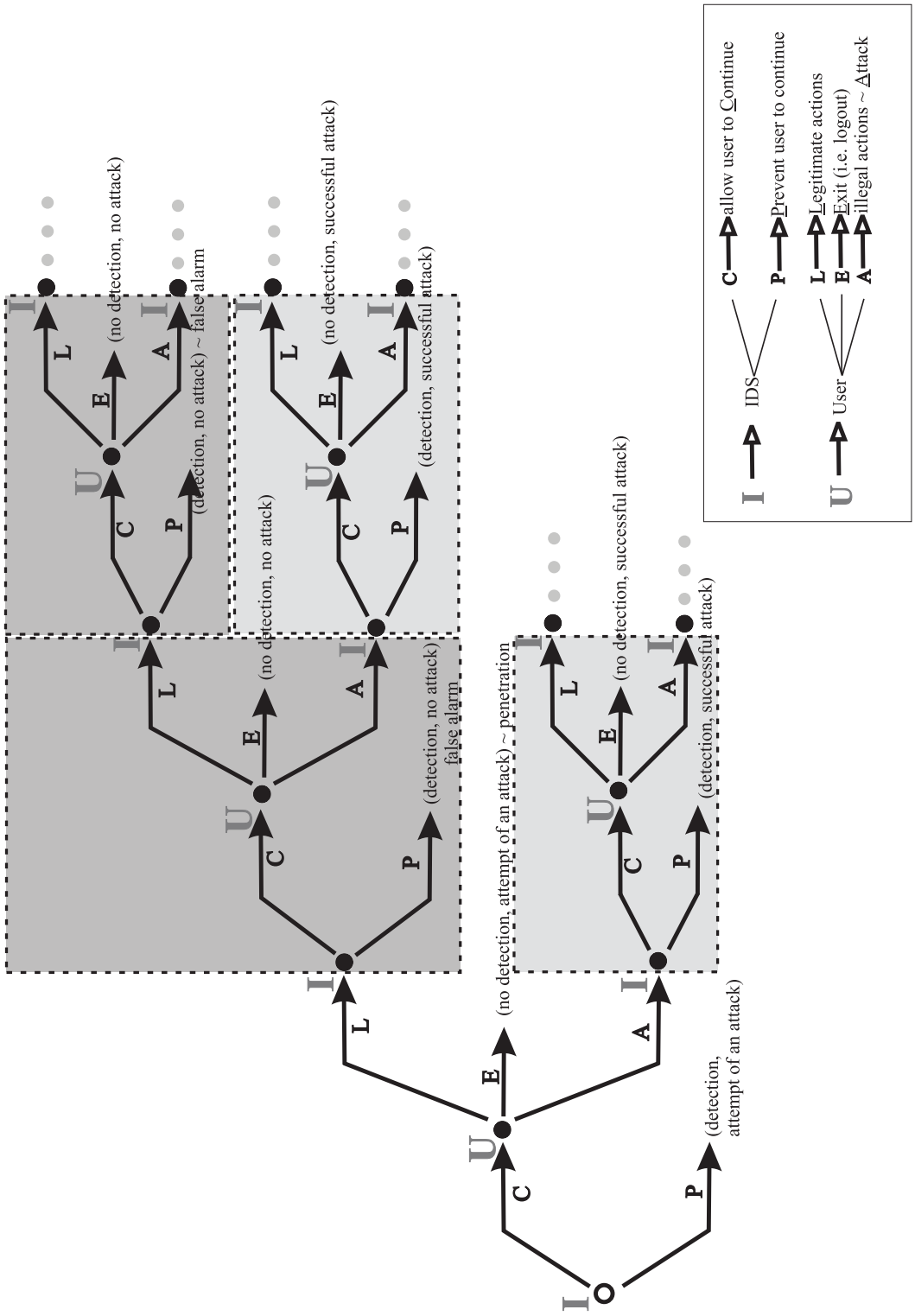

Figure 3.2: Intrusion Detection as an extensive form game

described stage of the game surrounded by a dashed line rectangle as shown in Fig. 3.2, is a repeated division of the game, which leads to the end of the game when player U chooses E.

Exploring further the extensive form of the ID game for repeated divisions, we locate two parts; the one is related to legal actions and the other one to attacks. Figure 3.3 represents the extensive form game explained in detail above, displaying two separate divisions and their iterations. Although the form of the ID game looks as never ending, each of the repeated divisions definitely has a branch where the game ends, and thus the game under study is a finite game.

3.4 Checking the Extensive Form

Extensive form games should give a picture of a tree. There are two rules that ensure this form [95]; first, the number of arrows pointing out from a node must be at least one, and the number of arrows pointing at a node must be at most one, and second, retracing the tree in a backward fashion from a node towards the initial node, the starting node should not be reached again drawing a cycle, but actually the initial node should be the end of this backtracking.

The first rule implies that a player has at least one action to perform when it is his turn to play, and that after an action of a player, either another player is next, or the game ends to a payoff vector of a specific outcome. The second rule aims at solving games in extensive form using backward induction, since they have the form of a tree.

The Intrusion Detection game described above has been modeled in the form of a tree. Checking the game against the first rule, it is apparent that the number of arrows pointing in any node, as well as the number of arrows pointing out from any node, satisfy this rule. Similarly, examining the plausibility of backtracking from any node towards the initial node of the game, no circle would be drawn and the initial node would be reached.

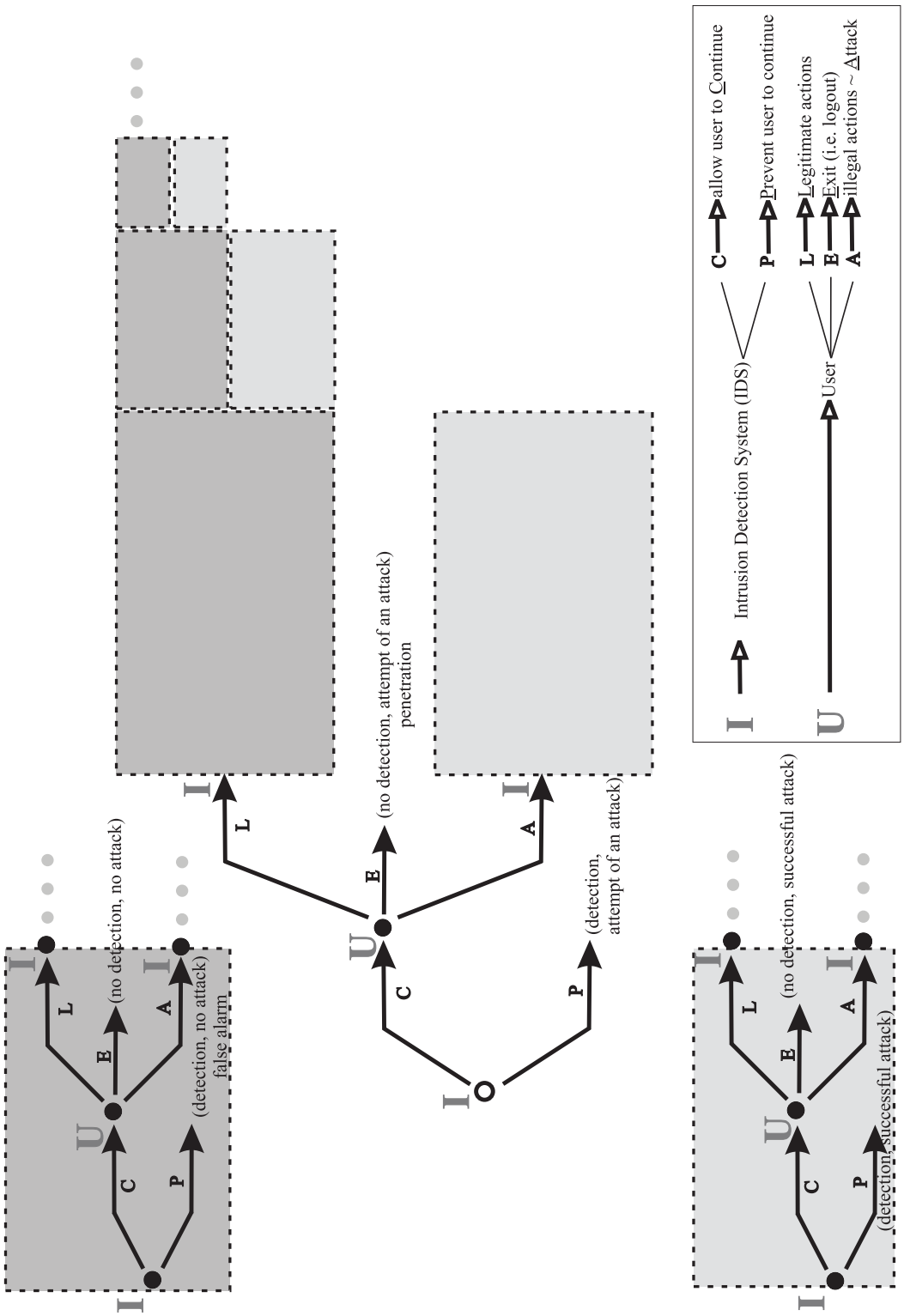

Figure 3.3: Intrusion Detection and the repeated divisions of the game

3.5 Formal Definitions

With the use of the definitions given for extensive form games with perfect and imperfect information in [135], we formulate the corresponding definitions in the next two sections for the game of ID.

3.5.1 The ID Game with Perfect Information

Definition 3. Let $G_p = \langle N, H, P, (\succsim_i) \rangle$ be an extensive game with perfect information that models Intrusion Detection, where

- $N = \{I, U\}$ is the set of players;
- H is the infinite set of sequences that consists of the histories $\varnothing$, C, P, (C, L), (C, E), (C, A), (C, L, C), (C, L, P), (C, L, C, L), (C, L, C, E), (C, L, C, A), (C, L, C, L, C), (C, L, C, L, P), (C, L, C, L, C, L), (C, L, C, L, C, E), (C, L, C, L, C, A),..., (C, L, C, A, C), (C, L, C, A, P), (C, L, C, A, C, L), (C, L, C, A, C, E), (C, L, C, A, C, A),..., (C, A, C), (C, A, P), (C, A, C, L), (C, A, C, E), (C, A, C, A),...;
- P is the player function that indicates the player who takes an action after a history ($P(h)$), i.e. $P(\varnothing) = I$, $P(I) = U$, and $P(U) = I$ for every $h \neq \varnothing$, that is to say for every nonterminal history;
- $\succsim_i$ is the preference relation on Z for player $i \in N$, that is, the preference relation of player I is (detection, unsuccessful attack) $\succ_I$ (detection, successful attack) $\succ_I$ (no detection, unsuccessful attack) $\succ_I$ (no detection, successful attack), and the preference relation of player U is (no detection, successful attack) $\succ_U$ (detection, successful attack) $\succ_U$ (no detection, unsuccessful attack) $\succ_U$ (detection, unsuccessful attack).

3.5.2 The ID Game with Imperfect Information

Definition 4. Let $G_i = \langle N, H, P, f_c, (\mathcal{I}_i)_{i \in N}, (\succsim_i) \rangle$ be an extensive game with imperfect information that models Intrusion Detection, where

- $N = \{I, U\}$ is the finite set of players;
- H is the infinite set of sequences that consists of the histories $\varnothing$, C, P, (C, L), (C, E), (C, A), (C, L, C), (C, L, P), (C, L, C, L), (C, L, C, E), (C, L, C, A), (C, L, C, L, C), (C, L, C, L, P), (C, L, C, L, C, L), (C, L, C, L, C, E), (C, L, C, L, C, A),..., (C, L, C, A, C), (C, L, C, A, P), (C, L, C, A, C, L), (C, L, C, A, C, E), (C, L, C, A, C, A),..., (C, A, C), (C, A, P), (C, A, C, L), (C, A, C, E), (C, A, C, A),...;
- P is the player function that indicates the player who takes an action after a history ($P(h)$) (a member of $N \cup c$, where c is the chance player), i.e. $P(\varnothing) = I$, $P(I) = U$, and $P(U) = I$ for every $h \neq \varnothing$, that is to say for every nonterminal history;
- f_c is the function that associates with every history h for which $P(h) = c$ a probability measure $f_c(\cdot|h)$ on $A(h)$, each independent of every other such measure, e.g. $(f_c(a|h))$ is the probability that a occurs after the history h;
- $\mathcal{I}_i$ is the information partition of player $i, i \in N$ and a set $I_i \in \mathcal{I}_i$ is an information set of player i;
- $\succsim_i$ is the preference relation on Z for player $i \in N$, that is, the preference relation of player I is (detection, unsuccessful attack) $\succ_I$ (detection, successful attack) $\succ_I$ (no detection, unsuccessful attack) $\succ_I$ (no detection, successful attack), and the preference relation of player U is (no detection, successful attack) $\succ_U$ (detection, successful attack) $\succ_U$ (no detection, unsuccessful attack) $\succ_U$ (detection, unsuccessful attack).

3.6 Summary

Constructing a generic game in extensive form to model Intrusion Detection, we specified first the formal elements of the game; players, actions, information, outcomes, and preferences. The general formal description of the ID game reveals the dynamic interactions between a user and an IDS, which make sense when examining players' preferences. By interpreting the functionality of the ID game model, we conclude that it is a lose-lose game with sequential and simultaneous moves. Two repeated divisions of the game have been located that correspond to legal and attacking actions respectively. The formal definitions of the ID game model with perfect and imperfect information formulate a mathematical construction of Intrusion Detection using the Theory of Games.

Chapter 4

Playing Repeatedly the ID Game

All movements go too far.

Bertrand Russell (1872 - 1970)

In Chapter 4, we examine the repeated form of the ID game with perfect and imperfect monitoring, following the generic ID game model established in Chapter 3. We first construct a static version of it, when the game is played once, known as the stage game [115], which constitutes the building block of the repeated game. Subsequently, we specify the players, the pure actions, the action profiles, the preferences of a normal user, an attacker, and an IDS, and the mixed actions of the game. The repeated game with perfect monitoring follows as the stage game that is being played again and again in every period. Then, the repeated ID game with imperfect public monitoring is formulated and discussed based on the specifications of the game with perfect monitoring.

4.1 Repeating with Perfect Monitoring

In perfect monitoring the players of the game are fully informed about each others' moves. A user is fully informed if he knows the existence of an IDS and what the IDS does at every move. On the other side, an IDS is fully informed if it identifies accurately a user's move.

The latter imposes that an IDS has 100% detection rate, which means that it is reliable, and it raises no false alarms at all, because it is accurate. Although such an IDS does not really exist, it is of great interest to study the situation of an IDS with 100% detection rate as the simplest case, which provides us with a clear benchmark. That is why it has been considered as the first thesis hypothesis stated in Section 1.5.

4.1.1 The Stage Game Model

In order to formulate the stage game appropriately for the construction of the repeated game, we look into the players, their pure and mixed actions, the action profiles, and players' preferences, as described in the subsequent paragraphs. The dynamic structure of the game restricts the use of an extensive form stage game, which is repeated in every period.

Players

Each of the users of the Target System plays a game with the IDS which protects it. Every such a game is independent from the other games, but, a user might take into account other users and their actions, and the IDS has to play with all of them. Therefore, the game has a set of $n-1$ user players and an IDS, that is, there are n players in this game, $N = \{U_1, U_2, \ldots, U_{n-1}, IDS\}$. Each user plays separately and isolated with the IDS, in some cases ignoring even the existence of other players. Thus, there is no cooperation between the user players $U_1, U_2, \ldots, U_{n-1}$. From this standpoint, we assume the game as a two player game, between a user U and an Intrusion Detection System IDS, forming $N = \{U, IDS\}$. Because an IDS plays such a game with each user, this is a representative form of a n-player game between an IDS and $n-1$ noncooperative users. The IDS is expected to play each game as effectively as possible and it has been designed, implemented, and configured appropriately for this purpose.

Pure Actions

A user U has a number of choices, called actions in the stage game, denoted by A_U. An element of this non empty set, indicated with a_{U_k}, is the k available action of player U. So, the set of pure actions for player U is given by,

$$A_U = \{a_{U_1}, a_{U_2}, \ldots, a_{U_k}\},\ k = 1, 2, \ldots$$

Player U's action set is a finite set. Similarly, the IDS has a number of available actions indicated as,

$$A_{IDS} = \{a_{IDS_1}, a_{IDS_2}, \ldots, a_{IDS_m}\},\ m = 1, 2, \ldots$$

In particular, player *IDS*, examining player U's actions, allows player U to continue using the TS by choosing C, if player *IDS* comes to the conclusion that player U acts legitimately. Conversely, player *IDS* chooses P to prevent additional damage to the TS, if it decides that player U is engaging in illegal actions. In short, in this game player *IDS* has two choices; choice C to allow player U to continue using the TS and choice P to prevent player U to attack or to further damage the TS.

In real cases, this binary approach reflects that player U requests a service or a resource from the TS, and player *IDS* either accepts to fulfil the request (choice C) or refuses it (choice P). Although other approaches might appear to include more than two choices (see Chapter 5), the interpretation is the same.

Similarly, player U has three possible actions; L when acting legitimately, A when acting illegally generating attacks, and E when he decides to exit the TS and so he logs out. Comparing to player *IDS*'s actions, player U has one more action to choose, that is, he has three actions.

The third action, the choice of exiting the game, has been selected to balance the ends of the game. So, player *IDS* ends the game by deciding to prevent (choice P) player U to

continue using the TS, when the user acts illegally, and similarly, player U ends the game by exiting (choice E) the TS, when he has a reason to do so.

To conclude, the set of pure actions for player U is

$$A_U = \{a_{U_1}, a_{U_2}, a_{U_3}\} = \{L, A, E\},$$

and the set of pure actions for player IDS is

$$A_{IDS} = \{a_{IDS_1}, a_{IDS_2}\} = \{C, P\}.$$

Action Profiles

The set of actions for every player is a compact subset of the Euclidean space $\mathbb{R}^2$, and because it is finite, the game is a finite game. The set of profiles, corresponding to the set of pure actions, is the combination of actions, one action for each player, defined as the Cartesian product,

$$A \equiv \prod_{i \epsilon N} A_i.$$

Specifically for this two-player game, the set of pure action profiles is defined by

$$A \equiv A_U \times A_{IDS} = \{(L, C), (A, C), (E, C), (L, P), (A, P), (E, P)\}.$$

The number of profiles that are elements of this set is $k \times m = 3 \times 2 = 6$. We indicate a member of a profile with $(x_i)_{i \epsilon N}$ or simply (x_i).

Preferences

Next we specify players' preference rankings over the action profiles. Each player $i \epsilon N$ ranges the action profiles from the most preferred to the least one. A preference relation $\succeq_i$ on the set $A = x_{i \epsilon N} A_i$ for player i specifies a binary relation, represented by a payoff

function $u_i : A \to \mathbb{R}$. The function u is a continuous function, known also as von Neumann-Morgenstern utility function. For two pure actions a_{i_1} and a_{i_2} of player i, $u_i(a_{i_1}) \geq u_i(a_{i_2})$, whenever $a_{i_1} \succeq a_{i_2}$. The values of this function are called payoffs or utilities [135].

Well, the fact is, preferences vary between different types of user players. Different user players have different motivations for doing something, and thus their intentions diverge. Besides, not all user players have the same skills to complete an activity.

For example, an internal user of a system decides one day to harm it. He is quite concerned in hiding his traces rather than successfully achieving his goals, because if caught he might lose his job. Moreover, he is patient enough in completing his goals, since he has plenty of time as an internal user.

On the contrary, suppose a user player gains access in a system. He is only concerned in causing damage for his own purposes, paying no attention in situations where he is being caught and stopped by an IDS. Under most circumstances, he is impatient, because he does not have time to lose.

In both cases, the user player might be skillful enough and fully informed to commit his actions. This happens either because of his original job (employee), or as a result of the work he has done so far towards this direction. For example, he might be fully informed because he has accomplished another attack before, usually named as reconnaissance attack. Nevertheless, the action profile is ranked in a different way if an internal user is considered in the place of an outsider attacker.

Just as user players' preferences depend on user types, the IDS's preferences are being adjusted, whenever a certain type of user is identified. Specifically, the IDS is concerned in not allowing an attacker to damage the system, but it is also concerned in allowing a normal user to continue his work as it should be. Interestingly, the IDS's preferences disagree if the opponent player has been identified as a normal user, or as an attacker, or if it is not identified yet.

Because when playing repeatedly the ID game the user type matters, in the sequel, we consider the preferences of two different types of user players, a normal user and an attacker, and the preferences of the IDS when playing with different user players. Then, we construct the corresponding utility functions following Binmore's method [21], to quantify the outcomes of the proposed game in a variety of instances.

Normal User's Preferences

Interpreting the action profiles when the user player is a normal user of a system, we consider the corresponding set of a Normal User's preferences, denoted by $\mathcal{N}$. This set includes the following four items:

$$\mathcal{N} = \{\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \mathcal{N}_4\},$$

where,

$\mathcal{N}_1$: A Normal User is acting legitimately and the IDS allows him to continue.

$\mathcal{N}_2$: A Normal User is being prevented by the IDS although he is acting legitimately.

$\mathcal{N}_3$: A Normal User is acting illegally but the IDS allows him to continue.

$\mathcal{N}_4$: A Normal User is being prevented by the IDS because he is acting illegally.

Although the action profiles for this game enumerates six items, as described before, the actual preferences of a normal user player are only four. The reason for this is the third action of a user player, the exit (E) action, which ends the game. In cases where the game ends at an exit action on behalf of the user player, the outcome is equivalent to the corresponding previous continue action on behalf of the IDS. As a result, the normal user's preferences are consistent with the action profiles of the game, as specified in the following:

$$\mathcal{N}_1 \rightarrow (L, C), (E, C)$$

$\mathcal{N}_2 \rightarrow (L, P), (E, P)$

$\mathcal{N}_3 \rightarrow (A, C), (E, C)$

$\mathcal{N}_4 \rightarrow (A, P), (E, P)$

For a Normal User it is most desirable to act legitimately without preventions and his next choice is to act illegally with no stops, because illegal actions are not intentional. Similarly, he prefers the IDS to prevent his actions when these are illegal rather than legitimate. Based on these lines of reasoning, the ranking of these preferences from the less preferred to the most one gives the following:

$\mathcal{N}_3 \prec \mathcal{N}_1$ and $\mathcal{N}_2 \prec \mathcal{N}_4$.

In order to get Normal User's preferences fully ordered, there is a need for connection between these two relations. A Normal User most prefers $\mathcal{N}_1$ and his worst choice is $\mathcal{N}_2$. Additionally, examining his preferences between $\mathcal{N}_3$ and $\mathcal{N}_4$, a Normal User prefers $\mathcal{N}_3$ because he has no intention to harm the TS, so he wants an uninterruptible use of it. This interpretation results into the following chain of preferences:

$$\mathcal{N}_2 \prec \mathcal{N}_4 \prec \mathcal{N}_3 \prec \mathcal{N}_1 \tag{4.1}$$

From this set of preferences and the defined relations that reflect the ranking, we will define the corresponding utility function for the Normal User. Suppose that $U_{\mathcal{N}} : \{\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \mathcal{N}_4\} \rightarrow \mathbb{R}$ is the utility function of the Normal User. With regard to his preferences, the worst action profile is $\mathcal{N}_2$. So, $U_{\mathcal{N}}(\mathcal{N}_2) = 0$. At the other end, he mostly prefers $\mathcal{N}_1$. Therefore, $U_{\mathcal{N}}(\mathcal{N}_1) = 1$. Selecting anyone between the other two preferences in the middle of the rank, we assign $\frac{1}{2}$ utility to the preference $\mathcal{N}_4$, that is, $U_{\mathcal{N}}(\mathcal{N}_4) = \frac{1}{2}$. Finally, because $\mathcal{N}_3$ is the intermediate between $\mathcal{N}_4$ and $\mathcal{N}_1$, we define $U_{\mathcal{N}}(\mathcal{N}_3) = \frac{3}{4}$, by dividing the distance between $U_{\mathcal{N}}(\mathcal{N}_1)$ and $U_{\mathcal{N}}(\mathcal{N}_4)$ with 2, that is,

$$U_{\mathcal{N}}(\mathcal{N}_3) = U_{\mathcal{N}}(\mathcal{N}_4) + \frac{U_{\mathcal{N}}(\mathcal{N}_1) - U_{\mathcal{N}}(\mathcal{N}_4)}{2} = \frac{1}{2} + \frac{1-\frac{1}{2}}{2} = \frac{1}{2} + \frac{\frac{1}{2}}{2} = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}.$$

In this way, instead of having a ranking of encoded preferences as presented in expression (4.1), we have real numbers to represent preference relations, very handy for calculations. This is a more convenient representation when making choices, where the criterion is the maximization of the utility function $U_{\mathcal{N}}$. Table 4.1 below summarizes in the second row the specified utilities for the Normal User. The third row describes the corresponding utilities free of fractions, after multiplying them by 4.

x	$\mathcal{N}_2$	$\mathcal{N}_4$	$\mathcal{N}_3$	$\mathcal{N}_1$
$U_{\mathcal{N}}(x)$	0	$\frac{1}{2}$	$\frac{3}{4}$	1
$4 \cdot U_{\mathcal{N}}(x)$	0	2	3	4

Table 4.1: Normal User's Utility Function

Attacker's Preferences

When the user player is an attacker, we consider in a similar way the set of an Attacker's preferences, denoted by $\mathcal{A}$.

$\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\}$,

where,

$\mathcal{A}_1$: An Attacker does not achieve his goals and he is not being detected.

$\mathcal{A}_2$: An Attacker does not achieve his goals and he is being detected and stopped by the IDS.

$\mathcal{A}_3$: An Attacker achieves his goals without being detected.

$\mathcal{A}_4$: An Attacker achieves his goals and is being detected and stopped by the IDS.

The attacker's preferences are consistent with the action profiles of the game. We assume that when an attacker does not achieve his goals, he acts legitimately, as specified below:

$\mathcal{A}_1 \rightarrow (L, C), (E, C)$

$\mathcal{A}_2 \rightarrow (L, P), (E, P)$

$\mathcal{A}_3 \rightarrow (A, C), (E, C)$

$\mathcal{A}_4 \rightarrow (A, P), (E, P)$

As for an Attacker, the most preferable outcomes of the game might be those where he is achieving his goals. Between being detected or not, he prefers the second. In addition, he does not prefer to be prevented when acting legitimately, but he prefers to continue. Ranking these preferences from the less preferred to the most one, we get:

$\mathcal{A}_2 \prec \mathcal{A}_1$ and $\mathcal{A}_4 \prec \mathcal{A}_3$

To connect the above relations and find an ordered ranking of Attacker's preferences, we examine further his profile. Taking into account that because he mostly prefers to achieve his goals no matter whether he will be detected or not, he is dedicated to his goals. Therefore, the other two preferences will eventually follow. This explanation results into the following ordered attacker's preferences:

$$\mathcal{A}_2 \prec \mathcal{A}_1 \prec \mathcal{A}_4 \prec \mathcal{A}_3 \tag{4.2}$$

Similarly, we will define the corresponding utility function for the Attacker, based on the set of preferences $\mathcal{A}$ and the defined preference relations. Suppose that $U_{\mathcal{A}} : \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\} \rightarrow \mathbb{R}$ is the utility function of the Attacker. With regard to his preferences, an Attacker dislikes action profile $\mathcal{A}_2$ and prefers mostly $\mathcal{A}_3$. For this reason, we define $U_{\mathcal{A}}(\mathcal{A}_2) = 0$ and $U_{\mathcal{A}}(\mathcal{A}_3) = 1$, respectively. If we select $\mathcal{A}_1$ as one intermediate between $\mathcal{A}_1$ and $\mathcal{A}_4$ which are left, we define its utility as $U_{\mathcal{A}}(\mathcal{A}_1) = \frac{1}{2}$. Finally, we calcu-

late $U_{\mathcal{A}}(\mathcal{A}_4)$, following the same reasoning as we did before for the Normal User, and we define $U_{\mathcal{A}}(\mathcal{A}_4) = \frac{3}{4}$. In Table 4.2 below, the second row summarizes the defined utilities for the Attacker. The third row describes the corresponding utilities free of fractions, after multiplying them by 4.

x	$\mathcal{A}_2$	$\mathcal{A}_1$	$\mathcal{A}_4$	$\mathcal{A}_3$
$U_{\mathcal{A}}(x)$	0	$\frac{1}{2}$	$\frac{3}{4}$	1
$4 \cdot U_{\mathcal{A}}(x)$	0	2	3	4

Table 4.2: Attacker's Utility Function

In a case where an Attacker has another profile, the preference ranking would be totally different. For example, if an Attacker is an internal attacker, an insider of the Target System, then he mostly prefers not to be detected rather than attacking. His preferences derive from the double role he plays, the mixture between a Normal user and an Attacker too. In such a situation, the ranking of his preferences might be as below:

$$\mathcal{A}_2 \prec \mathcal{A}_4 \prec \mathcal{A}_1 \prec \mathcal{A}_3 \tag{4.3}$$

In Chapter 5, we examine thoroughly insiders' preferences, when they play the ID game in a form especially constructed for them.

IDS's Preferences

The set of an IDS's preferences is denoted by $\mathcal{IDS}$ and includes four items, as described in the sequel:

$\mathcal{IDS} = \{\mathcal{IDS}_1, \mathcal{IDS}_2, \mathcal{IDS}_3, \mathcal{IDS}_4\}$,

where,

$\mathcal{IDS}_1$: The IDS does not detect any illegal action and allows the user to continue.

$\mathcal{IDS}_2$: The IDS detects a legitimate action as an attack and stops it (false positive alarm).

$\mathcal{IDS}_3$: The IDS does not detect an attempt of an attack and the user successfully completes it (false negative alarm).

$\mathcal{IDS}_4$: The IDS detects an attempt of an attack and stops it.

The IDS's preferences match the action profiles of the game, as specified below:

$\mathcal{IDS}_1 \rightarrow (L, C), (E, C)$

$\mathcal{IDS}_2 \rightarrow (L, P), (E, P)$

$\mathcal{IDS}_3 \rightarrow (A, C), (E, C)$

$\mathcal{IDS}_4 \rightarrow (A, P), (E, P)$

An IDS has been designed and implemented to detect attempts of attacks in real time. Therefore, the most preferable outcome of the game must be to detect attempts of attacks and stop them. The next preferable is to allow legitimate users to continue their work. Regarding the other two preferences, the IDS prefers stopping incorrectly a legitimate action rather than allowing an attack to be accomplished. The ranking of these preferences is presented in the ordered list below:

$$\mathcal{IDS}_3 \prec \mathcal{IDS}_2 \prec \mathcal{IDS}_1 \prec \mathcal{IDS}_4 \tag{4.4}$$

We will define the utility function for the IDS, based on the set of preferences $\mathcal{IDS}$ and the defined preference relations. Suppose that $U_{\mathcal{IDS}} : \{\mathcal{IDS}_1, \mathcal{IDS}_2, \mathcal{IDS}_3, \mathcal{IDS}_4\} \rightarrow \mathbb{R}$ is the utility function of the IDS. With regard to its preferences, an IDS dislikes preference $\mathcal{IDS}_3$ and prefers mostly $\mathcal{IDS}_4$. For this reason, we define $U_{\mathcal{IDS}}(\mathcal{IDS}_3) = 0$ and

$U_{\mathcal{IDS}}(\mathcal{IDS}_4) = 1$, respectively. If we select $\mathcal{IDS}_2$ as one intermediate between $\mathcal{IDS}_2$ and $\mathcal{IDS}_1$ which are left, we define its utility as $U_{\mathcal{IDS}}(\mathcal{IDS}_2) = \frac{1}{2}$. Finally, we calculate $U_{\mathcal{IDS}}(\mathcal{IDS}_1)$, following the same reasoning as we did before for the Normal User and for the Attacker, and we define $U_{\mathcal{IDS}}(\mathcal{IDS}_1) = \frac{3}{4}$. In Table 4.3 below, the second row summarizes the defined utilities for the IDS. The third row describes the corresponding utilities free of fractions, after multiplying them by 4.

x	$\mathcal{IDS}_3$	$\mathcal{IDS}_2$	$\mathcal{IDS}_1$	$\mathcal{IDS}_4$
$U_{\mathcal{IDS}}(x)$	0	$\frac{1}{2}$	$\frac{3}{4}$	1
$4 \cdot U_{\mathcal{IDS}}(x)$	0	2	3	4

Table 4.3: IDS's Utility Function

An IDS with different configuration, which has been tuned in order to reduce, for example, the large number of false positive alarms it generates, might have another preference ranking to reflect its settings and operation. In another example, for a Target System with many internal attackers, the IDS should be tuned appropriately to detect them. The IDS's preferences then change its second and third choice to be detection of a legitimate action as an attack ($\mathcal{IDS}_2$) and then no detection for normal activity ($\mathcal{IDS}_1$), respectively. This alternative ranking of the IDS will have the form below:

$$\mathcal{IDS}_3 \prec \mathcal{IDS}_1 \prec \mathcal{IDS}_2 \prec \mathcal{IDS}_4 \tag{4.5}$$

Mixed Actions

Besides the pure actions, there are also mixed actions available to players. Players use mixed actions when they want to introduce randomness into their behavior [135]. The set

of probability distributions on the action set A_i of player i is denoted by $\Delta(A_i)$. In general, α_j indicates a mixed action. For player U, the set of mixed actions is

$$\Delta(A_U) = \{\alpha_{U_1}, \alpha_{U_2}, \ldots, \alpha_{U_k}\},\ k = 1, 2, \ldots$$

and for the three available actions discussed above it turns into

$$\Delta(A_U) = \{\alpha_{U_1}, \alpha_{U_2}, \alpha_{U_3}\}, \alpha_{U_j}\ \epsilon\ [0, 1]\ s.t.\ \alpha_{U_1} + \alpha_{U_2} + \alpha_{U_3} = 1.$$

An example would be the set $\Delta(A_U) = \{\frac{1}{3}, \frac{1}{5}, \frac{7}{15}\}$, which indicates that player U will choose the action L with probability $\frac{1}{3}$, the action A with probability $\frac{1}{5}$, and the action E with probability $\frac{7}{15}$.

Likewise, the mixed actions for the IDS is indicated as,

$$\Delta(A_{IDS}) = \{\alpha_{IDS_1}, \alpha_{IDS_2}, \ldots, \alpha_{IDS_m}\},\ m = 1, 2, \ldots$$

and for the two corresponding actions it is defined as,

$$\Delta(A_{IDS}) = \{\alpha_{IDS_1}, \alpha_{IDS_2}\}, \alpha_{IDS_j}\ \epsilon\ [0, 1]\ s.t.\ \alpha_{IDS_1} + \alpha_{IDS_2} = 1.$$

As an example we consider the set $\Delta(A_{IDS}) = \{\frac{3}{4}, \frac{1}{4}\}$, which indicates that player IDS will choose the action C with probability $\frac{3}{4}$, and the action P with probability $\frac{1}{4}$.

The set of mixed profiles is the combination of mixed actions, defined as the Cartesian product of the probability distributions,

$$M \equiv \prod_{i \epsilon N} \Delta(A_i).$$

In this game, the set of mixed profiles is defined as $\Delta(A_U) \times \Delta(A_{IDS})$. In addition, the utility function includes expectations, to incorporate the mixed actions. The set of payoffs, generated by the utility function for the pure action profiles of A, is given as in [115] to be,

$$\mathcal{F} \equiv \{v\ \epsilon\ \mathbb{R}^2 : \exists\ a\ \epsilon\ A\ s.t.\ v = u(a)\}.$$

Among these payoffs, there are some feasible payoffs, the payoffs currently available. In particular, the set of feasible payoffs is the convex hull of the set of payoffs,

$$\mathcal{F}^{\dagger} \equiv \text{co}\mathcal{F}.$$

A set is convex if it contains the line segment joining two points whenever it contains them. The convex hull is the smallest convex set that contains the initial set [21].

4.1.2 The Repeated Game Model

The game model described above is being repeatedly played again and again in periods of time $t \, \epsilon \, \{0, 1, 2, \ldots\}$. The time period $t_x, \; x = 0, 1, 2, \ldots$ denotes the time when the x turn of the play has been finished, that is, the t_1 period indicates the time the game has been already played once. The number of times a game will be played depends on the players as well as on the type of user. If the user violates the security policy of the TS, then the IDS will prevent further usage of the TS and the game will end unexpectedly. If the user is an internal user, then the game lasts for a long time, as long as the user is an employee of the organization holding the TS. Such a game is a finite repeated game, but no player knows how many times they will play it, if the current play is the last one, or when it will stop in the future; it is an indefinitely repeated game.

In a repeated game, the choices made by the players are called strategies to distinguish them from the options in the stage game called actions. According to the IDS's view, at the end of a time period t_x, the IDS is able to monitor the action profile played before in periods $t_0, t_1, t_2, \ldots, t_{x-1}$. In particular, via a security auditing module that logs events and filters audit data, it is informed about the history of the play to this point. Therefore, the IDS perfectly monitors the actions of the other players, i.e. the user. Quite the opposite, at the end of a time period t_x, the user is hardly informed about what the IDS has chosen so far, or even if such a mechanism is operating. In closing, we consider this repeated game as a perfect monitoring game, because we take into account the IDS's point of view.

In a period of time $t_x,\ x = 0, 1, 2, \ldots$, the history of action profiles is a set containing all actions previously played, denoted as

$$\mathcal{H}^{t_x} \equiv A^{t_x},\ x = 0, 1, 2, \ldots.$$

As explained above, at period t_0 the game has not yet played at all. Consequently, the history at period t_0 is the initial history of the repeated game $\mathcal{H}^{t_0} \equiv A^{t_0} \equiv \{\varnothing\}$. Correspondingly, the infinite sequences of action profiles $(a^t)_{t=1}^{\infty}$ is denoted by A^{∞}. At any time period t_x, there is a list of x action profiles that assemble a history $h^{t_x}\ \epsilon\ \mathcal{H}^{t_x}$, that is to say, the history contains the actions played before, in periods $t_0, t_1, \ldots, t_{x-1}$. Likewise, the next time period, t_{x+1}, the history $h^{t_{x+1}}$ will contain $x + 1$ actions and the history set will be $\mathcal{H}^{t_{x+1}} = \mathcal{H}^{t_x} \times A$. Then, all possible histories of the infinitely repeated game is the union of the histories $\mathcal{H}^{t_x}$ at each time period $t_0, t_1, \ldots, t_x$, denoted as the set

$$\mathcal{H} \equiv \bigcup_{x=0}^{\infty} \mathcal{H}^{t_x}. \tag{4.6}$$

In repeated games, we call *strategies* the actions played in each period, to indicate the strategic planning the players adopt, because they repeat the stage game choosing from its actions' set. As the pure and the mixed actions have been explained and defined in Section 4.1.1, a pure strategy for a player of the ID game is a mapping from all possible histories $\mathcal{H}$ defined in (4.6), into the corresponding set of pure actions $A_i = \{a_{i_1}, a_{i_2}, \ldots, a_{i_k}\},\ i = \{U, IDS\},\ k = 1, 2, \ldots$ of this player, whereas, a mixed strategy is a mapping to the set of mixed actions $\Delta(A_i) = \{\alpha_{i_1}, \alpha_{i_2}, \ldots, \alpha_{i_k}\},\ i = \{U, IDS\},\ k = 1, 2, \ldots$. The same notation σ is used for a mixed strategy, which is equivalent to a behavioral strategy, and a pure strategy because it is a case of a behavioral strategy. The pure strategies and the mixed strategies are defined as follows:

$$\sigma_i : \mathcal{H} \rightarrow A_i,\ i = \{U, IDS\}. \tag{4.7}$$

$$\sigma_i : \mathcal{H} \rightarrow \Delta(A_i),\ i = \{U, IDS\}. \tag{4.8}$$

Then, we denote by $a(\sigma)$ the outcome of a pure strategy profile $\sigma \equiv (a^{t_0}(\sigma),\ a^{t_1}(\sigma),\ a^{t_2}(\sigma), \ldots)$ as the infinite sequence of action profiles with $a^{t_x}(\sigma),\ x = 0, 1, 2, \ldots$ to be the action profile played in period t_x. Since in a period t_x the corresponding payoff for the pure action profile $a^{t_x}(\sigma)$ is $u_i(a^{t_x}(\sigma))$, where $i = \{U, IDS\}$, then an outcome $a(\sigma)$ for a player i is an infinite stream of the payoffs determined in the stage game, and is given by $(u_i(a^{t_0}(\sigma)), u_i(a^{t_1}(\sigma)), u_i(a^{t_2}(\sigma)), \ldots)$. At every period, a payoff to a player i is discounted by a discounted factor $\delta \in [0, 1)$. To calculate the average discounted payoff for this player from the infinite stream of payoffs $(u_i^{t_0}, u_i^{t_1}, u_i^{t_2}, \ldots)$, we use the following formula

$$(1 - \delta) \sum_{t=0}^{\infty} \delta^t u_i^t. \tag{4.9}$$

Likewise, to calculate the average discounted payoff from a pure strategy profile σ, we use the next formula

$$U_i(\sigma) = (1 - \delta) \sum_{t=0}^{\infty} \delta^t u_i(\alpha^t(\sigma)). \tag{4.10}$$

In the following discussions we will use formula (4.10) for mixed and behavioral strategies as well.

4.2 Repeating with Imperfect Monitoring

In imperfect monitoring, the players of the game are not very well informed about each others' moves. Even if a user knows the existence of an IDS, he might not know what the IDS does at every move. Likewise, an IDS is not well informed when it does not identify a user's move, which means that the IDS is not reliable and has a detection rate less than

100%. Therefore, it raises a number of false alarms, because it is not accurate, as discussed in Section 1.1.4.

In repeated games with imperfect monitoring we examine signals from the actual actions that have been played in the previous period. When these signals are observable by all the players of the game, then the game is *repeated with imperfect public monitoring*, otherwise, the game is *repeated with imperfect private monitoring*.

We consider the ID game as a repeated game with imperfect public monitoring with the specifications of Section 4.1 of the perfect monitoring. In addition, there is a space Y of signals, and at the end of a period $t_x,\ x = 0, 1, 2, \ldots$, players observe a public signal y that derives from space Y. Consequently, the history h^{t_x} of the public signals $(y^{t_0}, y^{t_1}, \ldots, y^{t_{x-1}})$ is the only public information available in a period t_x, and the set of these public histories can be defined as

$$\mathcal{H} \equiv \bigcup_{x=0}^{\infty} Y^{t_x}. \tag{4.11}$$

Player IDS is the long-run player of the ID game, and therefore, the history of the IDS is given by

$$\mathcal{H}_{IDS} \equiv \bigcup_{x=0}^{\infty} (A_{IDS} \times Y)^{t_x}. \tag{4.12}$$

The ID game with imperfect public monitoring is a situation related to the second thesis hypothesis stated in Section 1.5. On the other hand, the ID game with imperfect private monitoring is outside the study of this research work, because games with imperfect private monitoring require different techniques and raise a significant number of questions.

4.3 Summary

Following the generic ID game model of Chapter 3, we examined the ID game when playing repeatedly. In compliance with the two thesis hypotheses of Section 1.5, we formulated the ID game as a perfect monitoring repeated game for an IDS that has 100% detection rate (Hypothesis 1), and as an imperfect monitoring repeated game for an IDS with less than 100% detection rate (Hypothesis 2). The specifications, the definitions, the notation, and the discussions constitute the basis for game constructions and their solutions in the following chapters.

Chapter 5

Insiders and their Games

> Life is nothing but a competition to be the criminal rather than the victim.
>
> Bertrand Russell (1872 - 1970)

In this chapter, we examine a special class of users, the internal attackers, also known as *insiders*. We clarify this class by specifying what an insider is, what an insider does, how risky is an insider's activity, and how we can reduce this risk. We define four specific actions for an insider and we discuss the reasons his activity is grouped in this action set.

Then, we construct a specific game between an insider and an IDS, to validate the functioning of the generic ID game model presented in Chapter 3. For the IDS, we define another set of four actions. Both action sets correspond to those of the generic model. The strategies and outcomes, the preferences and payoffs are all defined and examined systematically, following specific methodologies to get valuable results. Next, we consider the infinite rounds, and we provide the game definitions with perfect and imperfect information.

Based on the repeated game presented in Chapter 4, we solve the stage and the infinitely repeated game, in a step-by-step approach, and we explore scenarios in two different cases. Finally, we construct another game to be played with an unconventional insider, and we compare it with the first one.

5.1 Introducing the Insider Threat

In between 214-212 BC, Archimedes achieved to protect Syracuse for several months from being conquered by the Romans. His novel inventions defended effectively Syracuse, until a Syracusan traitor opened a gate, allowing Romans to seige the city. The insider threat is as old as any dispute in this world. Between two conflicting parties, there might appear someone in one of the parties, who will turn traitor, pursuing his own interests. Under these circumstances, any defending mechanism might proved inadequate to protect an establishment.

The 2007 E-Crime Watch Survey [54], which was conducted by the CSO magazine in cooperation with the U.S. Secret Service, the Carnegie Mellon University Software Engineering Institute"s CERT® Program and the Microsoft Corp., reported that the pie of damage caused by attacks is divided more or less equally between insiders (34%), outsiders (37%), and unknown (29%). Moreover, in 31% of the organizations the insiders have used special tools (password crackers, sniffers, etc.), whereas in the previous year it was only 17%. Unfortunately, the problem is likely to increase due to the great recession that generated thousands of fired employees, and put at risk others' jobs too.

5.1.1 Specifying an insider

The user of a system might have intentions and objectives that are opposed to those of the belonged organization. Such a user is a potential internal attacker of the system, also called an *insider*. Salem et al. [156] define an insider to be a malfeasant user that falls in one of two categories; traitors or masqueraders. Traitors are authorized users with specific privileges to use a system that belongs to an establishment; they exploit their accorded rights to achieve their goals and violate the security of the system, by affecting the confidentiality, integrity, and availability of its resources. Masqueraders are those who steal another user's identity, and by pretending that a legitimate user acts, they harm the system.

It is generally accepted that, an employee can become an insider because he is dishonest, or disgraceful, or dissatisfied, or disappointed, or following a conflict with the employer. Other settings include a fired employee or a retired one. But the list expands when the organization employs also part timers, some of them for short periods, and afterwards it sends them home; their frustration for being unemployed can turn them against the organization.

Depending on the size, the structure and the regulation state of the organization, auditors, consultants, customers, suppliers, and business partners might also have access rights to the information and communication systems owned by the organization. The reason is that they have to participate, interact, and accomplish their goals as external entities related to the organization. But, many of them might have conflicting interests with the organization, mainly because they also interact with other organizations at the same time, which may be competitors.

Although the list of the mentioned involved parties is long, there is a small number of common features between them; for example, the intention to harm the system, to cause damage, to increase individual profit, and in some cases just revenge. Apparently, it is hard to detect insiders, primarily because of their privileges that partially protect them from being caught. Such an endeavor usually generates a great number of false positive alarms, and a significant number of false negative alarms. Nevertheless, it is feasible to monitor their actions and whenever one violates the security policy of the system, then the system itself should react to prohibit such deviations.

The most up-to-date guide for the prevention and detection of insider threats was reported in January 2009 [34]. It includes sixteen practices, all based on real cases examined by CERT® that would assist early detection and prevention of insider incidents. Among them, it is practice 12 that suggests logging and monitoring of employee online actions.

5.1.2 Insider activity and actions

Randazzo et al. [151] report the findings of a study conducted in the banking and finance sector, regarding insider activity. Each of the 26 cases was examined from both behavioral and technical perspectives simultaneously. The findings of this study revealed that a) most incidents required little technical sophistication, b) the insiders planned their actions in advance, c) their motivation was mostly financial gain, d) they did not share a common profile, e) they were detected by several manual and automated methods and by a range of people, not just by security staff, f) the organization's loss was financial, and g) the insiders acted while working during normal business hours. In addition, Brackney and Anderson [28] provide a useful insider actions taxonomy, cross-referenced with vulnerabilities and exploits list.

Examining the ways an internal attacker of an organization might act, we define a set of four distinct actions. Normally, he acts in accordance with his commitments and duties, but occasionally he makes mistakes. If he is a naive user, then these mistakes might threaten the system as much as the actions of a real attacker. But, any user mistake is closely connected with the design and implementation of the system where problems can be located. Program bugs, incomplete program tests, luck of field validation are a few symptoms that usually lead to security relevant incidents.

The intentions of an insider however are malicious, and therefore, he plans the ways he can attack the system for his own purposes. So, he also acts systematically to prepare attacks. These actions can be characterized as actions of a pre-attack phase. Finally, an insider's actions can be included in the phase where real attack actions are executed following a plan. Summarizing these four actions, an insider either acts normally (N), or makes mistakes (M), or acts at a pre-attack phase (P), or attacks the system (A). The construction of the ID game for an insider in Section 5.2 is based on the definition of these four actions.

5.1.3 Measuring the insider risk

Considering the amount of threat each of the four insider action categories endangers a system, the (N) actions might severely threaten a system, especially when the insider is a high privileged user. For example, an insider that holds administrative permissions over some system resources might harm the system by acting 'normally', if he wishes.

An analogous amount of threat put at risk a system's security, when an insider performs (M) actions. For instance, an inexperienced user might cause resource exhaustion, by sending a mail message with an outsized attached file, to a lengthy recipients list. Recent reports identify the problem of mistakes caused by naive users as a security problem. It is the first time, the SANS Institute included human errors in its list [158], under the title "H1. Excessive User Rights and Unauthorized Devices". In 2003, a survey conducted by CompTIA showed that human errors cause many security breaches [40].

The problem becomes worse when the real identity of the naive user is an insider. In this case, if he realizes the harm of his (M) action, he would put no effort to reverse his mistake or to stop the expansion of the damage, as this is compliant with his intentions. Besides, such a mistake gives him excuses and good reasons to explain future (P) and (A) actions.

Pre-attack actions are special because they feed insiders with valuable information (e.g. phishing). As a (P) action might reveal private information, one of the three IT security principles, the confidentiality, would be affected. But an insider might already have legitimate access rights to several pieces of information, also classified information. For this reason, he might disclose private information of his organization, in order to obtain other pieces of information, necessary for him to complete an attack. In the literature, (P) actions are also known as reconnaissance attacks and are considered as critical for the security of a system and difficult to be detected. Moreover, social engineering attacks might be included in this category, when for example, an employee achieves to obtain valuable information

from an unsuspicious coworker of him.

Finally, (A) actions weights depend on the consequences created and the extent of the problem generated by them. Taking into account a number of other preceding actions, (N), (M), or (P) actions, with the effects and their significance described above, (A) actions are without doubt the most threatening actions among all. It is also the type of attacker, the internal attacker, that increases and strengthens the severity of the problem and the extend of the unexpected results. Therefore, detecting insiders becomes a demanding task, sometimes without findings.

For instance, in 2005 in Greece, a serious scandal revealed that at least 100 state cell-phones were being bugged and their conversations were probably being recorded for a long time. Among them it was the phones of the prime minister, the minister of national defense, and the minister of foreign affairs that were compromised. Although thorough investigations took place, no one has been accused for this crime until today. Only the company, one of the leading mobile telecommunications companies in the world, has been penalized with € 76 million, because of its fatal errors, oversights, bad handling, and faulty reactions when the interception was realized. Perhaps it was a great inside job [150].

Recently, the growing interest in studying insiders showed that a consistent definition is required [27]. Matt Bishop et al. have devoted an entire chapter in [24] to define "insider" precisely, to assess the associated threats, and to examine the classification of the involved entities (subjects and resources), in order to determine the risk of insider attacks and the consequences that derive when they happen. They use an access-model, the Attribute-Based Group Access Control Model (ABGAC) [25, 26], a generalization of Role-Based Access Control (RBAC). In the ABGAC model subjects and resources are both grouped into sets defined by attributes of interest. An organization can keep an ordered list of users who can severely damage it, by using the ABGAC model that identifies those users with access to resources with high value and significant information.

5.1.4 Reducing the risk

Although the insider threat has been recognized as a widespread problem, the lack of specific solutions to prevent, detect, and deter insiders is apparent. Only post attack actions, using forensics analysis, seem to be the most favor counteraction to confront the problem. These are some of the findings of a recent professional survey on insider attack detection that can be found in [156].

In particular, the survey shows that the insider attack detection research has been divided in two parts, the host-based user profiling and the network-based sensors. In the first part, the focus of research is mainly the command sequence analysis. The majority uses a UNIX®[1] system for implementation and testing, whereas few only have been applied to Windows®[2] environments. Most approaches utilize data collected either from user commands, with or without arguments, or from system calls. The target group of detection is masqueraders that make use of a UNIX® system commands. User profiling in web environments and program profiling approaches have been widely modeled in the literature.

Likewise, network based-sensors approaches have been used for traitor detection, and there is also a small number of honeypot-based detection methods. It is assumed that network-level audit sources are more suitable in detecting violations of need-to-know policies. Early attempts to integrate both divisions in a hybrid approach have insufficient documentation and reporting, to show the pros and cons of such an effort.

As a final point of this survey, it is acknowledged that, the most significant prospect, in the area of insider threat detection, is the modeling of user profiles that uncovers user's intention. Nevertheless, to distinguish a harmful action from a benign one relies on the security policy of the system and any violation that affects it. Besides, the lack of real data to effectively test the proposed solutions, in accordance with the uncertain utilization of

[1] UNIX® is a registered trademark of The Open Group.

[2] Windows® is a registered trademark of Microsoft Corporation in the United States and other countries.

them, is evidence for the need of new methods, in this open field of intrusion detection.

5.2 Constructing the ID Game with an Insider

We formulate the interactions between a user and an IDS as a 2-player non-cooperative game. We assume that the user, player I, is an internal attacker, as given by the insider's description above. His action set includes four action types, N for normal actions, M for mistaken actions, P for actions at a pre-attack phase, and A for attack actions, as already defined in Section 5.1.2. Following the notation given in Chapter 4 for the generic ID game model, the set of pure actions for player I is

$$A_I = \{a_{I_1}, a_{I_2}, a_{I_{3,a_{I_4}}}\} = \{N, M, P, A\}.$$

The second player of the game, player D, is an Intrusion Detection System (IDS), also called Detector, installed in the Target System (TS) that player I is using. Assuming that the IDS is a nice machine, in a sense that it makes the first move with the intention to cooperate, it decides among four alternatives. First, it allows the user to continue if nothing suspicious has been noticed; second it makes a recommendation whenever slight deviations are encountered; third it raises a warning to remind the user to be consistent with their agreement on the regulations of using the TS; and fourth it stops the user when a violation is detected.

Summarizing, player D has four actions too, C for allowing the user to continue, R for making recommendations, W for raising warnings, and S for stopping the user. Likewise, the set of pure actions for player D is

$$A_D = \{a_{D_1}, a_{D_2}, a_{D_3}, a_{D_4}\} = \{C, R, W, S\}.$$

Although for purposes of simplicity and appearance the generic ID game model implies a black and white representation, one of the things we examined in the ID game with an insider

was the action set of each player. For player I the first action N apparently corresponds to a legal action, while the rest of the actions are equivalent to an illegal action, as presented schematically below.

$$\underbrace{\mathbf{N}ormal}, \quad \underbrace{\mathbf{M}istake, \quad \mathbf{P}re-attack, \quad \mathbf{A}ttack}$$

$$\downarrow \qquad\qquad \downarrow$$

$$\mathbf{L}egal \qquad\qquad \mathbf{I}llegal$$

Likewise, for player D the first three actions correspond to a permission to continue, whereas the latter one, the S is equivalent to prevent, as depicted in the following schema.

$$\underbrace{\mathbf{C}ontinue, \quad \mathbf{R}ecommendation, \quad \mathbf{W}arning,} \quad \underbrace{\mathbf{S}top}$$

$$\downarrow \qquad\qquad \downarrow$$

$$\mathbf{C}ontinue \qquad\qquad \mathbf{P}revent$$

Depending on the type of player I's action, the IDS might remove privileges, decrease the amounts of certain allocated recourses, consider the insider as a user under supervision, or add him in a black list. We assume that a recommendation is linked with less severe counteractions than a warning. Nevertheless, player D monitors player I as a user under supervision when it makes a recommendation and places him in a black list. In the same way, when player D raises a warning, then it might remove privileges and decrease amounts of recourses allocated to player I.

Another thing we have examined is the different severity when this is assigned to the same counteraction, but against different actions. According to this, a warning varies from lenient to more strict, depending on which action it is raised, i.e. a mistake raises a relaxed warning, whereas a pre-attack action raises a more strict one. So, the amount of resources to be decreased and the privileges to be removed are adjusted to the corresponding action.

This has been taken into account when specifying the preferences in Section 5.2.2 and is quantified using discrete payoffs (see Figure 5.1).

We also assume that player I is perfectly informed of player D's past moves, because of the nature of his opponent's actions (recommend, warning, stop). Consequently, player D's actions are perfectly observable by player I. On the contrary, past choices of player I are imperfectly observed by player D, because player D classifies its opponent's actions under uncertainty. Situations with one-sided imperfect information are referred to as *imperfect monitoring*, as discussed in Section 4.2.

Following the dynamic nature of the generic ID game model clarified in Chapter 3, and due to the dynamic interactions that take place also between an insider and an IDS, we also model this game in an extensive form. Figure 5.1 depicts the one-shot game, also called the stage game, using the Gambit's [119] tool illustration style.

5.2.1 Strategies and outcomes

Extensive form games are portrayed by trees (see Figure 5.1). Player I moves first at the initial node (the root) of the game, denoted by a red circle. The player's name is displayed above the node. Below the node, the default labeling is the information set's number. It is a unique identifier of the information set, in the form *player number* : *information set number* (e.g. 1:1 means the first move of the first player, i.e. the first move of player I). Von Neumann defined information sets to model the progressive learning of which decisions will actually be made [21].

Similarly, player D's moves start at blue circles, above which there is a D standing for its name, and below, a corresponding pair, labeling its information set's number (2:1 means that the second player, player D, moves for the first time). Zeros and ones below each branch indicate the solution of the game, as will be described in Section 5.4. Zero means that an action will not be chosen (probability 0), and one means that an action will

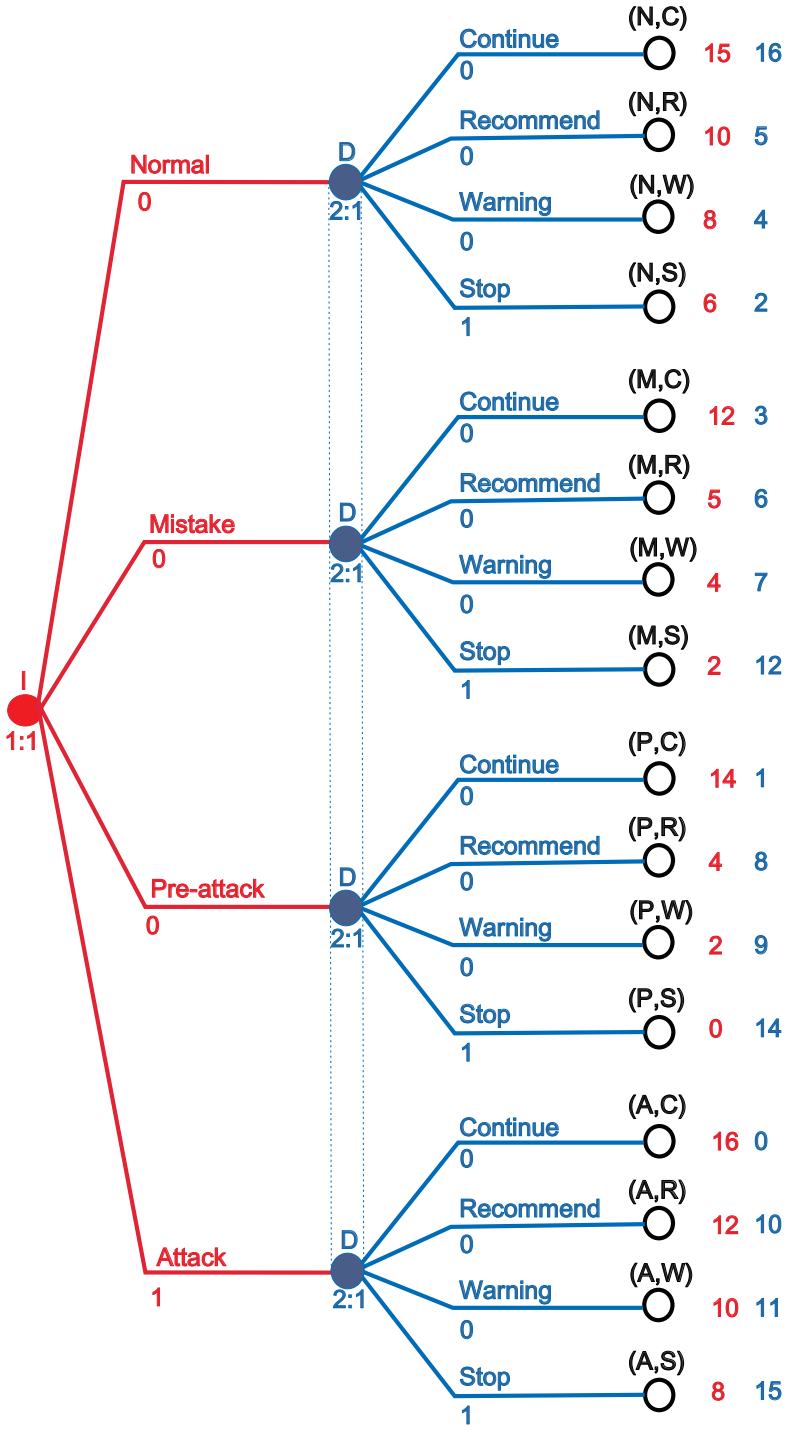

Figure 5.1: An extensive form game between an insider and an IDS.

be selected with certainty (probability 1).

We have assumed player D is not totally certain that player I has chosen one of the actions included in his action set. This is consistent with the hypothesis that there is no detection engine with 100% detection rate. Thus, player D's sub-trees belong to the same information set, connected with a dotted line to indicate this. In short, the dotted line connects player D's nodes to indicate the IDS accuracy, and thus, the degree of uncertainty whether player I has chosen an (N), (M), (P), or (A) action.

Looking at the ends of the branches, 16 outcomes are identified. The number of outcomes derives from all the possible combinations between the insider's actions and the IDS's actions $(4 * 4)$. Following the corresponding descriptions of Chapter 4, it is the set of action profiles of player I and player D defined as the Cartesian product,

$$A \equiv \prod_{i \epsilon N} A_i \equiv A_I \times A_D \tag{5.1}$$

which is

$$\begin{aligned} A \equiv \{&(N,C),(N,R),(N,W),(N,S),(M,C),(M,R),(M,W),(M,S),(P,C),\\ &(P,R),(P,W),(P,S),(A,C),(A,R),(A,W),(A,S)\} \end{aligned} \tag{5.2}$$

There is a pair of capital letters at the end of each branch and above the node that denotes player I's and player D's choices, respectively. This is an action profile. For example, (N,R) means that player I has chosen an (N) action while player D made a recommendation.

Finally, the pair of numbers next to each end node is the pair of players' payoffs, that is to say, the outcome a player receives when a certain action has been chosen, represented as a number (the red number belongs to player I and the blue number belongs to player D). In Section 5.2.2, we explain step by step the procedure employed to quantify the outcomes of this game.

The players play the game repeatedly an infinite number of times. The reason is that, the user is not a random attacker, but an internal user of the system, who spends a long time every day. We assume he is a traitor than a masquerader. As explained in Chapter 4, in repeated games, we call the actions strategies to distinguish them from the actions in the stage game.

The following list describes the meaning of the outcomes and gives the notation of all the possible strategy profiles, one strategy for each player:

1. The insider acts normally, and the IDS allows him to continue, *(N,C)*.

2. The insider acts normally, but the IDS makes wrongly a recommendation to him about his actions (false positive alarm), *(N,R)*.

3. The insider acts normally, but the IDS sends incorrectly a warning to him about his actions (false positive alarm), *(N, W)*.

4. The insider acts normally, but the IDS stops him from using the system, because, it erroneously classified his actions as attacking (false positive alarm), *(N,S)*.

5. The insider makes mistakes, but the IDS allows him to continue (false negative alarm), *(M,C)*.

6. The insider makes mistakes, and the IDS recommends him to avoid errors, *(M,R)*.

7. The insider makes mistakes, and the IDS sends a warning to him about his actions, *(M, W)*.

8. The insider makes mistakes, and the IDS stops him from using the system, because it classified his actions as threatening, *(M,S)*.

9. The insider acts at a pre-attack phase, but the IDS allows him to continue (false negative alarm), *(P,C)*.

10. The insider acts at a pre-attack phase, and the IDS recommends him to stop doing so, *(P,R)*.

11. The insider acts at a pre-attack phase, and the IDS sends a severe warning to him about his actions, *(P,W)*.

12. The insider acts at a pre-attack phase, and the IDS stops him from using the system, because it classified his actions as the preparation of an attack, *(P,S)*.

13. The insider uses attacking actions, but the IDS allows him to continue (false negative alarm), *(A,C)*.

14. The insider uses attacking actions, and the IDS recommends him to stop doing so, *(A,R)*.

15. The insider uses attacking actions, and the IDS sends a severe warning to him about his actions, *(A,W)*.

16. The insider uses attacking actions, and the IDS stops him from using the system, because it classified his actions as an attack, *(A,S)*.

5.2.2 Preferences and payoffs

To quantify the outcomes of the game, we first specify preferences over outcomes, and then we use the von Neumann-Morgenstern utility function. We use the same method [21] as in Chapter 4. A player prefers a strategy over another, because he gains more or he loses less. We use the symbol $\prec$ to denote preference and the symbol $\sim$ to denote no interest, i.e. indifference. For instance, if $a \prec b$ then it is said that b is preferred to a.

For an insider, the most desirable is to successfully attack the system without being stopped, or even caught. His second best is to act normally without being stopped. Similarly, the pre-attack actions follow, both without any deterrence, and attack actions that

get a recommendation is his next preference. Mistake actions that are not prevented are also indifferent, and more attractive than attack actions that raise a warning. These are indifferent too from normal actions that cause a recommendation. Attack actions that get stopped follow insider's favorites.

On the other hand, his worst choice is a pre-attack action followed by a stop action, then a mistake followed by a stop action too, which is also indifferent from a pre-attack action followed by a warning. A mistake that raises a warning is indifferent from a pre-attack action, followed by a recommendation. Likewise, a mistake that causes a recommendation is less preferable than a normal action that is being stopped, and than a normal action that raises a warning. Based on these lines of reasoning, we have assumed that the ranking of player I's preferences over outcomes, from the least preferable (PS) to the most preferable one (AC), gives the preference structure described below. We have dropped parentheses and commas from the pairs of choices, not to clutter the notation in the preference structures.

$$\begin{aligned} PS \prec_I MS \sim_I PW \prec_I MW \sim_I PR \prec_I MR \prec_I NS \prec_I NW \sim_I AS \prec_I \\ NR \sim_I AW \prec_I MC \sim_I AR \prec_I PC \prec_I NC \prec_I AC. \end{aligned} \tag{5.3}$$

For example, $PW \prec_I PR$ means that an insider prefers to get a recommendation for a (P) action than a warning. This is reasonable, because we have assumed that a warning might decrease allocated resources or remove privileges, whereas, a recommendation is a weaker reaction on behalf of player D. By removing privileges and resources, the insider might not be able to complete his plan and attack the system.

As already explained in Chapter 4, a preference relation $\preceq$ on the action set A of a player i specifies a binary relation, represented by a payoff function $u_i : A \to \mathbb{R}$, the von Neumann-Morgenstern utility function. For two pure actions a_1 and a_2, $u(a_1) \leq u(a_2)$, whenever $a_1 \preceq a_2$ [135]. We define the utility functions for the insider and the IDS respectively, and we use the values of these functions as the payoffs of the game.

We assume that player I's preference is described by the utility function $u_I : A \to \mathbb{R}$, where A is the set of action profiles defined earlier in Equation (5.2). Following Binmore's method [21], we assigned numbers to reflect these preferences, and we constructed player I's utility function u_I, as described in Chapter 4. We set 0 to strategy PS because it is the least preferable, and 1 to strategy AC as insider's best choice. So, $u_I(PS) = 0$ and $u_I(AC) = 1$. Using rational numbers, we assigned a value to every strategy, according to the ranking described in expression (5.3). Then, we got the values free of fractions, after multiplying with their least common factor. The utility values for an insider, as defined by the function u_I, are finally displayed in Table 5.1.

x	PS	MS	PW	MW	PR	MR	NS	NW	AS	NR	AW	MC	AR	PC	NC	AC
$u_I(x)$	0	2	2	4	4	5	6	8	8	10	10	12	12	14	15	16

Table 5.1: Insider's Utility Function

Regarding player D's preferences, the ranking is ordinary. There are three general lines of reasoning, the first to stop any non legitimate action, the second to raise a warning or to make a recommendation in attack, pre-attack, mistake or normal actions in that order, and the third to allow mistakes, pre-attack actions, and attack actions. But, the most preferable strategy for player D is the normal action followed by a continue action, whereas, the least preferable is an attack action followed by a continue action.

Following then the same steps to rank player D's preferences, we have established player D's preferences over outcomes, from the least preferable (AC) to the most preferable one (NC), as in the subsequent preference structure:

$$\begin{aligned} AC \prec_D PC \prec_D NS \prec_D MC \prec_D NW \prec_D NR \prec_D MR \prec_D MW \prec_D PR \prec_D \\ PW \prec_D AR \prec_D AW \prec_D MS \prec_D PS \prec_D AS \prec_D NC. \end{aligned} \tag{5.4}$$

Likewise, we assume that player D's preference is described by the utility function $u_D : A \rightarrow \mathbb{R}$, where A is the set of action profiles defined earlier in Equation (5.2). Using Binmore's method [21] and assigning numbers to reflect the preferences, we constructed player D's utility function u_D, to quantify its preferences. Therefore, we set 0 to strategy AC because it is the worst choice, and 1 to strategy NC as the best choice. So, $u_D(AC) = 0$ and $u_D(NC) = 1$. Using rational numbers, we assigned a value to every strategy, according to the ranking described in expression (5.4). Then, we got the values free of fractions, after multiplying with their least common factor. The utility values for an IDS, as defined by the function u_D, are finally displayed in Table 5.2.

x	AC	PC	NS	MC	NW	NR	MR	MW	PR	PW	AR	AW	MS	PS	AS	NC
$u_D(x)$	0	1	2	3	4	5	6	7	8	9	10	11	12	14	15	16

Table 5.2: IDS's Utility Function

In Figure 5.1, at the end of each branch, there is a pair of numbers attached. It is the payoffs pairs. In each pair, the first number is player I's payoff, and the second is player D's payoff, as established above.

5.2.3 Infinite rounds

In reality, the stage game is being played again and again, one round every period $t \, \epsilon \, \{0, 1, 2, \ldots\}$. The time period t_i, $i = 0, 1, 2, \ldots$ denotes the time when the i turn of the play has been finished, that is, the t_1 period indicates the time the game has been already played once. This endless iteration forms the infinitely repeated game. The continuous play of the game causes the rapid grow of the tree that represents it. The number of terminal nodes where payoffs are attached is determined by the number of strategies that have been played to this point.

As specified in Chapter 4, in an infinitely repeated game, the total number of strategy profiles in a period $t_i,\ i = 0, 1, 2, \ldots$, is the product of the history of action profiles played at all periods $0, 1, 2, \ldots, t_{i-1}$, $\mathcal{H}^{t_{i-1}}$, and the actions to be played at this period t_i, A^{t_i}. That is,

$$\mathcal{H}^{t_i} \equiv \mathcal{H}^{t_{i-1}} \times A^{t_i},\ i = 0, 1, 2, \ldots.$$

Because in our game there is an action, player D's action S, where the game ends, the number of strategy profiles, i.e. the number of terminal nodes at a n round, cannot be calculated using the common formula for infinitely repeated games.

Therefore, at any period $t_i,\ i = 1, 2, 3, \ldots$ in our game, the number of strategy profiles is calculated, by multiplying the number of strategies of the previous period with the combination of actions of the two players, excluding the terminal action S, i.e. $4 \times 3 = 12$. This yields the recurrence defined next that specifies the number of strategies $\mathcal{S}$, at time period $t_i,\ i = 1, 2, 3, \ldots$, as derived from the previous time period t_{i-1}.

Definition 5. The number of strategies $\mathcal{S}$ at time period $t_i,\ i = 1, 2, 3, \ldots$ is given by the formula

$$\mathcal{S}_{t_i} = 12 \times \mathcal{S}_{t_{i-1}}, \qquad i = 1, 2, 3 \ldots. \tag{5.5}$$

As (5.5) expresses $\mathcal{S}$ in terms of itself, we strived to another solution. We develop a formula, as described in Sorite 1, to calculate the number of terminal nodes of the repeated game tree, at any period $t \in \{0, 1, 2, \ldots\}$. The new formula is not in closed form, but it is not recursive.

Sorite 1. *At period $t_i,\ i = 0, 1, 2, \ldots$, the total number of strategy profiles $\mathcal{S}_{t_i}$ of the repeated game is*

$$\mathcal{S}_{t_i} = (A_I \times A_D) \cdot (A_I \times A_D - 4)^i, \qquad i = 0, 1, 2 \ldots. \tag{5.6}$$

Proof. In the stage game, at period t_0, the total number of action profiles is the combination of actions defined in Equation (5.1) as the Cartesian product,

$$A \equiv \prod_{i \epsilon N} A_i \equiv A_I \times A_D,$$

where A_I is the set of pure actions for player I, and A_D is the set of pure actions for player D.

The number of strategies at a period is calculated by multiplying the number of strategy profiles of the previous period with the combination of actions of the two players. Given that there is an action that ends the game, excluding this terminal action, the total number of strategy profiles at the second period t_1 is

$$\mathcal{S}_{t_1} = \mathcal{S}_{t_0} \times 12.$$

Following this reasoning, the total number of strategy profiles at the third period t_2 is

$$\mathcal{S}_{t_2} = \mathcal{S}_{t_1} \times 12 = \mathcal{S}_{t_0} \times 12 \times 12 = \mathcal{S}_{t_0} \times 12^2.$$

Assuming that $\mathcal{S}_{t_0} = A_I \times A_D$ and $A_I \times A_D - 4 = 12$ hold, then by induction, we define Sorite 1. □

Both formulae (5.5) and (5.6) start at the second stage of the game, and give identical results. The first stage has 16 strategies. Consequently, calculating the number of terminal nodes using formulae (5.6), we get the following figures for the first five periods:

$$\mathcal{S}_{t_0} = 16$$

$$\mathcal{S}_{t_1} = 192$$

$$\mathcal{S}_{t_2} = 2304$$

$$\mathcal{S}_{t_3} = 27648$$

$$\mathcal{S}_{t_4} = 331776.$$

It is apparent that the game grows rapidly as the periods increase. When repeating the game many times, the above formulae would assist the estimation of the generated complexity. It is important to determine whether predicting insider's behavior, as described in the following chapters, is an NP-complete problem, and examine alternatives to reduce its computational complexity.

5.3 Defining the ID Game with an Insider

By completing the construction and the formal description of the ID game with an insider, we define the game in the sequel as an *infinitely repeated game* with *perfect* and *imperfect information*. The definitions follow the notation of [135] and derive from the corresponding definitions introduced in Chapters 3 and 4 for the generic ID game.

5.3.1 The ID Game with an Insider of Perfect Information

Definition 6. Let $G_{I_p} = \langle N, H, P, (\succsim_i) \rangle$ be an extensive game with perfect information that models the ID game with an Insider, where

- $N = \{I, D\}$ is the set of players;
- H is the infinite set of sequences that consists of the histories $\varnothing$, N, M, P, A,
  (N,C), (N,R), (N,W), (N,S), (M,C), (M,R), (M,W), (M,S), (P,C), (P,R),
  (P,W), (P,S), (A,C), (A,R), (A,W), (A,S), ...,
  (N,C,N), (N,R,N), (N,W,N), (M,C,N), (M,R,N), (M,W,N),
  (P,C,N), (P,R,N), (P,W,N), (A,C,N), (A,R,N),
  (A,W,N), ...,
  (N,C,M), (N,R,M), (N,W,M), (M,C,M), ...,
  (N,C,P), (N,R,P), (N,W,P), (M,C,P), ...,
  (N,C,A), (N,R,A), (N,W,A), (M,C,A), ...,

(N, R, M), ...,

...,

(N, C, N, C), (N, R, N, C), (N, W, N, C), (M, C, N, C), (M, R, N, C), (M, W, N, C),

(P, C, N, C), (P, R, N, C), (P, W, N, C), (A, C, N, C), (A, R, N, C),

(A, W, N, C), ...,;

- P is the player function that indicates the player who takes an action after a history ($P(h)$), i.e. $P(\varnothing) = I$, $P(I) = D$, and $P(D) = I$ for every $h \neq \varnothing$, that is to say for every nonterminal history;

- $\succsim_i$ is the preference relation on Z for player $i \in N$, that is, the preference relation of player I is $PS \prec_I MS \sim_I PW \prec_I MW \sim_I PR \prec_I MR \prec_I NS \prec_I NW \sim_I AS \prec_I NR \sim_I AW \prec_I MC \sim_I AR \prec_I PC \prec_I NC \prec_I AC$, and the preference relation of player D is $AC \prec_D PC \prec_D NS \prec_D MC \prec_D NW \prec_D NR \prec_D MR \prec_D MW \prec_D PR \prec_D PW \prec_D AR \prec_D AW \prec_D MS \prec_D PS \prec_D AS \prec_D NC$.

5.3.2 The ID Game with an Insider of Imperfect Information

Definition 7. Let $G_{I_i} = \langle N, H, P, f_c, (\mathcal{I}_i)_{i \in N}, (\succsim_i) \rangle$ be an extensive game with imperfect information that models the ID game with an Insider, where

- $N = \{I, D\}$ is the finite set of players;

- H is the infinite set of sequences that consists of the histories $\varnothing$, N, M, P, A,

(N, C), (N, R), (N, W), (N, S), (M, C), (M, R), (M, W), (M, S), (P, C), (P, R),

(P, W), (P, S), (A, C), (A, R), (A, W), (A, S), ...,

(N, C, N), (N, R, N), (N, W, N), (M, C, N), (M, R, N), (M, W, N),

(P, C, N), (P, R, N), (P, W, N), (A, C, N), (A, R, N),

(A, W, N), ...,

(N, C, M), (N, R, M), (N, W, M), (M, C, M), ...,

(N, C, P), (N, R, P), (N, W, P), (M, C, P), ...,

(N, C, A), (N, R, A), (N, W, A), (M, C, A), ...,

(N, R, M), ...,

...,

(N, C, N, C), (N, R, N, C), (N, W, N, C), (M, C, N, C), (M, R, N, C), (M, W, N, C),

(P, C, N, C), (P, R, N, C), (P, W, N, C), (A, C, N, C), (A, R, N, C),

(A, W, N, C), ...,;

- P is the player function that indicates the player who takes an action after a history ($P(h)$) (a member of $N \cup c$, where c is the chance player), i.e. $P(\varnothing) = I$, $P(I) = D$, and $P(D) = I$ for every $h \neq \varnothing$, that is to say for every nonterminal history;

- f_c is the function that associates with every history h for which $P(h) = c$ a probability measure $f_c(\cdot|h)$ on $A(h)$, each independent of every other such measure, e.g. $(f_c(a|h))$ is the probability that a occurs after the history h;

- $\mathcal{I}_i$ is the information partition of player $i, i \in N$ and a set $I_i \in \mathcal{I}_i$ is an information set of player i;

- $\succsim_i$ is the preference relation on Z for player $i \in N$, that is, the preference relation of player I is $PS \prec_I MS \sim_I PW \prec_I MW \sim_I PR \prec_I MR \prec_I NS \prec_I NW \sim_I AS \prec_I NR \sim_I AW \prec_I MC \sim_I AR \prec_I PC \prec_I NC \prec_I AC$, and the preference relation of player D is $AC \prec_D PC \prec_D NS \prec_D MC \prec_D NW \prec_D NR \prec_D MR \prec_D MW \prec_D PR \prec_D PW \prec_D AR \prec_D AW \prec_D MS \prec_D PS \prec_D AS \prec_D NC$.

5.4 Solving the ID Game when Playing with an Insider

We proceed to solve the game using equilibrium analysis. Transferring this game from the extensive form of Figure 5.1 to a strategic form, we get the following 4x4 matrix, as

presented in Table 5.3. It is also called the *normal form* of the game (see Section 1.2 for details). The row player is the insider I and the column player is the detector D, that is, an Intrusion Detection System, which protects a Target System.

		D			
		C	**R**	**W**	**S**
I	**N**	(15,16•)	(10,5)	(8,4)	(7,2)
	M	(12,3)	(5,6)	(4,7)	(2,12•)
	P	(14,1)	(4,8)	(2,9)	(0,14•)
	A	(16•,0)	(12•,10)	(10•,11)	(**8**•,**15**•)

Table 5.3: A game between an insider and an IDS in normal form

We located the solution of this game by examining players' best responses. According to this, we examine what a player chooses as a best response to the other player's choice. Therefore, if player I chooses N, then player D chooses C, because this is its best response in the row of the N action. By choosing C it gets 16, which is the maximum payoff when comparing to 5, 4, or 2. Similarly, we locate player D's best responses for all possible choices of player I, and we mark them with blue circles. Likewise, we locate player I's best responses and we mark them with red circles.

This procedure leads us to locate a unique Nash Equilibrium (NE) that corresponds to the strategy profile (combination) AS with payoffs (8,15). In other words, the static NE is (A,S) and the static NE payoff is (8,15). It is a perfect NE that reveals the intention of player I to attack the system, and the reaction of player D to stop him doing so.

Interestingly, there is another pair of strategies, the NC strategy profile with payoffs (15,16), which are both greater than the corresponding of the NE. Besides, payoffs (15,16) are absolutely the highest each player can get in this game. In fact, strategy NC Pareto

dominates[3] the NE, and because the corresponding payoffs are the highest, the NC strategy is Pareto efficient[4]. In other words, any player between the two can increase his outcome by deviating from the equilibrium path, which is the strategy AS, and choose the Pareto efficient dominant strategy NC, even for a while. Some time in the future, he might turn back to the NE path.

In a one shot game, real circumstances cannot be depicted nor examined in depth, to find realistic solutions. Especially in this game, the players play the game repeatedly infinite number of times, as examined thoroughly before in Section 5.2.3. The reason is that the user is not a random attacker, but an internal user of the system, who spends a long time every day using it. In the generic game model described in Section 3.3, parts of the repeated divisions have already been located. Repeated games usually have multiple NE, and then NE selection becomes a problem. Solution to this problem originated in 1988 by Harsanyi and Selten [69] and continued with the Refinements literature and the Evolutionary Game Theory.

In addition, the presence of Pareto efficient strategies in this game shows that the NE would not definitely be players' choice for ever. Considering a case where player I plays a certain strategy at every period repeatedly, there must be certain circumstances under which he would deviate from this equilibrium path, and he would decide to play another strategy.

In a real case with a patient insider, player I would follow the NC strategy first for a number of periods, and then he would choose the AS strategy. It is the time a user of the system will turn traitor. In a future round, player I would go back to the NC strategy again, deviating from the equilibrium path. This shows that player I in some periods follows, and in some others deviates from the equilibrium path. But, player D should react by choosing a 'punishment' strategy, to keep player I attached with the NC strategy. When a punishment

[3] A strategy Pareto dominates another strategy, if the outcome of the first is higher than the outcome of the latter one (Vilfredo Pareto, 1848-1923).

[4] A strategy is Pareto efficient if no other strategy yields all players higher payoffs [136].

strategy is being followed for ever, it is known as *grim strategy.*

5.5 Repeating the ID Game with an Insider

To solve this game as a repeated game we followed David Levine's step-by-step procedure, as described in [102]. First, we decided to use the *average present value* method to aggregate payoffs obtained among different periods. Alternatives would have been to add payoffs together, to average them, or to take the present value. Second, we clarified that this game can be repeated infinite number of times, as mentioned in Section 5.2.3. Finally, regarding the discount factor δ (see Section 4.1.2) that shows how much impatient a player is, we examined two different cases. In the one case, Case A, we defined a common discount factor δ for both players. The discount factor δ varies between zero and one, with zero to match an impatient player and one a patient one. In this case, the internal attacker is a patient player, because he has plenty of time to organize and execute an attack. In addition, the IDS is inherently a patient player, because it plays infinitely with a user of the TS, although it does not know that he is an internal attacker. But in the other case, Case B, assuming that player I is about to be fired, we distinguished two different discount factors, δ_1 for the short-run player I of the game, and δ_2 for the long-run player D.

Following Levine's method, we start with the assumption that a repeated game is the iteration of the *stage game* (see Section 4.1.1), known also as the *static game.* We use the stage game between an insider and an IDS in normal form depicted in Table 5.3, and the best responses located previously for player I (red circles) and player D (blue circles), respectively. As solved in Section 5.4, there is a unique NE in the stage game, in the strategy profile (A,S) with static NE payoffs (8,15).

Next, we examine the Stackelberg equilibria when each of the players is the leader player of the game. Thus, we locate the Stackelberg strategy, and the corresponding payoff as a player's choice, in order to get the highest possible profit when the opponent will play his

best response. Table 5.4 displays for every strategy chosen first by player I (column 1), player D's best response (column 2) and the corresponding payoff for player I (column 3). The Stackelberg payoff is the most player I can get when player D plays a best response, that is, the maximum player I can get from Table 5.4, which is payoff 15, indicated with a red circle. Then, the Stackelberg strategy for player I is to play N.

Strategy of I	**Best response of D**	**Payoff to I**
N	**C**	15•
M	**S**	**2**
P	**S**	**0**
A	**S**	**8**

Table 5.4: Stackelberg equilibrium with player I leader

Likewise, Table 5.5 displays for every strategy chosen first by player D (column 1), player I's best response (column 2), and the corresponding payoff for player D (column 3). The Stackelberg payoff is the most player D can get when player I plays a best response, that is, the maximum player D can get from Table 5.5, which is payoff 15, indicated with a blue circle. Then, the Stackelberg strategy for player D is to play S. We summarize our findings on Stackelberg equilibria for both players in the following:

Stackelberg equilibrium with player *I* leader: The Stackelberg strategy for player I is to play N. The most that player I can get when player D plays a best response is the Stackelberg payoff, which is 15.

Strategy of D	Best response of I	Payoff to D
C	**A**	**0**
R	**A**	**10**
W	**A**	**11**
S	**A**	15•

Table 5.5: Stackelberg equilibrium with player D leader

Stackelberg equilibrium with player *D* leader: The Stackelberg strategy for player D is to play S. The most that player D can get when player I plays a best response is the Stackelberg payoff, which is 15.

Then, for each player, we find the least amount he receives when he plays a best response, which is called the minmax payoff. We construct Table 5.6 for player I, and we examine for each action of player D (column 1), player I's best response (column 2), and player I's payoff (column 3). The minimum of player I's payoffs listed in column 3 is the minmax payoff to player I, which is 8, indicated with a red circle.

Similarly, we construct Table 5.7 for player D, and we examine for each action of player I (column 1), player D's best response (column 2), and player D's payoff (column 3). The minimum of player D's payoffs listed in column 3 is the minmax payoff to player D, which is 12, indicated with a blue circle.

Summarizing the minmax payoffs for both players we have:

Minmax for player *I*: The least amount player I gets when he plays a best response is the minmax payoff 8.

Strategy of D	Best response of I	Payoff to I
C	**A**	**16**
R	**A**	**12**
W	**A**	**10**
S	**A**	8•

Table 5.6: Minmax for player I

Strategy of I	Best response of D	Payoff to D
N	**C**	**16**
M	**S**	12•
P	**S**	**14**
A	**S**	**15**

Table 5.7: Minmax for player D

Minmax for player *D*: The least amount player D gets when he plays a best response is the minmax payoff 12.

Then, we find the strategies that strictly Pareto dominate the static NE with payoffs (8,15). In strict Pareto dominance both players receive higher payoffs than the static NE payoffs (see Section 5.4 for details). In our game with an insider, only strategy *(N,C)* with payoffs $(15, 16)$ *strictly Pareto dominates* the static NE, and because there is no other

strategy with higher payoffs, i.e. there is only one such strategy, *(N,C)* is the *Pareto efficient strategy.*

The Folk Theorem: To carry this discussion further and understand more the repeated ID game with an insider, we need to talk about the Folk theorem. According to the Folk theorem, any payoff in a repeated game can be an equilibrium [115]. Given that the game is being played repeatedly and both players are sufficiently patient, then we can exclude those payoffs that are obviously uninteresting. The remaining payoffs are the feasible payoffs for each player (see Section 4.1.1), which offer more than his corresponding minmax payoff. The feasible payoffs for a player are the payoffs that Pareto dominate his minmax and they are the subgame perfect equilibrium payoffs.

For our game, we start plotting the set of payoffs from the normal form of the game (Table 5.3), which are illustrated in Figure 5.2 with blue diamonds. Then, we locate the minmax payoffs for both players, which is 8 for player I as derived from Table 5.6, and 12 for player D as derived from Table 5.7. The minmax (8,12) is depicted with a big red circle.

Finally, we bound the area with the folk theorem outcomes by drawing an horizontal line and a vertical line in parallel with axis x and axis y respectively, and isolate the shaded area above the minmax (8,12). The Folk theorem ensures that any payoff profile included in the shaded green area, can be achieved as a subgame perfect payoff profile, if players' discount factors are sufficiently close to 1. These are the (8,15) and the (15,16) payoff profiles, which are the static NE and the Pareto efficient respectively.

Grim Strategies: Another interesting point we examined is the grim strategies equilibria. Considering a case where the internal attacker plays a certain strategy at every period, we examined the circumstances under which he would deviate from this equilibrium path, and he would decide to play another strategy. But the IDS should react by choosing a 'punishment' strategy against such a deviation, known as grim strategy.

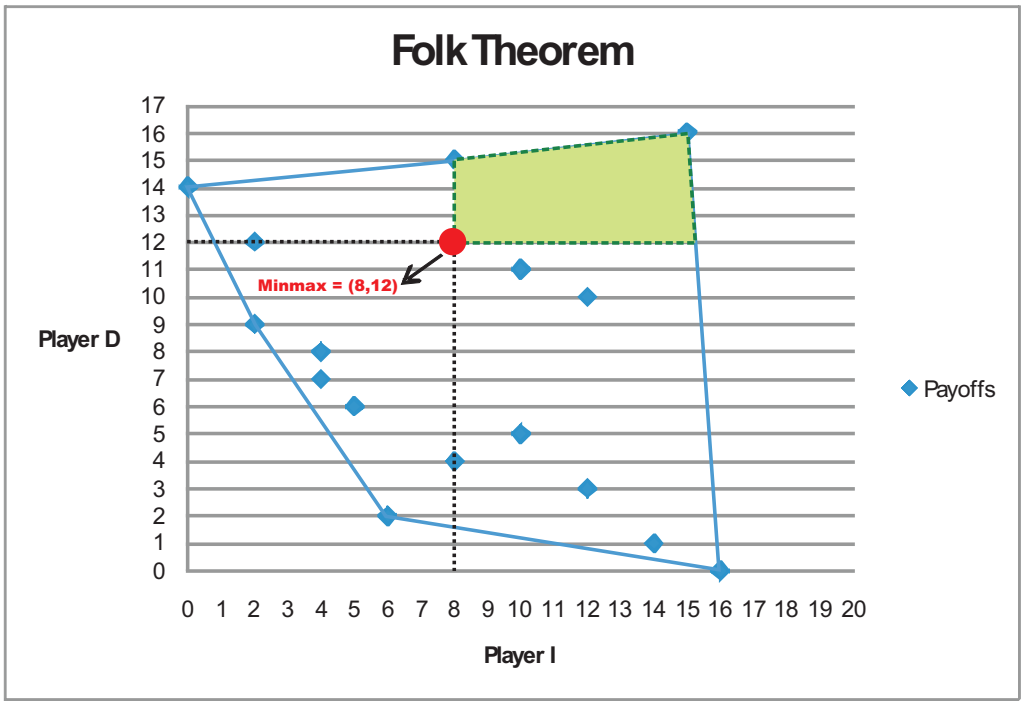

Figure 5.2: The Folk Theorem for the ID game with an insider

In our game, we have constructed a grim strategy equilibrium, which keeps players with the equilibrium path. It is the strategy profile *(*N,C) with payoffs (15,16) which strictly Pareto dominates the static NE *(A,S)* with payoffs (8,15). If player I deviates from this equilibrium path, then player D will punish him for this deviation with the static NE. After this, player I will get only 8 instead of 15 for ever then. We looked specifically at grim strategies in three different scenarios with both players patient (Case A), to define the discount factor δ above which an insider will deviate from the equilibrium path and attack the Target System, and one scenario with a patient and an impatient player (Case B), to examine how player D will react, as described in the sequel.

[Case A] Assuming that both players are patient, we use a common discount factor δ for both players, and the *average present value* to accumulate the payoffs among the time

periods of repetition. For such a case, we examine three different scenarios. We calculate the average present value (APV) for a player on the equilibrium path using Equation (4.9) of Section 4.1.1, which has the following simpler form

$$(1-\delta)\cdot\left(u_1+\delta u_2+\delta^2 u_3+\delta^3 u_4+\ldots\right). \tag{5.7}$$

where $u_i, i=1,2,3...$, is the payoff a player receives at period i and δ is the common discount factor. In addition, the identity below has been also used because it is related with Formula (6.2),

$$1+\delta+\delta^2+\delta^3+\ldots=\frac{1}{1-\delta} \tag{5.8}$$

[Scenario 1] When players play *(N,C)* for ever, that is, they stick with the Pereto efficient strategy, then the average present values for players I and D respectively are:

$APV_I=(1-\delta)\cdot(u_1+\delta\cdot u_2+\delta^2\cdot u_3+\delta^3\cdot u_4+\ldots)=$
$(1-\delta)\cdot(15+\delta\cdot 15+\delta^2\cdot 15+\delta^3\cdot 15+\ldots)=$
$(1-\delta)\cdot 15\cdot(1+\delta+\delta^2+\delta^3+\ldots)=$
$(1-\delta)\cdot 15\cdot\frac{1}{(1-\delta)}=15.$

$APV_D=(1-\delta)\cdot(u_1+\delta\cdot u_2+\delta^2\cdot u_3+\delta^3\cdot u_4+\ldots)=$
$(1-\delta)\cdot(16+\delta\cdot 16+\delta^2\cdot 16+\delta^3\cdot 16+\ldots)=$
$(1-\delta)\cdot 16\cdot(1+\delta+\delta^2+\delta^3+\ldots)=$
$(1-\delta)\cdot 16\cdot\frac{1}{(1-\delta)}=16.$

It is important to note that if the same strategy is being played at every period of the game, then the average present value is equal to the fixed payoff that corresponds to this strategy. The above calculations confirm this statement.

[Scenario 2] Suppose players play *(N,C)* in the first period and continue with the same strategies in the next periods, until one of the players plays something else. The strategy profile *(N,C)* is an equilibrium path of the game, with an average present value (15,16). But the insider, player I, will attempt a deviation from this equilibrium path, by choosing action A to attack the Target System, for his own purposes. In order to prevent such a game subversion, player D will punish him by playing *(A,S)*, which is the static NE, and will stick with it for ever. So, players play repeatedly *(N,C)* until period $t-1$, when player I decides to play A in period t, considering that he will increase his payoff from 15 to 16, if player D continues playing C, as listed in the following:

period 1: *(N,C)*

period 2: *(N,C)*

period 3: *(N,C)*

...

period $t-1$: *(N,C)*

period t: *(A,S)*

period $t+1$: *(A,S)*

...

for ever.

The strategy profile *(N,C)* will be the equilibrium path if the value of the common discount factor δ is so high to deter a deviation. We proceeded to determine δ, that is, to describe how patient player I must be to avoid an attack. The procedure is comprised of the following four steps.

Step 1 As calculated in the first scenario, on the equilibrium path players play *(N,C)* and get $(15, 16)$ each period. Then, the average present value on the equilibrium path is just $(15, 16)$.

Step 2 As presented before on Table 5.6, on the equilibrium path player D plays C, so the most that player I can get is 16, by playing A. Likewise, as shown on Table 5.7, on the equilibrium path player I plays N, so the most that player D can get is 16, by playing C. Therefore, we located that player I's best response is A and player D's best response is C, each time the other player chooses to continue on the equilibrium path.

Step 3 To calculate the average present value for player I for deviating from the equilibrium path, we consider that he receives 16 just for one period, the period t of deviation, and because player D punishes him for this, he gains 8 for ever then. Then, the average present values for players I and D respectively are:

$$APV_I = (1-\delta)\cdot(u_1 + \delta\cdot u_2 + \delta^2\cdot u_3 + \delta^3\cdot u_4 + \ldots) =$$

$$(1-\delta)\cdot(16 + \delta\cdot 8 + \delta^2\cdot 8 + \delta^3\cdot 8 + \delta^4\cdot 8 + \ldots) =$$

$$(1-\delta)\cdot(16 + \delta\cdot 8\cdot(1 + \delta + \delta^2 + \delta^3 + \ldots)) =$$

$$(1-\delta)\cdot 16 + (1-\delta)\cdot\delta\cdot 8\cdot\tfrac{1}{1-\delta} =$$

$$(1-\delta)\cdot 16 + \delta\cdot 8 =$$

$$16 - \delta\cdot 16 + \delta\cdot 8 =$$

$$16 - \delta\cdot 8.$$

$$APV_D = (1-\delta)\cdot(u_1 + \delta\cdot u_2 + \delta^2\cdot u_3 + \delta^3\cdot u_4 + \ldots) =$$

$$(1-\delta)\cdot(16 + \delta\cdot 15 + \delta^2\cdot 15 + \delta^3\cdot 15 + \delta^4\cdot 15 + \ldots) =$$

$$(1-\delta)\cdot(16 + \delta\cdot 15\cdot(1 + \delta + \delta^2 + \delta^3 + \ldots)) =$$

$$(1-\delta)\cdot 16 + (1-\delta)\cdot\delta\cdot 15\cdot\tfrac{1}{1-\delta} =$$

$$(1-\delta)\cdot 16 + \delta\cdot 15 =$$

$$16 - \delta\cdot 16 + \delta\cdot 15 =$$

$16 - \delta$.

Step 4 In the final step, we compare the average present value of sticking to the equilibrium path (calculated in step 1) with that of deviating (calculated in step 3). To determine the value of δ above which a deviation will make no sense, we construct and solve the following inequality:

$15 \geq 16 - \delta \cdot 8 \Rightarrow \delta \cdot 8 \geq 1 \Rightarrow \delta \geq \frac{1}{8}$.

The fraction $\frac{1}{8}$ is the critical discount factor for player I, that is to say, player I will deviate from the equilibrium path *(N,C)* if his payoffs are discounted with a factor less than $\frac{1}{8}$. Otherwise, player I will prefer to stick with this equilibrium path, because he will gain more than deviating.

As for player D, because in the equilibrium path the payoff is the maximum (see Table 5.7), there is no point for deviation. Besides, our main interest is how we will predict player I's future attacking actions and the circumstances under which this might happen.

Closing scenario 2, a patient player I will attempt to attack the Target System, if the common discount factor δ is less than $\frac{1}{8}$. Given that the maximum patience a player can have is 1 and the least is 0, it is remarkable that in our result player I is not that patient. It is his preferences (see Section 5.2.2) that have constructed such a profile of an insider. In the next section, we examine another insider, an unconventional internal attacker, and we calculate his δ, which reveals a more patient player.

[Scenario 3] We assume the same as in scenario 2, except that, players play the equilibrium path in the first and the second period only, and then, player I deviates by choosing A. To calculate the common discount factor δ for player I, steps 3 and 4 change, as described in the sequel:

Step 3 To calculate the average present value for player I for deviating from the equilibrium path, we consider that he receives 16 for two periods, and because player D punishes him for this, he gains 8 for ever then. Then, the average present value for player I is:

$$APV_I = (1-\delta)\cdot(u_1 + \delta\cdot u_2 + \delta^2\cdot u_3 + \delta^3\cdot u_4 + \ldots) =$$

$$(1-\delta)\cdot(16 + \delta\cdot 16 + \delta^2\cdot 8 + \delta^3\cdot 8 + \delta^4\cdot 8 + \ldots) =$$

$$(1-\delta)\cdot(16 + \delta\cdot 16 + \delta^2\cdot 8\cdot(1 + \delta + \delta^2 + \delta^3 + \ldots)) =$$

$$(1-\delta)\cdot 16 + (1-\delta)\cdot\delta\cdot 16 + (1-\delta)\cdot\delta^2\cdot 8\cdot\tfrac{1}{1-\delta} =$$

$$(1-\delta)\cdot 16 + (1-\delta)\cdot\delta\cdot 16 + \delta^2\cdot 8 =$$

$$16 - \delta\cdot 16 + \delta\cdot 16 - \delta^2\cdot 16 + \delta^2\cdot 8 =$$

$$16 - \delta^2\cdot 8.$$

Step 4 In the final step, we compare the average present value of sticking to the equilibrium path (calculated in step 1) with that of deviating (calculated in step 3). To determine the value of δ above which a deviation will make no sense, we construct and solve the following inequality:

$$15 \geq 16 - \delta^2\cdot 8 \Rightarrow \delta^2\cdot 8 \geq 1 \Rightarrow \delta^2\cdot 8 - 1 \geq 0 \Rightarrow 8\cdot(\delta + \tfrac{\sqrt{2}}{4})\cdot(\delta - \tfrac{\sqrt{2}}{4}) \geq 0.$$

Solving the inequality derived from factoring, we exclude the negative values of δ and we keep only that $\delta \geq \frac{\sqrt{2}}{4}$. The fraction $\frac{\sqrt{2}}{4}$ is the critical discount factor for player I, that is to say, player I will deviate from the equilibrium path *(N,C)* if his payoffs are discounted with a factor less than $\frac{\sqrt{2}}{4}$. Otherwise, player I will prefer to stick with this equilibrium path, because he will gain more than deviating.

Comparing the value of δ in the second scenario with the corresponding in the third scenario, we conclude that the results are reasonable. This is because playing the equilibrium path *(N,C)* twice (scenario 3) instead of once (scenario 2), we determined $\delta \geq \frac{\sqrt{2}}{4}$ which reveals a more patient player than the one that has $\delta \geq \frac{1}{8}$, given that $\frac{\sqrt{2}}{4} > \frac{1}{8}$.

[Case B] Assuming that player I is about to be fired, he becomes the short-run player of the game with a discount factor $\delta_1 = 0$. Player D is the long-run player of the game, with a discount factor δ_2, which is closer to 1. Therefore, in this case we use two different discount factors.

It is expected for player I to play a best response whenever player D plays. This is called *rational expectations*, because, the short-run player tries to get as much as possible before the game ends for him. Therefore, the best the long-run player D can get is Stackelberg payoff, as defined in Table 5.5.

In this case, we are more interested in what player D can get, or better, what is the least player D can lose. Since the repeated static NE is always a subgame perfect equilibrium, player D can earn 15 at each period it plays the static NE. In this scenario, δ_2 is very close to 1, and therefore player D can get an equivalent amount as the Stackelberg payoff, because the corresponding Stackelberg subgame perfect equilibrium matches with the static NE, which is the *(A,S)* strategy profile. This happens because player I's best response in all player D's strategies is A.

The results of this case show that player D will stick with choice S in order to stop any attacking attempts on behalf of player I, as expected before he leaves. In this way, player D will avoid a great loss and will protect the system from being compromised and crushed. In days of economic crisis, there are cases in which employers inform an employee that their cooperation discontinues exactly in the day of leaving, to evade bad consequences in their business.

5.6 Playing with an Unconventional Insider

To evaluate the effectiveness of the ID game model, we constructed another game also to be played between an insider and an IDS. In this second game, the insider has a different inexplicable preference list, and thus we call him *unconventional*. The game has two players,

the internal attacker, player $I^{'}$, and the IDS, player D. We assume that player I has four strategies, Normal, Mistake, Pre-Attack, Attack, and player D has another set of four strategies, Continue, Recommend, Warning, Stop, as it has been described in Section 5.2.

As for the preferences, player D has exactly the same as in the previous game constructed in Section 5.2. But, player $I^{'}$'s preferences are ranked over the game outcomes differently, as described in the structure below.

$$\begin{aligned} PS_{I'} \prec MS_{I'} \sim PW_{I'} \prec PR_{I'} \sim MW_{I'} \prec MR_{I'} \sim AR_{I'} \prec AW_{I'} \sim NS_{I'} \prec \\ AS_{I'} \sim NW_{I'} \prec NR_{I'} \prec MC_{I'} \prec PC_{I'} \prec NC_{I'} \prec AC_{I'}. \end{aligned} \tag{5.9}$$

Notice that preferences (A,W) and (A,R) have been repositioned in between (M,R) and (N,S) because this type of insider prefers less to get a warning or a recommendation when attacking than e.g. preparing an attack without being stopped. Moreover, player $I^{'}$ places in between (N,W) and (N,R) his preference on outcome (A,S), because he prefers to complete an attack and being stopped than receiving a warning for a normal action.

These preferences are described by the utility function $u_{I'} : A \rightarrow \mathbb{R}$, where A is the set of action profiles defined in Equation 5.2. We assigned numbers to reflect these preferences, following again Binmore's method, and we constructed player $I^{'}$'s utility function $u_{I'}$. The normal form of this game is presented in the 4x4 matrix of Table 5.8. The row player is player $I^{'}$ and the column player is player D.

Examining players' best responses, we conclude that the game has the same strategy profile *(A,S)* with payoffs (8,16) as the unique NE. Moreover, the (N,C) strategy profile, with the absolutely highest payoffs each player can get in this game, is again the Pareto efficient strategy, which dominates the NE, but with payoffs (13,17). Then, how different is this game with the unconventional player $I^{'}$ comparing to the game with player I? The answer is given in the next simple example.

I' \ D	**C**	**R**	**W**	**S**
N	(13,17•)	(9•,5)	(8•,4)	(7,2)
M	(10,3)	(6,6)	(5,7)	(4,14•)
P	(12,1)	(5,8)	(4,9)	(3,15•)
A	(19•,0)	(6,10)	(7,11)	(**8**•,**16**•)

Table 5.8: A game between an unconventional insider and an IDS in normal form

A Simple Example

The internal attacker starts playing the game acting legitimately, by choosing strategy N. The IDS reacts by playing strategy C. This goes on at every period as long as NC strategies are being played. But then, under which circumstances the internal attacker will commit an attack, i.e. deviate from this equilibrium path? We are looking for the discount factor δ that will motivate the internal attacker to do so because his benefit will be higher.

Solution: We calculate the average present value (APV) for each player on the equilibrium path using Equation (6.2) and the related identity (6.3). We found $APV_U = 13$ and $APV_I = 17$ respectively as expected, because the same strategies are being played at every period. Following this, we examined players' best responses when the opponent follows the selected equilibrium path. When the internal attacker follows the strategy N, then the IDS's best response is strategy C. But when the IDS follows the strategy C, then the internal attacker's best response is strategy A, because his highest payoff is 19. Next, we calculated the APV for each player if each follows the equilibrium path at the first period but then they both deviate and continue playing the static NE as calculated. The results are $APV_I = 17 - \delta$ and $APV_U = 19 - 11 \cdot \delta$.

Finally, we compared the average present value to remaining on the equilibrium path with that to deviating. In other words, we determined the discount factor δ for which a player will stick with his first choice and will not change afterwards by attacking the TS. The δ discount factor must be greater or equal to $\frac{6}{11}$, which is reasonable for this type of attacker. The fact is that, the internal attacker behaves as a patient player when he takes his time to complete an attack, but he is impatient enough when time pushes him to finish with his illegal activities. The result can be verified by calculating the limit of the derived average present values when δ is close to 1, that is to say, when players are patient.

$$\lim_{\delta \to 1} (17 - \delta) = 16 \tag{5.10}$$

$$\lim_{\delta \to 1} (19 - 11 \cdot \delta) = 8 \tag{5.11}$$

Apparently from (6.4) and (6.13), the derived formulas for the average present values for both players give the payoffs of the static NE, when δ is close to 1.

Comparing the discount factor of this game $\delta \geq \frac{6}{11}$ with $\delta \geq \frac{1}{8}$ (Scenario 2) or $\delta \geq \frac{\sqrt{2}}{4}$ (Scenario 3) calculated for the game solved in Section 5.5, we come to the conclusion that although both games have the same unique NE and the same Pareto efficient strategy profile, because in their preference lists the game outcomes have been ranked in such ways to reflect the distinct types of insiders, the solutions show that the type of attacker differs by how much patient each insider is. Besides, the unconventional insider has a patience at the midpoint ($\delta \geq \frac{6}{11} \approx \frac{1}{2}$), so that, one cannot clearly categorize him as a patient or an impatient player.

The game with the unconventional insider revealed a relation between a player's preferences and his patience. The slight changes in the ranking of the preference structure did not affect the NE or its uniqueness, nor the Pareto efficient strategy profile that dominated the NE. But the calculations of δ showed that these changes actually affected the degree

of patience for players I and $I^{'}$ respectively, and thus how two different types of insiders formulated two different games.

5.7 Summary

Insiders might threaten organizations' systems any time. By interacting with a system, an insider plays games with the security mechanisms employed to protect it. Therefore, we decided to give special attention to insiders when they interact with an Intrusion Detection System. In order to ascertain the insider threat, we elucidated who is an insider and the different ways he might act. In addition, we ranked the corresponding risk of an insider's actions, and we briefly reviewed existing solutions that address the problem of insiders.

Subsequently, we constructed a game with two players, an insider and an IDS, following the interpretation of the generic game model presented in Chapter 3, and its repeated formal description given in Chapter 4. The use of preference lists for both players and the construction of their utility functions, quantified players' preferences in a realistic way.

Solving the repeated game, we determined the discount factor that formulates this game differently at each turn of the play, at the end of a time period $t_x,\ x = 0, 1, 2, \ldots$. We concentrated our work in determining the critical discount factor above which an insider has no incentive to attack the TS. To study further these cases, we constructed another game with an unconventional insider, the solution of which, comparing to the first game, resulted in some valuable conclusions.

Unfortunately, determining the discount factor of a repeated game does not solve other complexity problems the increased number of repetitions generates, as the multiple NE. Although NE has been largely accepted as a solution concept in Game Theory, other solution concepts that generalize the NE have been proposed (see Section 1.2), to cover weaknesses and overcome limitations. The Quantal Response Equilibrium (QRE) discussed in Chapter 7 is such an alternative.

Chapter 6

Uncertainty in ID Signaling Games

I have always thought the actions of men the best interpreters of their thoughts.

John Locke (1632 - 1704)

Doubt is not a pleasant condition, but certainty is absurd.

Voltaire (1694 - 1778)

Not to be absolutely certain is, I think, one of the essential things in rationality.

Bertrand Russell (1872 - 1970),
"Am I An Atheist Or An Agnostic?", 1947

The current chapter examines the ID game model established in Chapters 3 and 4 as a signaling game. In signaling games players have no complete information and therefore they exchange signals to play the game. A signal reflects private information a player holds and its recipient encodes it in order to take an action. First, we construct the ID signaling

game in an extensive form by defining the corresponding payoffs. Next, we represent it in a payoff matrix as a normal form game. We examine then the solution of the game by removing the dominated strategies. Finally, we compute all the equilibria of the ID signaling game in pure and behavioral strategies.

Attempting to model ID as a signaling game, we assume a User who has already gained access to a Target System (TS) and starts using it, regardless whether he is a legitimate user, a masquerade, a hacker, or a cracker. The method used to get into the system, in cases where the User is not authorized, are not considered here because it is covered by the area of Access Control in IT Security. In our case, we care about users of the system we do not know whether they are going to behave legitimately or illegally, accidentally or intentionally.

6.1 Constructing the ID Signaling Game

The *User* will move in two ways, acting *Legitimately* (L) or acting *Illegally* (I). The Target System is equipped with an Intrusion Detection System (IDS) ready to play with this User. The *IDS* will decide to *Prevent* (P) the User from further using the TS or to allow the User to *Continue* (C). To make this decision, the IDS should conclude if the User is an enemy of the TS, i.e. an Attacker no matter whether he is an insider or an outsider.

The IDS does not know for sure if the *User* is a *Normal User* or an *Attacker*. This means that there is a simple probability distribution. Assuming that the number of reported attacks is for example the 25% of the occurring events in a Target System, then the *IDS* knows with probability $\frac{1}{4}$ that the *User* is an *Attacker* and with probability $\frac{3}{4}$ that the *User* is a *Normal User*. Later on, this number will be refined to reflect the actual number of attacks that take place in this specific Target System. This means that the proposed system will be self tuning and adjustable to current data related to the Target System itself.

Examining the set of alternative circumstances, the *IDS* will prevent the *User* if he is

an *Attacker* and the *Attacker* will run off because he was caught by the *IDS*. But if the IDS prevent a *Normal User* from using the Target System, then this *Normal User* might request justice, because a false positive alarm has been raised against him unfairly. The *IDS* receives signals from the User, and the decision whether he is a *Normal User* or an *Attacker* derives from the examination of these signals.

6.1.1 Defining the Payoffs

The utility functions U_N and U_A defined in Chapter 4 determine the corresponding payoffs at each node of the extensive form game and at each cell of the normal form game respectively. As for the utility function U_{IDS}, we adjust it to reflect all the preferences, when the opponent is either a *Normal User* or an *Attacker* in a signaling game, as described in the sequel.

IDS's Preferences

The set of an IDS's preferences is denoted by $\mathcal{IDS}$ and includes eight items in a signaling game, as described in the sequel:

$$\mathcal{IDS} = \{\mathcal{IDS}_1, \mathcal{IDS}_2, \mathcal{IDS}_3, \mathcal{IDS}_4, \mathcal{IDS}_5, \mathcal{IDS}_6, \mathcal{IDS}_7, \mathcal{IDS}_8\},$$

where,

$\mathcal{IDS}_1$: The IDS allows a Normal User who is acting legitimately to continue.

$\mathcal{IDS}_2$: The IDS prevents a Normal User who is acting legitimately to continue.

$\mathcal{IDS}_3$: The IDS allows a Normal User who is acting illegally to continue.

$\mathcal{IDS}_4$: The IDS prevents a Normal User who is acting illegally to continue.

$\mathcal{IDS}_5$: The IDS allows an Attacker who is acting legitimately to continue.

$\mathcal{IDS}_6$: The IDS prevents an Attacker who is acting legitimately to continue.

$\mathcal{IDS}_7$: The IDS allows an Attacker who is acting illegally to continue.

$\mathcal{IDS}_8$: The IDS prevents an Attacker who is acting illegally to continue.

Ranking these preferences from the most disliked to the most preferred one, we get:

$$\mathcal{IDS}_7 \prec \mathcal{IDS}_5 \prec \mathcal{IDS}_2 \prec \mathcal{IDS}_3 \prec \mathcal{IDS}_1 \prec \mathcal{IDS}_4 \prec \mathcal{IDS}_6 \prec \mathcal{IDS}_8 \tag{6.1}$$

Next, we will define another utility function for the player IDS. Suppose that $U_{\mathcal{IDS}} : \{\mathcal{IDS}_1, \mathcal{IDS}_2, \mathcal{IDS}_3, \mathcal{IDS}_4, \mathcal{IDS}_5, \mathcal{IDS}_6, \mathcal{IDS}_7, \mathcal{IDS}_8\} \rightarrow \mathbb{R}$ is the utility function for the IDS. The IDS has an aversion to preference $\mathcal{IDS}_7$, because this is the worst case scenario for it that raises a false negative alarm. For this reason, we define $U_{\mathcal{IDS}}(\mathcal{IDS}_7) = 0$. Furthermore, because it mostly prefers $\mathcal{IDS}_8$, we define $U_{\mathcal{IDS}}(\mathcal{IDS}_8) = 1$. Selecting between $\mathcal{IDS}_3$ and $\mathcal{IDS}_1$, which are intermediate preferences, we decide to define $U_{\mathcal{IDS}}(\mathcal{IDS}_3) = \frac{1}{2}$. Next, because $\mathcal{IDS}_4$ is the intermediate between $\mathcal{IDS}_3$ and $\mathcal{IDS}_8$, we define $U_{\mathcal{IDS}}(\mathcal{IDS}_4) = \frac{3}{4}$, by calculating the value of $U_{\mathcal{IDS}}(\mathcal{IDS}_4)$ which is the middle between $\mathcal{IDS}_3$ and $\mathcal{IDS}_8$, that is:

$$U_{\mathcal{IDS}}(\mathcal{IDS}_4) = U_{\mathcal{IDS}}(\mathcal{IDS}_3) + \frac{U_{\mathcal{IDS}}(\mathcal{IDS}_8) - U_{\mathcal{IDS}}(\mathcal{IDS}_3)}{2} = \frac{1}{2} + \frac{1-\frac{1}{2}}{2} = \frac{1}{2} + \frac{\frac{1}{2}}{2} = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}.$$

Calculating the utilities for $\mathcal{IDS}_1$ and $\mathcal{IDS}_6$ respectively, we get:

$$U_{\mathcal{IDS}}(\mathcal{IDS}_1) = U_{\mathcal{IDS}}(\mathcal{IDS}_3) + \frac{U_{\mathcal{IDS}}(\mathcal{IDS}_4) - U_{\mathcal{IDS}}(\mathcal{IDS}_3)}{2} = \frac{1}{2} + \frac{\frac{3}{4}-\frac{1}{2}}{2} = \frac{1}{2} + \frac{\frac{1}{4}}{2} = \frac{1}{2} + \frac{1}{8} = \frac{5}{8}.$$

$$U_{\mathcal{IDS}}(\mathcal{IDS}_6) = U_{\mathcal{IDS}}(\mathcal{IDS}_4) + \frac{U_{\mathcal{IDS}}(\mathcal{IDS}_8) - U_{\mathcal{IDS}}(\mathcal{IDS}_4)}{2} = \frac{3}{4} + \frac{1-\frac{3}{4}}{2} = \frac{3}{4} + \frac{\frac{1}{4}}{2} = \frac{3}{4} + \frac{1}{8} = \frac{7}{8}.$$

Finally, we calculate in a similar way the utilities for $\mathcal{IDS}_5$ and $\mathcal{IDS}_2$ as described in the sequence:

$U_{\mathcal{IDS}}(\mathcal{IDS}_5) = U_{\mathcal{IDS}}(\mathcal{IDS}_7) + \frac{U_{\mathcal{IDS}}(\mathcal{IDS}_3) - U_{\mathcal{IDS}}(\mathcal{IDS}_7)}{3} = 0 + \frac{\frac{1}{2}-0}{3} = \frac{\frac{1}{2}}{3} = \frac{1}{6}$.

$U_{\mathcal{IDS}}(\mathcal{IDS}_2) = U_{\mathcal{IDS}}(\mathcal{IDS}_5) + \frac{U_{\mathcal{IDS}}(\mathcal{IDS}_3) - U_{\mathcal{IDS}}(\mathcal{IDS}_5)}{2} = \frac{1}{6} + \frac{\frac{1}{2}-\frac{1}{6}}{2} = \frac{1}{6} + \frac{\frac{2}{6}}{2} = \frac{1}{6} + \frac{1}{6} = \frac{2}{6} = \frac{1}{3}$.

In Table 6.1, we summarize the utilities defined for the IDS, in the second row. The third row contains the corresponding utilities transformed into integer numbers instead of fractions.

x	$\mathcal{IDS}_7$	$\mathcal{IDS}_5$	$\mathcal{IDS}_2$	$\mathcal{IDS}_3$	$\mathcal{IDS}_1$	$\mathcal{IDS}_4$	$\mathcal{IDS}_6$	$\mathcal{IDS}_8$
$U_{\mathcal{IDS}}(x)$	0	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{5}{8}$	$\frac{3}{4}$	$\frac{7}{8}$	1
$24 \times U_{\mathcal{IDS}}(x)$	0	4	8	12	15	18	21	24

Table 6.1: IDS's Utility Function in a Signaling Game

Concerning the payoffs of this game on behalf of the *User*, if the *IDS* permits a *Normal User* to *Continue*, then the *Normal User* gains 4 points when he is acting *Legitimately*, and 1 point less, i.e. he gets 3 points when he is acting *Illegally*. Moreover, if the *IDS* permits an *Attacker* to *Continue*, then the *Attacker* gains 2 points if he acts *Legitimately* so he doesn't achieve his goals, and 4 points if he acts *Illegally* so he achieves his goals.

Similarly, if the *IDS Prevents* a *Normal User* to use the Target System, then the *Normal User* gets no points (0 points) if he acts *Legitimately* and he gets 2 points if he acts *Illegally*, because he does not act by purpose. Likewise, if the *IDS Prevents* an *Attacker* to use the Target System, then the *Attacker* gets nothing if he acts *Legitimately* and 3 points if he acts *Illegally*.

As the *User*'s payoffs start from 0 and goes to 4 (see Chapter 4), the *IDS*'s payoffs vary between 0 and 24. There is a difference between the two payoffs' scales, because a *Normal User* has 4 payoffs, an *Attacker* another 4, whereas the *IDS* has 8 payoffs, since he plays the game with both, either the *Normal User* or the *Attacker*.

Specifically, the *IDS* gains 15 points if it permits a *Normal User* who acts *Legitimately* to *Continue*, and 12 points if it permits a *Normal User* to *Continue* although he acts *Illegally*. In the case it permits an *Attacker* to *Continue* because he acts *Legitimately*, the *IDS* gains only 4 points because this is a false negative alarm. If it *Prevents* a *Normal User* to *Continue* although he acts *Legitimately*, the *IDS* gets 8.

In addition, the *IDS* loses by getting no points at all, when it permits an *Attacker* with *Illegal* actions to *Continue*. On the contrary, the *IDS* gains 18 points if it *Prevents* a *Normal User* from acting *Illegally*, 21 points if it *Prevents* an *Attacker* from acting *Legitimately*, and finally, 24 points if it *Prevents* an *Attacker* from acting *Illegally*.

Apparently, the ID game as a signaling game is not a zero-sum game, neither a constant-sum game. An *Attacker* is pretty happy if he commits an attack without being caught by the *IDS* (4 points), but he is a loser if the *IDS* detects his intentions correctly and stops him before he achieves his goals (0 points).

In the same way, a *Normal User* is satisfied by using the Target System in a *Legitimate* manner and nobody disturbs or stops him. But when the *IDS Prevents* him unfairly from doing so, he is one hundred per cent a loser of the game.

Conversely, the *IDS* maximum payoff is when it detects accurately an *Attacker* who acts *Illegally* and stops him (24 points), that is, when the *User* too gets some payoff (3 points) because he has already acted *Illegally*. Finally, the *IDS* gets no payoff (0 points) when it leaves undetected an *Attacker* who acts *Illegally* and permits him to *Continue* using the TS.

Figure 6.1 shows the extensive form of the ID game as a signaling game, drawn by the

GAMBIT tool.

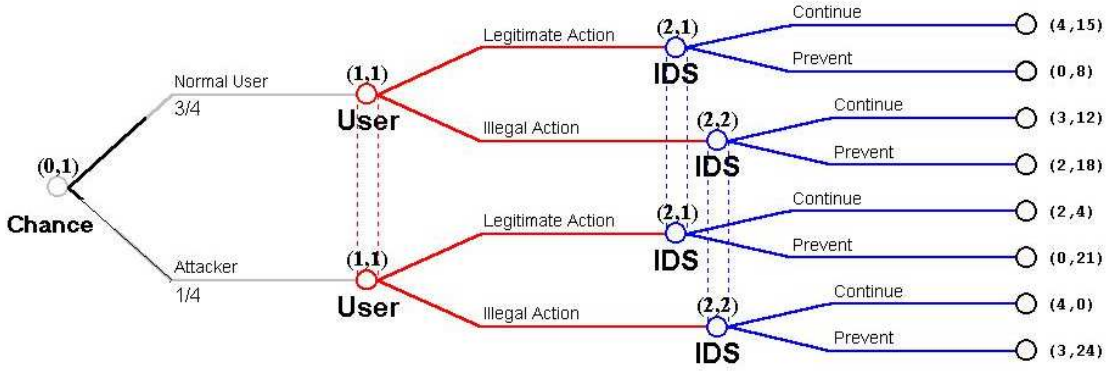

Figure 6.1: Intrusion Detection as a signaling game

Since the ID game starts with a *Chance* node, where there is a probability p with which a *User* is a *Normal User*, and a probability $1 - p$ with which the *User* is an *Attacker*, the game is an incomplete information game, because the *IDS* does not know for sure what is the type of *User*. This is the private information the *User* holds.

Incomplete information games were formulated and studied by John Harshanyi in 1967[1]. They are amongst the most challenging games to be solved. They model strategic problems in which the players have no complete information about each other's preferences. They also have the potential to model irrationality as was shown in the famous "gang of four" paper in 1982 (Kreps, Milgrom, Roberts, and Wilson).

6.2 Constructing the Normal Form of the ID Signaling Game

In the ID game there are two players, the *IDS* which protects the Target System and a *User* who uses the Target System. In reality, there are a number of users who act on the

[1]Harshanyi shares the 1994 Nobel Prize in Economics, together with John Nash and Reihard Selten, mainly because of this formulation.

Target System, but this is a more complicated setting, as it has already been considered in Chapter 4. We assume that player *User* has a binary choice between two actions: he can either act Legitimately (L) or he can act Illegally (I). These actions are the signals sent to the *IDS* from the *User*. The *IDS* has also two actions, to Prevent (P) the *User* to continue using the Target System or to permit him to Continue (C), because it decided that the signals come from an *Attacker* or from a *Normal User* respectively.

The possible actions described above lead to the sets of strategies that correspond to each player. Two capital letters are assigned to each strategy, the first corresponds to an action when the *User* is a *Normal User*, and the second corresponds to an action when the *User* is an *Attacker*. So, the *User* has four strategies. First, he can act Legitimately regardless he is a *Normal User* or he is an *Attacker* (LL). He can act Legitimately if he is a *Normal User* and Illegally if he is an *Attacker* (LI). He can act Illegally if he is a *Normal User* (accidentally) and Legitimately if he is an *Attacker* (bluffing) (IL). Ultimately, he can act Illegally no matter what he is (II).

The *IDS* has four strategies too. It can Prevent the *User* whatever he is (including false positives) (PP). It can Prevent the *User* if he is an *Attacker* and allow him to Continue if he is a *Normal User* (CP). It can allow the *User* to Continue if he is an *Attacker* (false negatives) and Prevent the *User* if he is a *Normal User* (false positives) (PC). Finally, it can allow the *User* to Continue regardless he is a *Normal User* or an *Attacker* (including false negative) (CC). It is remarkable that all these strategies encompass false alarms except the second one which is the optimal case, to allow a *Normal User* to Continue and to Prevent an *Attacker*. Besides, strategy PC seems irrational, but in fact, in this case the *IDS* does not trust the signals it gets from the *User*.

The payoffs assigned to each strategy are summarized in Table 6.3 below, where both cases of a *Normal User* or an *Attacker* are included. Rows correspond to *User*'s strategies and columns to the *IDS*'s strategies.

	PP	CP	PC	CC
LL	(0,8)/(0,21)	(4,15)/(0,21)	(0,8)/(2,4)	(4,15)/(2,4)
LI	(0,8)/(3,24)	(4,15)/(3,24)	(0,8)/(4,0)	(4,15)/(4,0)
IL	(2,18)/(0,21)	(3,12)/(0,21)	(2,18)/(2,4)	(3,12)/(2,4)
II	(2,18)/(3,24)	(3,12)/(3,24)	(2,18)/(4,0)	(3,12)/(4,0)

Table 6.2: IDS's Utility Function in a Signaling Game

Each cell includes a couple of payoffs pairs. The first pair corresponds to the case of a *Normal User* and the second pair corresponds to the case of an *Attacker*. The first number in each pair is *User*'s payoff and the second is *IDS*'s payoff. In Figure 6.2 we zoom at the matrix for details.

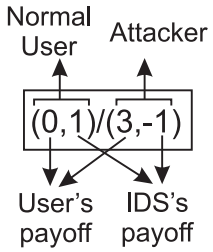

Figure 6.2: Details of the notation used in a cell of the payoffs matrix

Because the *IDS* knows with probability $\frac{1}{4}$ that the *User* is an *Attacker* and with probability $\frac{3}{4}$ that he is a *Normal User*, the expected return to the players should be calculated by adding the first half of the matrix multiplied by $\frac{3}{4}$ and the second half of the matrix multiplied by $\frac{1}{4}$. The calculations are given in the sequence:

$$\frac{3}{4}\cdot\begin{pmatrix} (0,8) & (4,15) & (0,8) & (4,15) \\ (0,8) & (4,15) & (0,8) & (4,15) \\ (2,18) & (3,12) & (2,18) & (3,12) \\ (2,18) & (3,12) & (2,18) & (3,12) \end{pmatrix}+$$

$$\frac{1}{4}\cdot\begin{pmatrix} (0,21) & (0,21) & (2,4) & (2,4) \\ (3,24) & (3,24) & (4,0) & (4,0) \\ (0,21) & (0,21) & (2,4) & (2,4) \\ (3,24) & (3,24) & (4,0) & (4,0) \end{pmatrix}=$$

$$\begin{pmatrix} (0,6) & (3,11\frac{1}{4}) & (0,6) & (3,11\frac{1}{4}) \\ (0,6) & (3,11\frac{1}{4}) & (0,6) & (3,11\frac{1}{4}) \\ (1\frac{1}{2},13\frac{1}{2}) & (2\frac{1}{4},9) & (1\frac{1}{2},13\frac{1}{2}) & (2\frac{1}{4},9) \\ (1\frac{1}{2},13\frac{1}{2}) & (2\frac{1}{4},9) & (1\frac{1}{2},13\frac{1}{2}) & (2\frac{1}{4},9) \end{pmatrix}+$$

$$\begin{pmatrix} (0,5\frac{1}{4}) & (0,5\frac{1}{4}) & (\frac{1}{2},1) & (\frac{1}{2},1) \\ (\frac{3}{4},6) & (\frac{3}{4},6) & (1,0) & (1,0) \\ (0,5\frac{1}{4}) & (0,5\frac{1}{4}) & (\frac{1}{2},1) & (\frac{1}{2},1) \\ (\frac{3}{4},6) & (\frac{3}{4},6) & (1,0) & (1,0) \end{pmatrix}=$$

$$\begin{pmatrix} (0,11\frac{1}{4}) & (3,16\frac{1}{2}) & (\frac{1}{2},7) & (3\frac{1}{2},12\frac{1}{4}) \\ (\frac{3}{4},12) & (3\frac{3}{4},17\frac{1}{4}) & (1,6) & (4,11\frac{1}{4}) \\ (1\frac{1}{2},18\frac{3}{4}) & (2\frac{1}{4},14\frac{1}{4}) & (2,14\frac{1}{2}) & (2\frac{3}{4},10) \\ (2\frac{1}{4},19\frac{1}{2}) & (3,15) & (2\frac{1}{2},13\frac{1}{2}) & (3\frac{1}{4},9) \end{pmatrix}$$

To avoid having payoffs in a fraction format, we multiply the above matrix by 4, and we get the following final payoff matrix:

	PP	CP	PC	CC
LL	0,45	12,66	2,28	14,49
LI	3,48	15,69	4,24	16,45
IL	6,75	9,57	8,58	11,40
II	9,78	12,60	10,54	13,36

Table 6.3: Payoff Matrix for the ID Signaling Game

6.3 Removing Dominated Strategies

We first solve the ID signaling game by applying the domination criterion, which says that a rational player should not use a dominated strategy. Binmore [22] expresses the domination criterion by assuming two strategies s_1 and s_2 of a player I and three strategies t_1, t_2, and t_3 of a player II. Then we decide that for player I, strategy s_2 strongly dominates strategy s_1 when

$$\pi_1(s_2, t) > \pi_1(s_1, t) \tag{6.2}$$

for all three values of player II's strategy t. Moreover, if the relation between two strategies is $\geq$, then the one strategy weakly dominates the other. In our game we express in algebraic terms the above criterion to check if it holds. First, we consider that IL is dominated by II because:

```
[6,9,8,11] > [9,12,10,13]
```

Using this domination argument, we remove strategy IL from the payoff matrix and the matrix changes to the following:

Second, PC is dominated by PP because:

```
[45,48,78] > [28,24,54]
```

	PP	CP	PC	CC
LL	0,45	12,66	2,28	14,49
LI	3,48	15,69	4,24	16,45
II	9,78	12,60	10,54	13,36

and thus we reduce the payoff matrix again by removing strategy *PC*. The payoff matrix now has the following form:

	PP	CP	CC
LL	0,45	12,66	14,49
LI	3,48	15,69	16,45
II	9,78	12,60	13,36

Third, *LL* is dominated by *LI* which can be expressed in algebraic terms as

```
[3,15,16] > [0,12,14]
```

So, the strategy *LL* is also out of the matrix and the payoff matrix turns into the following:

	PP	CP	CC
LI	3,48	15,69	16,45
II	9,78	12,60	13,36

Finally, strategy *CC* is dominated by strategy *CP* as

```
[69,60] > [45,36]
```

Consequently, the above deletions lead to a smaller $2x2$ matrix as shown in Table 6.4.

	PP	CP
LI	3,48	15,69
II	9,78	12,60

Table 6.4: Reduced Payoff Matrix for the ID Signaling Game

Studying the resulting matrix, it makes sense that a *Normal User* can either act Legitimately or Illegally, while an *Attacker* acts only Illegally. In addition, we should check if there is a mixed strategy equilibrium. Consider that the probability for player *User* of playing strategy *LI* is p, and the probability of playing strategy *II* is $1-p$. Then, because the *IDS*'s payoffs in strategies *PP* and *CP* are 48 points in the *LI* strategy, 78 points in the *II* strategy, and 69 points in the *LI* strategy and 60 points in the *II* strategy respectively, we get the following equation:

$$48p + 78(1-p) = 69p + 60(1-p) \tag{6.3}$$

Solving Equation 6.3 to determine p, we get $p = \frac{18}{39}$ which is very close to 0.5 ($\simeq$ 0.461538). The inference is that there is no Nash equilibrium for which the player *User* will decide to play the *LI* strategy. It sounds reasonable that an *Attacker* will think about acting Illegally all the time and that a *Normal User* makes mistakes[2].

Similarly, consider that p' is the probability the player *IDS* will choose strategy *PP*, and $1-p'$ is the probability player *IDS* will choose strategy *CP*. Then the following equation must hold:

[2]A *Normal User* who is acting legitimately for a while, but at the next point of time accidentally acts illegally, could be modeled with Selten's "Trembling-Hand Equilibrium".

$$3p' + 15(1 - p') = 9p' + 12(1 - p') \tag{6.4}$$

Solving Equation 6.4 to determine p', we get $p' = \frac{1}{3}$. Therefore, player *IDS* will Prevent the *User* to Continue with probability $\frac{1}{3}$ regardless he is a *Normal User* or an *Attacker*. Furthermore, the *IDS* will let a *Normal User* to Continue and Prevent an *Attacker* with probability $\frac{2}{3}$ which is the most rational strategy.

It seems that the *User* is indifferent between the strategy *LI* and the strategy *II*. Therefore, the results lead our reasoning to different approaches, as those described in the following section.

6.4 Computing Equilibria in the ID Signaling Game

In a signaling game with two players, the one player knows something the other doesn't, that is, the one player holds information the other doesn't, but he sends signals to give hints of this private information to the second player. When the other player picks up the signal, then he decides upon this what action to take as a response. The corresponding (assigned) payoffs show the winner and the loser of the game. A signaling game usually admits more than one Nash equilibria.

The Intrusion Detection game described previously is an example of a signaling game, because the player *User* knows if he is a *Normal User* or an *Attacker* whereas the Intrusion Detection System doesn't. In addition, the *User* sends signals to the *IDS* by using the Target System, the *IDS* collects the events generated by this activity and decides whether to prevent or to allow the *User* to continue using the TS, by judging if this activity belongs to a *Normal User* or to an *Attacker*. In the Intrusion Detection game, *Chance* should start the game by deciding if the player *User* is a *Normal User* or an *Attacker*. Then, the player *IDS* is at the opposite side of the *User* and it might either Prevent the *User* or it might allow the *User* to Continue using the TS. The *IDS* would Prevent the *User* if it were aware

that the *User* would damage the system, i.e. he is an *Attacker*, and it would allow the *User* to Continue if it were aware that no damage would be caused, i.e. he is a *Normal User*. Unfortunately, only the *User* knows for sure that he is a *Normal User* or an *Attacker*. In other words, only the *User* knows his type.

However, by using the TS, the *User* is sending a signal of Legitimate activity when he is a *Normal User*, and a signal of Illegal activity when he is an *Attacker*. To determine the type of signal that corresponds to each activity, i.e. to decode a signal, we assume that the event reception module hosted by the *IDS*, collects the lowest level functions of the operating system (e.g. system calls) and examines their return values. If the return value of a function indicates that the User has attempted a system violation, or a security relevant event has successfully taken place, then an illegal activity is assigned with this. Otherwise, a legitimate action has taken place.

On average, a *Normal User* will act Legitimately and an *Attacker* Illegally. Nevertheless, a *Normal User* might act accidentally Illegally, because for instance he is a novice and as such he makes mistakes (Selten's trembling hand perfect equilibrium). Likewise, an *Attacker* might act Legitimately as an attempt to bluff so he can avoid detection, or because he is an insider as examined in Chapter 5, so he is authorized for a number of activities, but he takes advantage of them to cause damage. In conclusion, the *User* who sends a signal might confuse the *IDS* on purpose or unintentionally.

In the next subsections, we follow Binmore's reasoning for the quiche game [21] and Gintis's concepts [60], to solve the ID game by locating any Nash equilibria, first in pure strategies and afterwards in behavioral strategies.

6.4.1 Locating Nash Equilibria in Pure Strategies

First, we examine the ID signaling game for Nash equilibria in pure strategies. Assume that player *IDS* chooses the strategy *PP*. Then, the best response for player *User* is strategy *IL*,

because it is reasonable to play with Legitimate actions if he is a *Normal User* and to act Illegally if he is an *Attacker*, not bluffing since he will be caught anyway. Besides, the payoff matrix shows that he is loosing less by playing *IL* than in any other choice. Considering the other way, if player *User* chooses *IL*, then the best response for player *IDS* is again *PP*. Therefore, the pair of strategies *IL* and *PP* is a Nash equilibrium.

Next, assuming that player *IDS* plays strategy *PC*, then player *User* plays *II* as his best response. But reversing the argument, shows that, if player *User* plays *II*, then player *IDS* plays *PP* and not *PC*, because his payoff is maximized with *PP* (6 points instead of 1). So, there are no Nash equilibria in which player *IDS* chooses strategy *PC*.

Similarly, if player *User* uses strategy *LL*, then player *IDS*'s best reply is strategy *CP*, whereas, if player *IDS* plays strategy *CP*, then player *User* will choose either strategy *LL* or strategy *IL* as best reply, because their payoffs are equal (5 points). Namely, it is undetermined what player *User* will do; he will act Illegally or Legitimately in the case he is a *Normal User*? Still, player *IDS* should counteract if player *User* acts Illegally and Prevent him from damaging the TS. There is a point here that requires further consideration.

In certain environments, we care not only about information related to the knowledge of the players, but also about information related to their beliefs. In our case, we examine player *IDS*'s beliefs after receiving a signal from player *User*, i.e. collects an event from the TS generated by the *User*. It is coherent for player *IDS* to allow the *User* to Continue using the TS, if it gets a signal of *Normal User*, and to Prevent him if it gets a signal of *Attacker* from him. So, the fact that player *IDS* chooses *CP* adds no more information. If the initial probability that player *User* is a *Normal User* is p, and that he is an *Attacker* is $1 - p$, this remains unchanged. However, the payoffs lead player *User* to act Legitimately when the probability of being a *Normal User* is higher. For that reason, it is XL (i.e. *IL* or *LL*) the best reply to strategy *CP*. As a result, there are no Nash equilibria in which player *IDS* chooses strategy *CP*.

Following the same reasoning, consider that player *IDS* uses strategy *CC*. Consequently, player *User* will use one of the strategies *LI* or *II*, that is, it is undetermined if a *Normal User* will act Illegally or Legitimately, whereas an *Attacker* will definitely act Illegally because he will evade detection. Reversing the case, if player *User* chooses *LI*, then the best response for player *IDS* is not strategy *CC* but strategy *CP* (3 points instead of -2). The conclusion is that there are no Nash equilibria when player *IDS* chooses strategy *CC*.

To end with the pure strategies, there is only one Nash equilibrium in pure strategy *IL* for player *User* and strategy *PP* for player *IDS*. Verifying this finding, the *IDS* Prevents a *Normal User* when he is acting Illegally either by purpose or unintentionally, but it prefers also to Prevent an *Attacker* to continue using the TS when he is acting Legitimately, because this legal activity might form the first steps of a complete attack scenario (see Section 1.1.3 for details).

Although a Nash equilibrium has been located in pure strategies, it is necessary to look for other Nash equilibria in mixed strategies. Such a task is quite difficult and complicated, but Nash has proved that *every finite game has at least one Nash equilibria if mixed strategies are allowed* [21]. In any case, we will achieve valuable results upon completion.

In order to facilitate this work, one can replace mixed strategies by behavioral strategies, as mentioned in Section 4.1.2. In the next section, there is an explanation why this can be done, and a description of the behavioral strategies search for Nash equilibria.

6.4.2 Locating Nash Equilibria in Behavioral Strategies

Perfect recall games are those where no player ever forgets any piece of information that was once in his knowledge. Thus, the ID game is a perfect recall game. In addition, Kuhn has proved the following theorem for perfect recall games [21]:

> *Kuhn's theorem*
> Whatever mixed or behavioral strategy s that player i may choose in a game of perfect recall, he or she has a strategy t of the other type with the property

that, however the opponents play, the resulting lottery over the outcomes of the game is the same for both s and t.

With this theorem, Kuhn has proved that in perfect recall games, mixed strategies and behavioral strategies are identical. Therefore, instead of searching for Nash equilibria in mixed strategies, we will examine the behavioral strategies of the ID game.

A behavioral strategy is a decentralized mixed strategy, that is, like a pure strategy, it is clear for a player what action to take at each information set, but, a probability is assigned to each action [21]. Based on this probability, a player decides how to proceed the game.

In our game, for player *User*, a behavioral strategy must assign a probability p with which the *User* will act Legitimately at the information set *Normal User*, and a probability q with which player *User* will act Legitimately at the information set *Attacker*. Correspondingly, the probability with which player *User* will act Illegally at the information set *Normal User* is $1 - p$, and the probability with which player *User* will act Illegally at the information set *Attacker* is $1 - q$.

Similarly, considering player *IDS*'s behavioral strategies, a probability r must be assigned to the action Prevent at the information set *Illegal Activity*, and a probability s to the action Prevent at the information set *Legitimate Activity*. The probabilities $1 - r$ and $1 - s$ must be assigned to the action Continue, at the information sets *Illegal Activity* and *Legitimate Activity*, respectively.

The established probabilities affect the initial probabilities with which the game starts. In particular, we mentioned before that at the beginning of the game, player *IDS* knows with probability $\frac{1}{4}$ that player *User* is an *Attacker*, and with probability $\frac{3}{4}$ that he is a *Normal User*. This is said to be the player's prior probabilities for the events that the *User* is an *Attacker* or a *Normal User*, respectively. Now, another piece of information is added to the *IDS*'s knowledge. It is the behavioral strategy (p, q), which represents the case of acting Legitimately, whatever player *User* is (*Normal User* or *Attacker*). If player *IDS* knows the

probabilities p and q, because someone wrote them in a game theory book as Binmore says, it should update its beliefs about what player *User* is. These new probabilities are called posterior probabilities, and the process we follow from prior to posterior probabilities is called Bayesian updating.

Assuming that player *User* chooses to play strategy (p, q), the probability that the upper branch will be followed and node (a) will be reached is $\frac{3}{4} * p$ whereas the probability to reach the corresponding node (b) is $\frac{1}{4} * q$. Thus, at the information set *Legitimate Activity*, the posterior probability for player *IDS* when the *User* is a *Normal User*, is

$$\text{prob(User is Normal|User acts Legitimately)} = \frac{prob(a)}{prob(a)+prob(b)} = \frac{\frac{3}{4}*p}{\frac{3}{4}*p+\frac{1}{4}*q}$$

and when the *User* is an *Attacker*, is

$$\text{prob(User is Attacker|User acts Legitimately)} = \frac{prob(b)}{prob(a)+prob(b)} = \frac{\frac{1}{4}*q}{\frac{3}{4}*p+\frac{1}{4}*q}$$

Analyzing player *IDS*'s behavior first at the information set *Legitimate Activity*, we take into account that player *IDS* prefers to Prevent player *User*, when the latter is an *Attacker*. Since the probability at the information set *Legitimate Activity* is $\frac{3}{4} * p$ for a *Normal User* and $\frac{1}{4} * q$ for an *Attacker*, the *IDS* will Prevent the *User* at this node of the game, if the following inequality holds:

$$\frac{1}{4}q > \frac{3}{4}p \Rightarrow q > 3p \tag{6.5}$$

On the other hand, player *IDS* will allow the *User* to Continue at the information set *Legitimate Activity*, if the reverse inequality holds, that is,

$$q < 3p \tag{6.6}$$

Finally, player *IDS* has no interest in choosing either to Prevent or to allow the *User* to Continue, if the probabilities are equal, that is,

$$q = 3p \tag{6.7}$$

Regarding the *IDS*'s behavior at the information set *Illegal Activity*, the player *IDS* will choose to Prevent the *User* from using the TS if the following inequality holds:

$$\frac{1}{4}(1-q) > \frac{3}{4}(1-p) \tag{6.8}$$

Simplifying the inequality (6.8) we get

$$q > 3p - 2 \tag{6.9}$$

Similarly, player *IDS* will allow the *User* to Continue at the information set *Illegal Activity*, if the reverse inequality holds, that is,

$$q < 3p - 2 \tag{6.10}$$

Finally, player *IDS* has no interest in choosing either to Prevent or to allow the *User* to Continue when his signals indicate *Illegal Activity*, if the probabilities are equal, that is,

$$q = 3p - 2 \tag{6.11}$$

When we examined the existence of Nash equilibria in pure strategies, we faced the case of undetermined choices. In particular, we found that if player *IDS* plays strategy *CP*, then player *User* will choose either strategy *LL* or strategy *IL* as best reply, because their payoffs are equal (5 points). That is to say, we do not know what player *User* will do at this node of the game. As a *Normal User*, he will either act *Illegally* or *Legitimately*, and he is apathetic in choosing whichever strategy. This was the reason we switched to behavioral

strategies, in order to determine all Nash equilibria. The equations (6.7) and (6.11) reveal such cases.

Assuming that the hypothesis (6.11) is true, then it is also true that

$$q < 3p \tag{6.12}$$

So, the conclusion that derives from (6.12) is that, player *IDS* will allow the *User* to Continue at the information set *Legitimate Activity*. Consequently, there is no point for player *User* to act Legitimately when he is an *Attacker*, so he will better decide to act Illegally. Besides, this might be closer to his aims and temperament. Therefore, it should be $1 - q = 1$, which results in $q = 0$. But then probability p can be calculated by (6.11), that is $p = \frac{2}{3}$. As a result, there is a Nash equilibrium with $q = 3p - 2$, when $q = 0$ and $p = \frac{2}{3}$.

To make it meaningful, player *User* will decide to act Legitimately with probability $\frac{2}{3}$ if he is a *Normal User*, while he will definitely decide to act Illegally with probability 1 if he is an *Attacker*. Moreover, player *User* will play Illegally with probability $\frac{1}{3}$ if he is a *Normal User*.

Furthermore, the following equation must hold when examining player *IDS*'s behavior at the information set *Attacker*:

$$(-2)r + 4(1 - r) = 3 - 2 \tag{6.13}$$

Solving Equation 6.13 we get $r = \frac{1}{2}$. Consequently, the next equation must also hold:

$$(-1)s + 3(1 - s) = 2 - 1 \tag{6.14}$$

Solving also Equation 6.14 we get $s = \frac{1}{2}$.

Likewise, at the information set *Normal User*, the following equation must hold:

$$0r + 2(1 - r) = 1 - 1 \tag{6.15}$$

Solving Equation 6.15 we get $r = 1$. Consequently, the next equation must also hold:

$$(-1)s + 2(1 - s) = -1 + 0 \tag{6.16}$$

Solving also Equation 6.16 we get $s = 1$.

Decoding these findings, we realize that the *IDS*'s behavior is indifferent between Preventing an *Attacker* from acting either Illegally or Legitimately ($r = s = \frac{1}{2}$). In addition, it sounds strange that the *IDS* will Prevent a *Normal User* to continue using the TS for sure, when acting either Illegally or Legitimately. All these happen when assuming that initially player *User* is a *Normal User* with probability $\frac{3}{4}$ or an *Attacker* with probability $\frac{1}{4}$. If the initial probabilities change, then the above results will be affected significantly. Specifically the probability with which a *Normal User* acts Illegally ($\frac{1}{3}$) will be decreased, if we decrease the initial probability $\frac{1}{4}$ with which a *User* is an *Attacker*. This is really high to be true.

Next, if the hypothesis (6.7) is true, then the inference is that player *IDS* has no interest in deciding either to Prevent or to allow the *User* to Continue, if the probabilities are equal. But from Inequality 6.12 we know that there is a Nash equilibrium when $q < 3p$. Therefore, there is no way to have another Nash equilibrium when $q = 3p$.

6.4.3 Solving with Gambit

Solving the game illustrated in Figure 6.1 with the Gambit tool, we get six profiles from which two are not Nash equilibria. The profiles and the corresponding calculated probabilities are presented in Figure 6.3.

Figure 6.3: The Gambit's solution of the ID signaling game

6.5 Summary

In the ID signaling game we met again, as in Chapter 5, the problem of multiple NE. Therefore, there is a need for Nash equilibrium refinements, in order to choose one that might be selected. But also, the ID signaling game shows the need for defining new signals, which will support its formulation and will give us a better understanding in the interactions that take place between attackers and IDSs. This is discussed as future work in Chapter 9.

Chapter 7

Calculating QRE: Beyond the NE Solution Concept

> It has been said that man is a rational animal. All my life I have been searching for evidence which could support this.
>
> Bertrand Russell (1872 - 1970)

In Chapter 7, we explain first the reasons we employed another solution concept of the Theory Games, the Quantal Response Equilibrium (QRE). We briefly describe the QRE and then we calculate it for the ID game with an insider, to predict insider's future behavior, as an extension of the work presented in Chapter 5. The results are discussed and illustrated graphically to interpret the efficiency of QRE. Finally, we summarize the results obtained in two different periods of the game repetition, and we compare the results with the corresponding when the NE solution concept is used.

7.1 Quantal Response Equilibria - QRE vs. NE

Solving the ID game with an insider we located a unique NE in the stage game (see Section 5.4). But, trying to solve the repeated form of this game, we located 11 NE when the game repeats only for two periods. Considering the rate this game expands as the time periods proceed (see Section 5.2.3), we realize that the problem of multiple NE and which one to select is the most important we face when using the NE solution concept. As mentioned in Section 1.2, the Theory of Games has not a method to systematically check whether any one of these NE is the actual solution of our game, and if so, to indicate which one. Moreover, the backward induction, which is used in extensive form games with multiple NE, unfortunately is not sufficient to solve this problem, especially when the tree is extremely expanded.

Furthermore, according to the motivation of our work (see Section 1.4), we have been focused in predicting the behavior of a user for future actions, in order to prevent an incoming attack. This has been incorporated in the construction of the ID game, the generic one and that with an insider. The way the players play the game, the outcomes of the game, their preferences over these outcomes, they are all carefully examined in such a way to allow behavioral prediction. Besides, the Theory of Games have been especially chosen to serve as a tool for behavioral prediction. But, is the most commonly used solution concept, the well known NE, sufficient for such a requirement?

We recall Halpern and his work in [65] referred in Section 2.1, who enumerates and discusses the problems of NE and the problems that can be observed from a Computer Science point of view. Among them is that NE does not handle unexpected behavior or any erroneous behavior. In our approach, we have concluded that an insider has the intention to attack the TS, but we do not know when and how (see Section 5.4). Even by determining the discount factor δ (see Section 5.5) we solve the same game with different payoffs that might lead to the same problems of NE. In addition, we have defined in insider's action set

the M action for mistakes, to characterize his erroneous behavior.

Selten and Chmura also compared experimentally five stationary concepts for 2x2 games [162]. Among them were the Nash equilibrium and the quantal response equilibrium. Experimental findings indicated the insufficiency of mixed Nash equilibrium in predicting players' behavior. This verifies a ten years older conclusion stated in [55] that characterizes the Nash equilibrium prediction *very bad* in some games. Consequently, we directed our work towards QRE, beyond the classical computation of NE, where the bulk of the literature has focused.

McKelvey and Palfrey defined the Quantal Response Equilibria (QRE) for normal form games [120] and for extensive form games [121], as a probabilistic way to model games and evaluate them, to capture players' bounded rationality. The QRE is analogous to the logit function[1]. It is a generalization of Nash equilibrium, which has also been used to give reasons why players deviate from the equilibrium path. In particular, QRE has been used in signaling games, centipede games, two-stage bargaining games, and overbidding in auctions to explain the irrational players' behavior [61]. The Gambit [119] tool provides methods for computing the logit quantal response equilibrium correspondence for games in both extensive and strategic form, using the tracing method of Turocy [174].

7.2 Computing the QRE for the ID Game with an Insider

We first calculated the QRE in the one shot game depicted in Figure 5.1, and the results verify the NE located previously. That is to say, there is no interest when calculating the QRE of this one shot game. Taken into account that the number of outcomes in the second period of the game is really big (192 outcomes as calculated in Section 5.2.3), and that we are only interested in predicting the insider's behavior and not the IDS's, we decided

[1]The logit function of a number that ranges between 0 and 1, as probability does, facilitates the binary interpretation of an outcome, and is given by the formula: $logit(p) = log(\frac{p}{1-p}) = log(p) - log(1-p)$.

to study the extensive form of the game, in which player I moves twice, before and after player D. Osborne and Rubinstein illustrate and examine an analogous example in [135]. Because the Gambit tool, used to create the game and calculate the QRE, requires manual data input, it is not possible to extend the game to next periods, given the large outcome spaces calculated in Section 5.2.3.

Consequently, we extended the game to include another action, i.e. player I moves first, player D acts next as a response to his action, and player I moves again. The number of outcomes is significantly less, i.e. only 52. Then, we adjusted the payoffs of the extended game to reflect the preferences over the outcomes, following the same method as in the stage game, and we constructed the corresponding utility function.

The new game has the features of a repeated game. Locating all possible NE with the Gambit tool, we get four NE; in all of them, player I's first move is an A action, whereas his second move is either a N, a M, a P, or an A action. Then, which one will be the actual set of moves chosen by the insider, and how the IDS will react? This is the equilibrium selection problem, which remains an open question in Game Theory. The QRE solution concept is an attempt to address this problem [121], by computing the probability with which each action would be selected.

Consequently, we calculate the QRE that will show us how this game would actually be played. Calculations start with equal probabilities for each strategy. Because there is a set of four strategies for each player, every strategy has a probability of 0.25 to be selected. This is the starting point of calculations with $\lambda = 0$ at step 1. The λ is a logit precision parameter, in reliability theory also known as the hazard rate. This method assigns equal probabilities to every strategy to be selected, just the contrary of what the best responses method does when calculating NE.

In appendix 9, Table 9.1 presents the data from selected steps of these long calculations. The first column counts the steps of the QRE calculations; the second column gives the λ

value at each step, and three sets of four columns follow, each for a player's information set. The first information set is player I's first move, the second is player I's second move, and the IDS information set is player D's single move.

7.3 Interpreting QRE

There are several interpretations of QRE [63]. We consider QRE as a generalization of Nash equilibrium to capture players' bounded rationality. It is remarkable that, up to step 10, no probability has changed significantly to show any preference, except the slight growing of the A action at the first move (col.6). Continuing the calculations, it is step 57 when the selection by player D of an S action becomes certain ($p = 1$ for $\lambda = 3.317$). Interestingly, the probabilities of M and P actions, which belong to player I's 1^{st} information set, have been already eliminated at this step. The same holds for M actions of the 2^{nd} information set. This reveals that an insider will avoid mistakes as a first or a second move, and pre-attack actions as a first move.

Looking at the 2^{nd} information set, in step 57 ($\lambda = 3.317$) the probability of a P action has reached its highest value, 0.384, higher than the final it gets at the end of calculations (0.33). Similarly, at the same information set, the probability of an A action is 0.343, while it also ends at the value of 0.33. These slight differences show the intention of player I to choose between a P and an A action, rather than an M action that continuously decreases from the first step of calculations, or a N action that shortly increases from the 4th to the 6th step.

Next, at step 67, an A action at the 1^{st} information set is for sure the best choice of player I ($p = 1$ for $\lambda = 7.985$). In addition, the probabilities that correspond to the 2^{nd} information set have been split almost equally into N, P, and A actions. This result will not change until the last step, step 191, where λ exceeds the value of 1,000,000, which is the end point of the QRE calculations. The Gambit tool has set a threshold and stops the

calculations, when λ reaches the value of 1,000,000 and just above it.

The following two figures present the growth of parameter λ in accordance with the actions probabilities for the 1^{st} and the 2^{nd} information set correspondingly. The x axis measures the λ logit precision parameter. λ ranges between 0 and ∞. The y axis represents the probability a player will choose a certain strategy. It ranges between 0 and 1. On the line diagram, each of the four actions for player I, is colored differently.

Reconsidering the results, Figure 7.1 depicts diagrammatically the probabilities calculated for the 1^{st} information set, revealing player I's intention to harm the system from the first action, if this is also the last one. N, M, and P actions have the lowest probabilities to be selected at the 1^{st} information set.

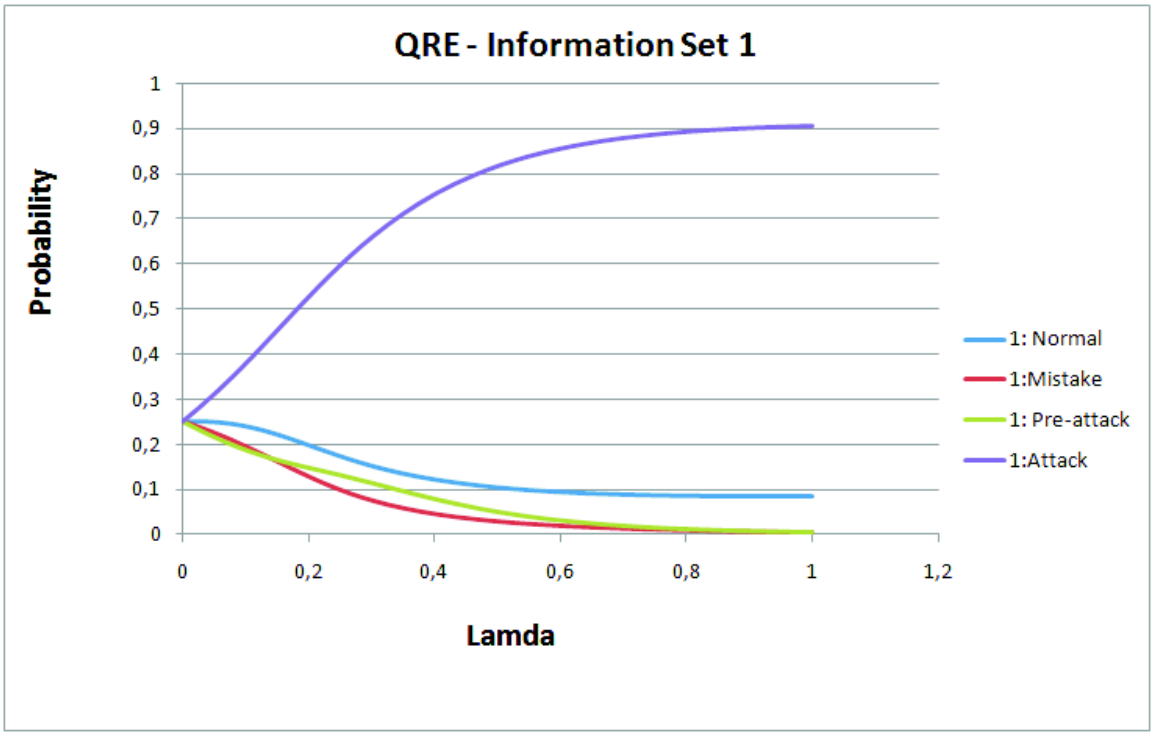

Figure 7.1: QRE - Information Set 1

But because the game is not a one shot game, Figure 7.2 illustrates player I's intentions when playing for second time. According to these, player I will choose between a N, a P, and an A action with nearly equal probabilities, and will avoid mistakes. Notice that the QRE is not the same as the calculated NE.

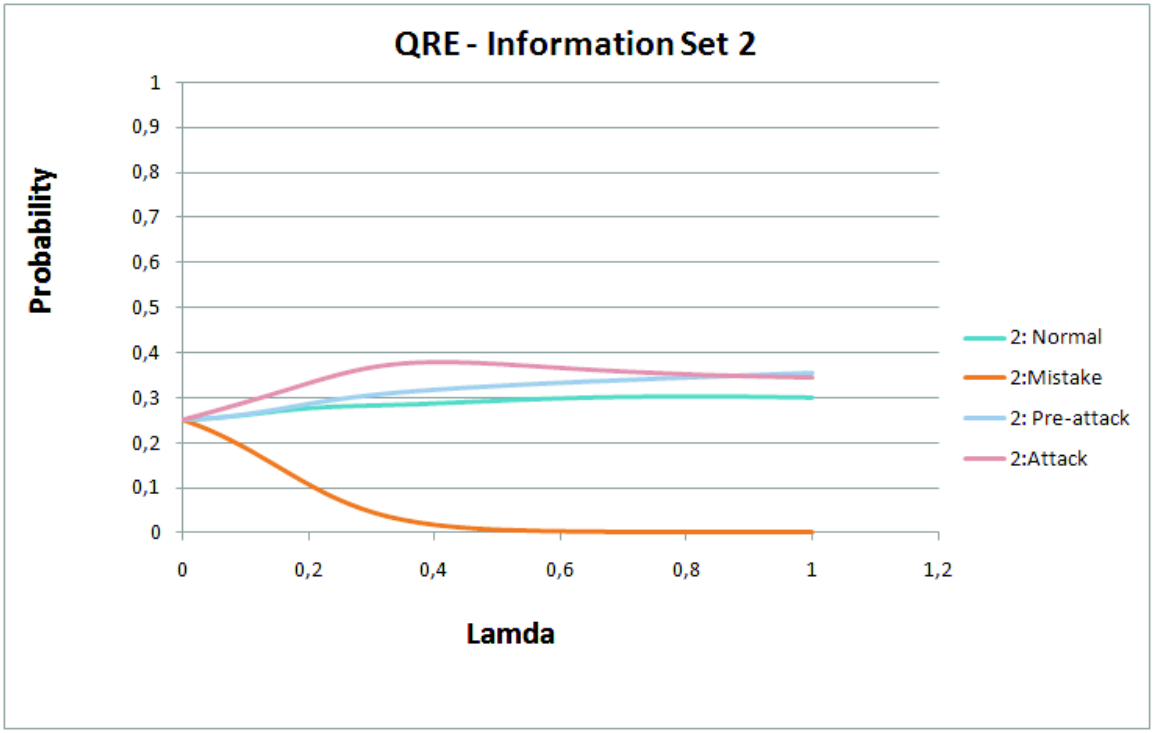

Figure 7.2: QRE - Information Set 2

Concluding, the QRE calculations present an insider's intention to deviate from the calculated equilibrium path at his second move. In other words, although player I moves first by selecting an A action, if player D responds with an action other than an S action, then player I has equal probabilities to choose between an N, a P, or an A action as a second move. Even though the IDS has left an attack undetected (false negative alarm) at real time, there is another chance to predict future behavior and intentions of an insider, and prevent subsequent attack attempts and further system damage. But because the user is an

insider, it is unlikely to choose to attack the TS from his first move. Therefore, the results obtained for the 2^{st} information set, where the M action receives no probability at all, expose very valuable conclusions regarding this type of insider with the certain preferences described in Section 5.2.2.

Figure 7.3 summarizes the QRE calculations for player I. On the x axis the λ logit precision parameter changes in three time periods.

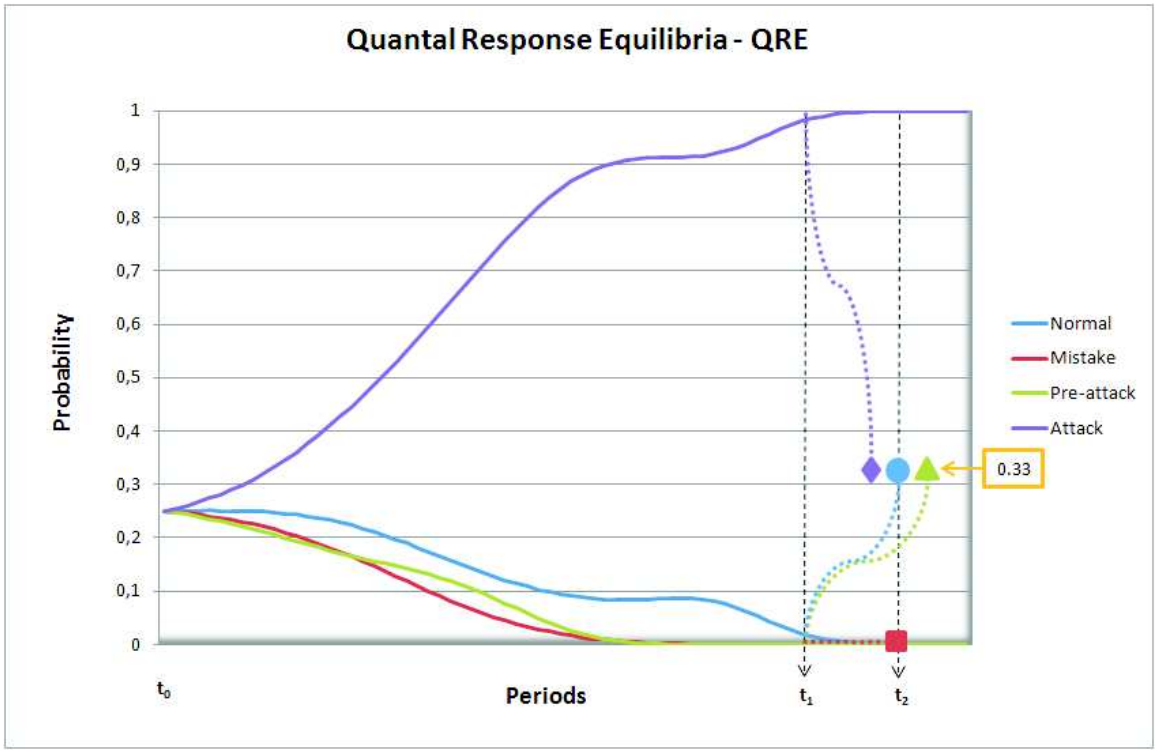

Figure 7.3: QRE calculations for player I in three moves repeated game.

Period t_0 is the start point of QRE calculations with equal probabilities for each strategy, before any player's move. Period t_1 is QRE calculated probabilities for player I's first move, where the A action is the most likely selection with probability 1. Accordingly, period t_2 is QRE calculated probabilities for player I's second move, where N, P, and A actions share

the same amount of likelihood to be selected, $p = 0.33$. As a final point, this diagram depicts briefly the estimations of the insider's intentions to move, which derive from the QRE calculations.

7.4 Summary

We have extended the NE notion to the logit Quantal Response Equilibrium (QRE), to capture players' bounded rationality and model insider's behavior. For this, we considered the ID game with an insider for two periods and calculated the QRE. The results are more realistic, and show that the solution of the game might be significantly different than the corresponding NE solution. Thus, we determine how an insider will interact in the future, and how an IDS will react to protect the system.

Chapter 8

Game-based ID Application Models to Ensure Trust and Reputation

> One can survive everything, nowadays,
> except death, and live down everything
> except a good reputation.
>
> Oscar Wilde (1854 - 1900)

In this chapter, we propose two possible implementation schemes to apply the findings of our research work in Intrusion Detection in IT Security. The first is a *game-based Intrusion Detection model* constructed appropriately to incorporate the results obtained by the use of the Theory of Games. This model represents a *game-based architecture* for the field of Intrusion Detection. The second is a *Detection Mechanism* to easily exploit QRE results in Intrusion Detection. For the *Detection Mechanism* we give the application model and detailed game-based detection algorithms to verify its operation.

8.1 1^{st} Implementation Scheme: A Game-based ID Model

In the first implementation scheme, we consider the development of a new Intrusion Detection System that will consist of an entire game-based detection engine, without excluding other anomaly or misuse detection engines that might be combined with it. In the sequel, we present the components of this model and we discuss how they cooperate to make the IDS operate properly.

8.1.1 The Architecture

The model performs four main functions. It records events, it correlates events, it detects attacks, and it takes countermeasures as a defense when an attack is detected. Therefore, it was decided that the model consists of ten components: seven modules, two databases, and a "library". The model's architecture is consistent with Bishop's general architecture of an Intrusion Detection System [23]. In Particular, the model components are the following:

1. Move Recorders (MR)
2. Moves Data Base (MDB)
3. Correlator (C)
4. Correlated Moves Data Base (CMDB)
5. Maintenance Procedures (MP)
6. Detection Engine (DE)
7. Game Library (GL)
8. Game Creator (GC)
9. Configuration Mechanism (CM)

10. Defense Mechanism (DM)

In the sequel, each component is described in detail to reveal the features and functioning of the model. Figure 8.1 depicts an overview of the proposed model design and identifies the links that connect its components.

1. **Move Recorders (MR)**

   A Move Recorder (MR) is a module that resides in the Target System (TS). It is designed as an event collector from any source of the TS, that is, from a host, a network, or both. It records any event as it happens and stores it in the Moves Data Base (MDB), in two different standard format, one that serves as a node of an extensive form game (tree format), and another one that serves as a cell in normal form game (tabular format). Each record format is appropriate to keep all the necessary information of an event as a move of a game. For this reason, the operation of this module justifies its name. There are many Move Recorders in a Target System that operate as Agents in accordance with Bishop's architecture.

2. **Moves Data Base (MDB)**

   The model maintains a special database for all the recorded events, the Moves Data Base (MDB). This database accumulates a large number of records, each of them represents a move (an action) of an extensive form game. To ensure the integrity of the whole database, the integrity of every record, and the integrity of every piece of information stored in this database, strong cryptographic schemes are used.

3. **Correlator (C)**

   Today, most if not all the attacks that take place are multi-step attacks instead of single-step attacks. Therefore, to successfully detect such attacks, the individual steps should be correlated to form the complete attack scenario. The Correlator module

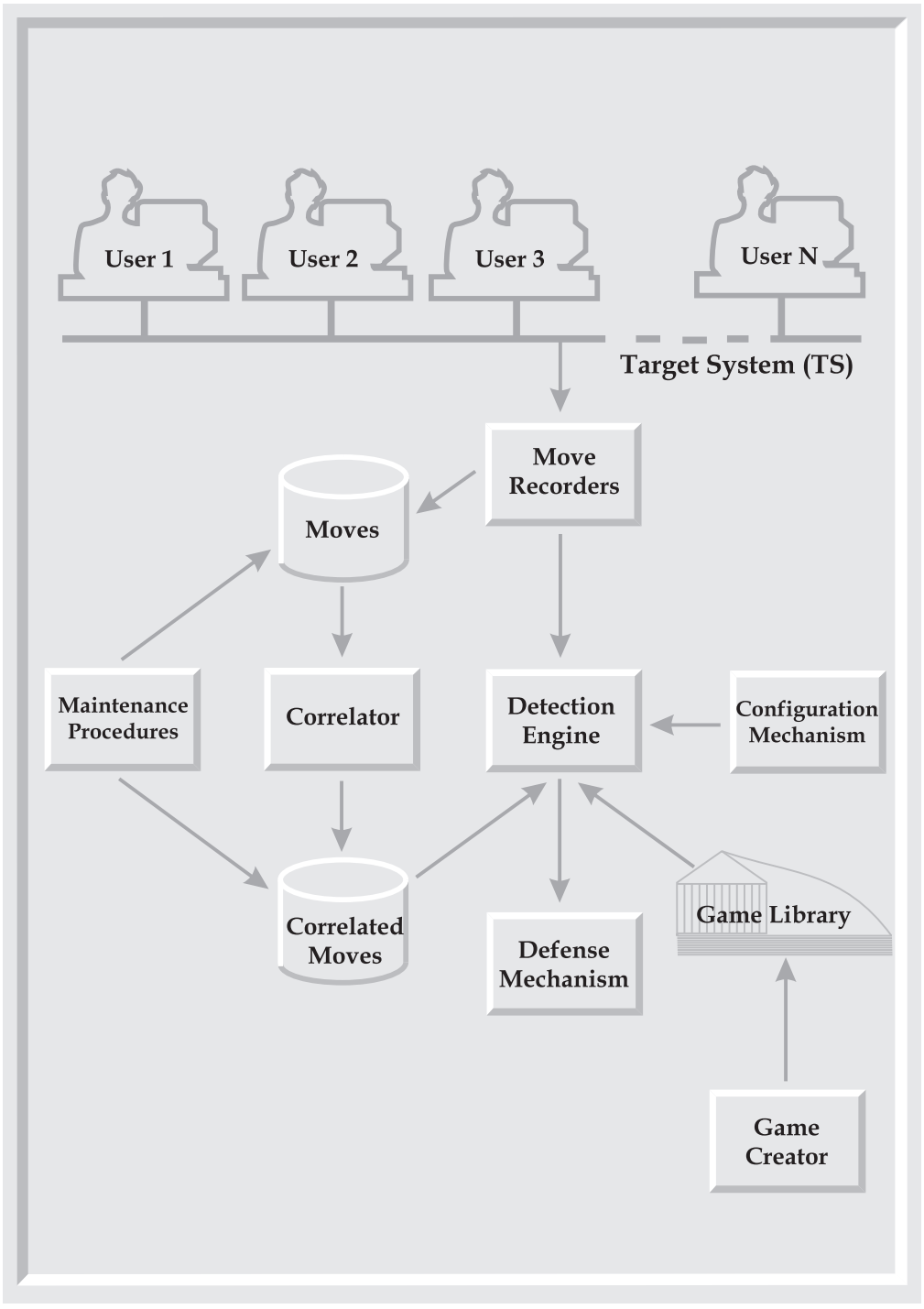

Figure 8.1: The Architecture of the Game-based Intrusion Detection Model

aims at associating different events, that is single moves of a game, by grouping together records stored in the Moves Data Base. Once a correlated record is formed by the Correlator, it is stored in the Correlated Moves Data Base.

4. **Correlated Moves Data Base (CMDB)**

   In addition to the Moves Data Base, another database is necessary to keep all the combined records that form multi-step attacks. The Correlated Moves Data Base has a smaller amount of stored records than the Moves Data Base. Its record format is similar to a tree data structure, to keep information related to a chain of moves, instead of a stand alone move. The need for granular integrity from the whole database to one bit of it, drives towards the use of special security mechanisms for the database protection.

5. **Maintenance Procedures (MP)**

   Both databases require a set of procedures for their maintenance. Procedures for record backup and record deletion are minimum requirements to reduce the bulks of the databases used in the model. In addition, any security mechanisms used to protect these databases involve the use of additional procedures for them, to provide their installation, use, configuration, and maintenance.

6. **Detection Engine (DE)**

   The Detection Engine is the main module of the proposed model. As long as the Intrusion Detection Model exists and works on a Target System, the Detection Engine should perform its tasks in an optimum way, achieving the highest detection rates with the lowest generation of false alarms.

   In this model, the Detection Engine is a bi-functional module. Simultaneously, it carries out two processes. The first process is similar to a matching problem. The difference is that we do not seek a game that corresponds to an attack and as such it

would give us an answer to the decision problem. We are trying to discover the exact game that is in progress, by changing between games stored in the Game Library. This is a game refinement or better a strategy refinement. Games are handled in normal or strategic form, that is their information is in tabular format. Whenever a user player acts, the Detection Engine checks if a game from the Game Library matches the current one. If so, it examines subsequently the solution to a minmax problem which is already set. If the player's action is different from the one expected according to the selected game, then the engine changes to another game that matches the new action, etc. One of the benefits in this approach is that when a game has been chosen as the matching game with the current one, the solution is known, so the engine can proceed to the detection immediately, to be precise, in real-time.

The second process is the online construction of the game played at a point of time modeling and solving the minimax problem. Once a player starts using the Target System, a play is initiated. Each action performed by this player, makes a node in an extensive form game, that is, it is a new move that carries additional information for the type of game it is played. This information is recorded in a tree data structure form. Given this amount of information, the Detection Engine examines if the player is a maximizer, that is, if the player has already followed a path from the tree maximizing his payoffs, or as his line of direction shows, he has decided a route from the tree that leads to payoff maximization. In any of these cases, the player is suspicious and the engine should signal an alarm. To decide whether the player is maximizing or he is intending to maximize, the tree must be searched. Searching big trees is time consuming and difficult to produce real-time answers. Besides, exhaustive tree search looking for max payoffs degrade significantly the IDS performance and the TS's operation and performance. Therefore, special methods should be used to overcome such problems.

These two processes are complementary in the sense that when the one gives first a solution, then detection is completed and this is considered as the output of the detection procedure. When the other process gives a solution too, then the Detection Engine examines both results and gets feedback for its configuration and tuning.

7. **Game Library (GL)**

   The Game Library (GL) is a special component of the proposed model. It is a collection of games that serves as a repository of all possible games that might be played between a player or a group of players and an Intrusion Detection System. It does not have the structure and the usage of a database, and that is why it has been clearly distinguished from the two databases described above. Once it is created, no one modifies it because there is no need for adjustments. In case of destruction, there is a special module that recreates it, the Game Creator (GC). Every game in this model is considered to be in a normal form so that an array data structure can easily implement it. The whole library has a great number of information kept in tables which correspond to the payoff matrices of normal form games, each of which represents a single game. There is a clear distinction between a base constructed with attack signatures used in an anomaly-based IDS and the Game Library.

8. **Game Creator (GC)**

   The Game Creator (GC) is a game constructor that sets up the Game Library (GL). Normally it works once and in cases of library destruction it regenerates the library. The Gambit toolkit includes a specific language for the generation, modification and the solution of games. This language can be used for a simple program development that will create games for the Game Library, at a level of model prototyping. Consequently, a Game Creator should be developed as a special tool that will work with certain method, for the construction of every possible game that will compose a com-

plete Game Library. The Monte Carlo algorithm shows the potential and feasibility of being incorporated into the Game Creator module.

9. **Configuration Mechanism (CM)**

   Upon completion, an Intrusion Detection Model needs configuration and tuning. Its operation shows the pros and cons of every individual component and of the whole model as a system. A separate component, the Configuration Mechanism (CM), provides a wide range of capabilities and options for fine tuning, aiming at optimal results by increasing the detection rate and eliminating all the false alarms. In order to operate this powerful mechanism for the model configuration, there is a need for special metrics setup that would give essential figures related to the IDS performance. These figures would underline the whole configuration process.

10. **Defense Mechanism (DM)**

    When a suspicious player has been detected, the Detection Engine will raise an alarm. The Defense Mechanism (DM) module will rate the severity of the event and the grade of urgency, based on the information related to the game payoffs. The produced ranking will offer the starting point for the selection of the proper countermeasure among a short list of available defense actions. Next, the selected countermeasure will be carried out as a first and immediate action to protect the Target System from further damage.

8.2 2^{nd} Implementation Scheme: A Detection Mechanism

An IDS, as the open source Snort®[1], could be installed and configured to cooperate within the detection mechanism. Then the detection mechanism will receive its output as input and go through an algorithm to predict the user's intent. Depending on the result, the

[1] Snort® is a Registered Trademark owned by Sourcefire, Inc.

detection mechanism must have an option to adjust players' preferences, and reassign numbers that represent payoffs. Such an adjustment will optimize its operation and eventually will perform better in the interactions with the insiders of the system. Detecting insiders is not only connected with punishment, but also systematically aims at pointing in the right direction every user of the system. This implementation scheme has been designed towards this direction.

Moreover, because an IDS has already been integrated in the kernel of an operating system (the Linux Intrusion Detection System (LIDS) as a patch to the Linux®[2] kernel), the feasibility of including such a detection mechanism in the kernel of an operating system that will cooperate with an IDS should be considered as an alternative.

8.2.1 The Application Model

Figure 8.2 gives a picture of the data flow application model of the proposed scheme. The model consists of six processes, an interface, and a data source. The data flow starts with the *Target System* (TS) that is being monitored by an *IDS*. The *IDS* captures the events of the *TS* and uses a detection engine to decide whether an event is normal or abnormal. The detection engine might incorporate two or more of the three well known detection techniques, an anomaly, a misuse, or a specification-based detection technique (see Section 1.1 for details). Upon completion, the *IDS* characterizes the event as normal or abnormal, and sends a message to the *Security Officer*'s interface to inform him. Depending on the *IDS* design and operation, the result might be more concrete and might include the cases of mistakes and of pre-attack actions. The *IDS* result is also transferred to the *Connector* that will combine it with the result obtained by the *QRE Calculator*.

Existing IDSs feature a configuration option, allowing a Security Officer his intervention to adjust their operation, and make them more reliable and accurate. In the present model,

[2]Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

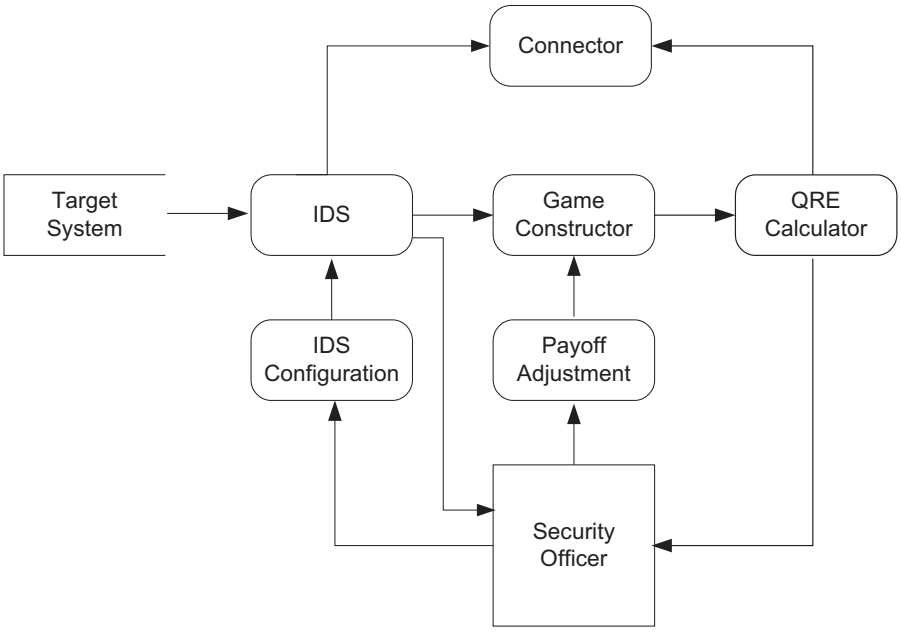

Figure 8.2: The Application Model of the proposed implementation scheme.

the *IDS Configuration* process implements this feature. On the other hand, the *Game Constructor* is the process that builds up a game, based on the actions that have been played so far. It receives the output of the *IDS* process and formulates the tree path, as it has been defined by the players' moves. The preferences and the payoffs are defined manually first by the *Security Officer*, and the *Payoff Adjustment* process allows him to continuously alter players' preferences and reassign payoffs to them, in order to optimize the detection mechanism's operation.

The heart of the model is the *QRE Calculator* process that receives input from the *Game Constructor* and implements an algorithm to calculate the QRE probabilities, following Turocy's method [174]. The QRE calculations require the definition of the maximum value of the logit precision parameter λ, to ensure the end of the calculations. When the *QRE Calculator* terminates, the output of the process, the calculated probabilities, will be sent to the *Connector*. The *Connector* will combine the results obtained by the *IDS* and by the

QRE Calculator. The overall aggregated result will be used for immediate counteraction to prevent further damage.

The output of the *QRE Calculator* will also be displayed on the *SO*'s interface, to inform him about the intention of the user to play normally or abnormally in the next move. The *SO* on his turn, taking into account the output of the *QRE Calculator*, is able to choose between two options, to configure appropriately the *IDS*, through the *IDS Configuration* process, and avoid a forthcoming attack, or adjust the payoffs of the game, through the *Payoff Adjustment* process, in cases where the QRE results conflict significantly with those derived from the *IDS*. Generally, QRE is used for payoff adjustment and correction when considering real data.

As presented in Section 7.3, it is feasible to calculate the QRE probabilities for the current and also for the next stage of a game. Taking advantage of this, we have drawn two ways the QRE modeling affects the IDS. First, by combining the probabilities derived from the *IDS* and from the *QRE Calculator*, within the *Connector* process. The overall aggregated result will be taken into account for immediate counteraction, if necessary. Second, by informing the SO with the probabilities calculated for a user's next move that might prompt the SO to trigger the *Payoff Adjustment* process, or the *IDS Configuration* process, or both, in order to avoid a forthcoming attack. Especially if the two results are conflicting or deviate significantly, then it is essential to configure the IDS and adjust the payoffs of the constructed game.

8.2.2 The Game-based Detection Algorithm

For every event that is being logged, a game-based detection algorithm has been drawn in Figure 8.3, to illustrate the functionality of the proposed implementation scheme. Considering that the joint IDS includes a module for event capturing, filtering and storing in a data store, the algorithm starts with the *Intrusion Detection Engine* process, which gets a record

from the *Filtered Data* store. The records in the *Filtered Data* store contain information of the security relevant events only, in a predefined standard format.

The *Intrusion Detection Engine* examines the obtained record according to the implemented detection algorithm, which might encompass any combination of the known intrusion detection techniques. For example, Snort incorporates both the anomaly and the misuse detection technique. Eventually, the output of this process will be either an intrusion alert, or information related to the event classified as normal. This output is reported to the Security Officer's monitor in real time, to notify him about the IDS operation and the Target System's state.

Every IDS result is combined with history results, to calculate important detection parameters. The SO assesses the detection parameters and decides to configure the IDS or not. The *IDS Configuration* process is a feature that allows the SO to reduce or increase thresholds, to alter rules, to change or add attack signatures, etc. At the end of this process, the core of the IDS has been modified, and the SO continues supervising its operation to ensure improvement.

In the next step, the *Game Construction* process will first check whether this is the initial event triggered by the corresponding user, or his actions have already created a game that is being played to this point. If the game has already been created, then the *Game Construction* process retrieves the corresponding game from the *Stored Games*. Otherwise, it constructs a new game and associates it with the specific user. The new game will consist only of two moves, whereas the retrieved game will be updated to include the last actions.

The game data, which comes out of the *Game Construction* process, is the actions played at each period of the repeated game and the calculated payoffs. Then, the game data enters the *Quantal Response Equilibrium Calculations* process and following Turocy's method [174], the probabilities for the current and for the next actions are determined to reveal the intention of the user for his future actions. The action probabilities are being

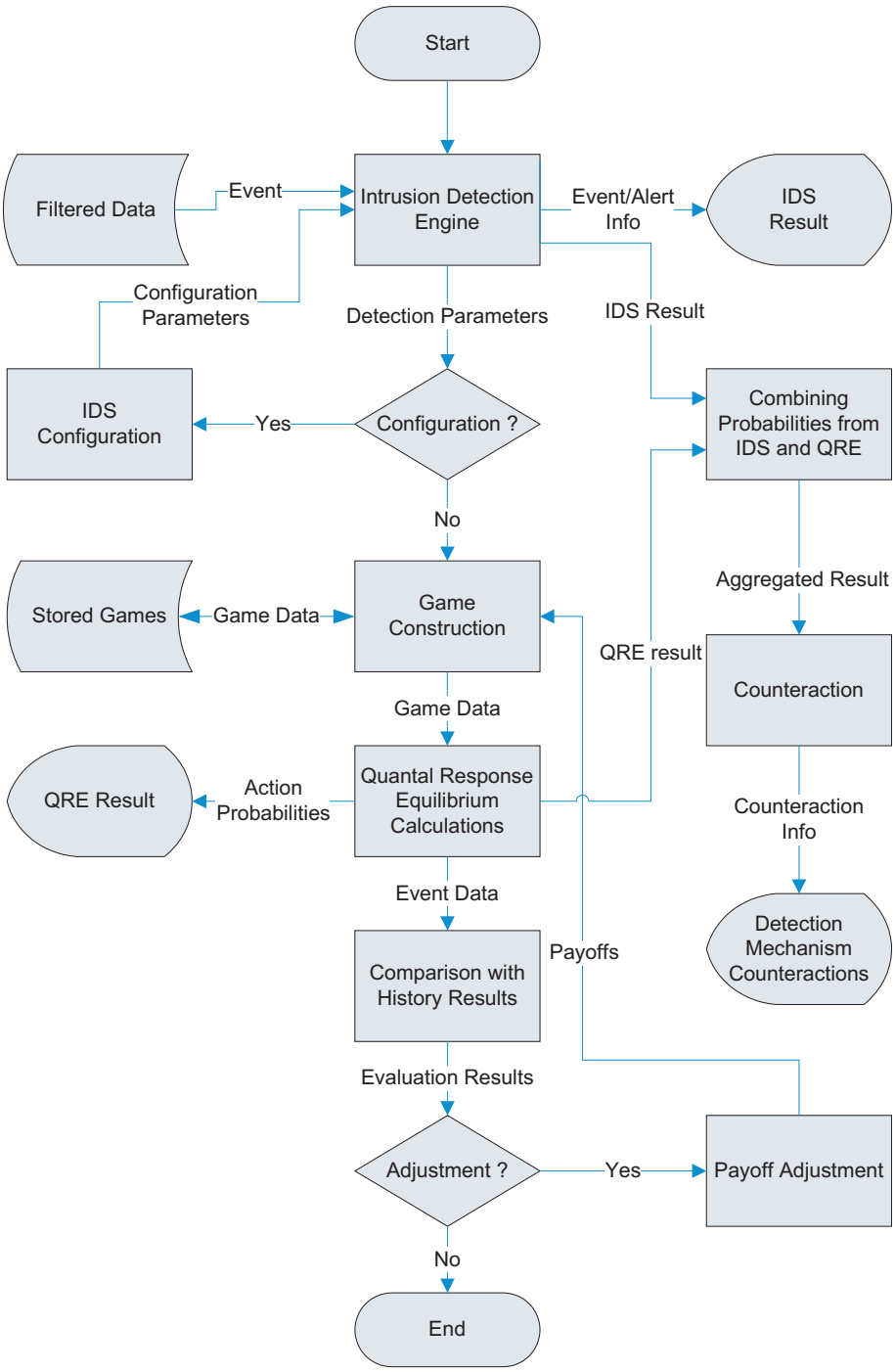

Figure 8.3: The Game-based Detection Algorithm

displayed onto the Security Officer's monitor in real time, and the SO examines the findings.

Then, the previous QRE result is compared with the current IDS result, through the *Comparison with History Results* process. The derived evaluation results will be judged by the SO, to adjust or not the game with the specific user. In practice, this denotes that players' preferences are under review, and that payoffs will be altered accordingly, through the *Payoff Adjustment* process. The last output will enter the *Game Construction* process to modify the stored game.

Both the IDS result and the QRE result are transferred to the *Combining Probabilities from IDS and QRE* process, in order to aggregate a final decision result. Assuming that the IDS result is the probability an event to be normal or not, and as the QRE result is the probability distributions over the players' actions, a method of aggregating an overall decision is used in this process, to resolve any conflicting information and improve the IDS effectiveness. The simplest method to combine probabilities is to average them. Another one is Winkler's normal model [182]. A study on combining probability distributions in risk analysis is provided by Clemen and Winkler [39].

Then, the aggregated result will be sent to the *Counteraction* process, so that, the proposed detection mechanism will counteract against a forthcoming attack. There are two general counteraction types, *continue* and *prevent*. The first one allows the user to continue working with the TS. The second one includes all the counteractions that gradually remove privileges from the user, reduce the amounts of his allocated resources, and place him at the proper position of a black list. The last line of defense is the *stop* counteraction that ends the user session, and adds him in the worst place of the black list. The counteraction information will also be displayed on the *SO*'s monitor, to inform him properly.

The algorithm starts again when a new event is recorded in the *Filtered Data* store. For better results, it is preferable for the SO to have also special knowledge on game theoretic aspects, to manage effectively all the components of the proposed detection mechanism.

8.3 Summary

To facilitate the incorporation of the Theory of Games and the exploitation of QRE results in Intrusion Detection, we proposed two possible implementation schemes. The first is to develop a new IDS that will consist of an entire game-based detection engine. The second is to use an existing IDS and develop only the *Detection Mechanism* that will cooperate with it. Because the second implementation scheme keeps the application model too simple, and because we wanted to concentrate on the operability of the *Detection Mechanism*, we described the entire *Game-based Detection Algorithm* to some extend.

Both schemes operate in a two-folded way. The Intrusion Detection System follows a trust line in order to avoid false positive alarms. On the other side, a user player of the Target System estimates its reputation and because he does not prefer to be in a black list or to work with less privileges, services, or resources, he tries to be a "right" user who follows the security policy of the system. In a long time, he will be benefited by this attitude.

Chapter 9

Conclusions and Open Questions

> The point of philosophy is to start with something so simple as not to seem worth stating, and to end with something so paradoxical that no one will believe it.
>
> Bertrand Russell (1872 - 1970),
> The Philosophy of Logical Atomism, 1918

Studying the area of Intrusion Detection we realized that the trends have turned IDSs from host-based to network-based, from centralized to distributed, and that IDSs need to be platform independent in order to expand their success. Moreover, among the standards that specify the area are the hybrid IDSs, the real time detection, the active counteractions, and the problem of imbalance between true positive, false positive, and false negative alarms. But their flexibility is too limited to allow effective responses to dynamically evolving events. Therefore, there is a need for an evolutionary framework that will assist Intrusion Detection to be more effective and fulfil current requirements.

To overcome the existing ID techniques and make IDSs work with a complete different approach, e.g. a game theoretic one, new signs must be identified and new methods must be developed to read these signs. Then, a game-based detection engine will be efficient to

predict users' future actions, recognize their behavioral intentions, and thus detect attempts of attacks rather than completed successful attacks. Consequently, false alarms would be reduced, positive and negative, which are an important problem to be solved in ID.

The problem in ID is practically unsolvable, but approximate solutions can always give satisfactory results. There is no IDS with 100% detection rate, no false alarms, and no tremendous system overhead, and cannot be developed (see Section 1.1). Considering such a dead end, we were directed our focus on how we can improve the field of Intrusion Detection, rather than trying again to solve the problem of signal identification. In this regard, we have chosen an approach for behavioral detection given that an IDS identifies with a certain accuracy a user's actions.

The proposed model addresses the problem of predicting an insider's behavior, so that, the appropriate strategies would be chosen to avoid a forthcoming attack. To extend and direct future work from this focal point and forward, mechanism design should be used. Then, the game theoretic approach will be able not only to predict a user's future action but also to influence his future behavior in such a way that will ensure the Target System's security. In the view of the cooperative nature of security, users and systems should safely interact to keep their own interests.

As Camerer explains, the Theory of Games is *simply analytical*. Even when players choose different strategies from those the theory suggests they should play, *their behavior has not proved the mathematics wrong* [33]. Game model reconsideration and reconstruction, preference alteration, and payoff adjustment are a few practices to handle these situations, and they all have been considered and included in our model.

Similarly, Dixit and Skeath[51] accept that *the Theory of Games provide some general principles for thinking about strategic interactions, but it cannot hope to offer surefire recipes for action*. This justifies the use of Game Theory in the area of Intrusion Detection so far, and the corresponding deficiency of a game theoretic mechanism able to detect intrusions

in the sense the well known established intrusion detection techniques have been applied in the past. In the view of this drawback, only research work on signaling games might give results satisfactory enough to reverse this conclusion.

In traditional intrusion detection techniques attackers try to deceive the detection engine in order to hide their clues and the committed crime. Because valuable information are stored and special mechanisms to alter them are included in detection engines, attackers focus on steeling or modifying appropriately this information to evade detection. In our approach there is no stored information such as user profiles, or program and system specifications or attack signatures, as in all anomaly-based, specification-based and misuse approaches are used. This adds an advantage to the proposed model regarding the problem of evading detection. An insider, who not only wants to benefit from an attack but also aims at evading detection, will confront one more difficulty in the presence of such a detector. Game Theory itself is eventually another problem to attackers, especially because of its complexity that discourages unaware attackers to discover the algorithm used in the detector.

As for the area of Algorithmic Game Theory, it is not a coincidence that the same person, John von Neumann, who established the *Computer Science* with the well known *von Neumann architecture*, pioneered also a new theory, the *Theory of Games* [132]. It was actually the same time and the same place, in 1944 at Princeton University, when he designed EDVAC and published also the joint work with Oscar Morgenstern for the Theory of Games. This is probably a strong reason why these two disciplines can go together, fit well in several areas, and justify the new era of the *Algorithmic Game Theory* that commenced with their merge.

Future research directions in this interdisciplinary area include the establishment of the credibility of game theoretic approaches adopted in IT security problems. Before extending the Algorithmic Game Theory we must convince the IT security experts of the advantages

of these approaches. Then, the foundation of a new scientific prospect for IT Security will derive.

As Robert Frank describes in [58], in Darwin's model the selection unit is the individual and not the group or the species. It is said that, in front of a choice between an action, which will benefit the others, and another action, which will only serve his own interest, every being has been programmed by the evolutionary powers to follow the second path, the selfish one. This also holds in cases of deception. But, consciousness conveys signals of risk when someone acts illegally. Unfortunately, they are ignored in deceptive situations totally concealed, although there are perfect crimes that went absolutely wrong. Therefore, it is supposed honesty to be the best policy. An attacker, who has been caught once for illegal actions, is placed in a black list, as someone able to repeat an analogous action in the future. Even a less "right" person, is better to overcome any prospects for deception, in order to obtain the reputation of an honest person. This reputation will be beneficial to him in the future.

Bibliography

[1] Abraham, Ajith and Ravi Jain. Soft Computing Models for Network Intrusion Detection Systems. In Saman K. Halgamuge and Lipo Wang, editors, *Classification and Clustering for Knowledge Discovery*, volume 4 of *Studies in Computational Intelligence*, pages 191–207. Springer Berlin / Heidelberg, 2005. doi: 10.1007/11011620_13.

[2] Abraham, Ajith, Ravi Jainb, Johnson Thomasand, and Sang Yong Han. D-SCIDS: Distributed Soft Computing Intrusion Detection System. *Journal of Network and Computer Applications*, 30(1):81–98, January 2007.

[3] Agah, Afrand, Kalyan Basu, and Sajal K. Das. Preventing DoS Attack in Sensor Networks: A Game Theoretic Approach. In *Proc. of the IEEE International Conference on Communications (ICC 2005)*, volume 5, pages 3218–3222, May 2005.

[4] Agah, Afrand and Sajal K. Das. Preventing DoS Attacks in Wireless Sensor Networks: A Repeated Game Theory Approach. *International Journal of Network Security*, 5(2):145–153, September 2007.

[5] Agah, Afrand, Sajal K. Das, and Kalyan Basu. A Non-Cooperative Game Approach for Intrusion Detection in Sensor Networks. In *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, volume 4, pages 2902–2906, September 2004.

[6] Agah, Afrand, Sajal K. Das, Kalyan Basu, and Mehran Asadi. Intrusion Detection in Sensor Networks: A Non-Cooperative Game Approach. In *Proc. of the Third*

IEEE International Symposium on Network Computing and Applications, 2004. (NCA 2004), pages 343–346, August-September 2004.

[7] Alazzawe, Anis, Asad Nawaz, and Murad Mehmet Bayaraktar. Game Theory and Intrusion Detection Systems. Student research project, Computer Science, Carnegie Mellon University, Qatar, Spring 2006. accessed 25 Nov 2010, http://www.qatar.cmu.edu/iliano/courses/06S-GMU-ISA767/project/papers-/alazzawe-mehmet-nawaz.pdf.

[8] Alpcan, Tansu and Tamer Başar. A Game Theoretic Approach to Decision and Analysis in Network Intrusion Detection. In *Proc. of the 42rd IEEE Conference on Decision and Control (CDC)*, pages 2595–2600, Maki, HI, December 2003.

[9] Alpcan, Tansu and Tamer Başar. A Game Theoretic Analysis of Intrusion Detection in Access Control Systems. In *Proc. of the 43rd IEEE Conference on Decision and Control (CDC)*, Paradise Island, Bahamas, December 2004.

[10] Alpcan, Tansu and Tamer Başar. An Intrusion Detection Game with Limited Observations. In *Proc. of the 12th International Symposium on Dynamic Games and Applications*, Sophia Antipolis, France, July 2006.

[11] Androutsopoulos, Ion, Evangelos F. Marigou, and Dimitrios K. Vassilakis. A Game Theoretic Model of Spam E-Mailing. In *Proc. of the 2nd Conference on Email and Anti-Spam (CEAS 2005)*, Stanford University, CA, USA, 2005.

[12] Apt, Krzysztof R. and Erich Grädel (Eds.). *Lectures in Game Theory for Computer Scientists*. Cambridge University Press, January 2011.

[13] Aumann, Robert J. and Michael Maschler. Game Theoretic Analysis of a Bankruptcy Problem from the Talmud. *Journal of Economic Theory*, 36(2):195 – 213, August 1985.

[14] Axelsson, Stefan. Research in Intrusion-Detection Systems: A Survey. Technical Report 98-17, Dept. of Computer Engineering, Chalmers University of Technology, 19 August 1999.

[15] Axelsson, Stefan. The Base-Rate Fallacy and the Difficulty of Intrusion Detection. *ACM Transactions on Information and Systems Security*, 3:186–205, August 2000.

[16] Axelsson, Stefan. Intrusion Detection Systems: A Survey and Taxonomy. Technical Report 99-15, Dept. of Computer Engineering, Chalmers University of Technology, 14 March 2000.

[17] Axelsson, Stefan. A Preliminary Attempt to Apply Detection and Estimation Theory to Intrusion Detection. Technical Report, 2000.

[18] Barbará, Daniel, Ningning Wu, and Sushil Jajodia. Detecting Novel Network Intrusions using Bayes Estimators. In *Proc. of the First SIAM Conference on Data Mining*, April 2001.

[19] Barika, Farah A., Nabil El Kadhi, and Khaled Ghédira. Agent IDS based on Misuse Approach. *Journal of Software*, 4(6):495–507, August 2009. Academy Publisher.

[20] Bavisi, Sanjay. Penetration testing. In John R. Vacca, editor, *Computer and Information Security Handbook*, pages 369 – 382. Morgan Kaufmann, Boston, 2009.

[21] Binmore, Ken. *Playing for Real - A Text on Game Theory.* Oxford University Press, 2007.

[22] Binmore, Ken and Paul Klemperer. The Biggest Auction Ever: The Sale of the British 3G Telecom Licences. *Economic Journal*, 112:C74–C96, 2002.

[23] Bishop, Matt. *Computer Security: Art and Science.* Addison-Wesley, 2003.

[24] Bishop, Matt, Sophie Engle, Deborah A. Frincke, Carrie Gates, Frank L. Greitzer, Sean Peisert, and Sean Whalen. A Risk Management Approach to the "Insider Threat". In Christian W. Probst, Jeffrey Hunker, Dieter Gollmann, and Matt Bishop, editors, *Insider Threats in Cyber Security*, volume 49 of *Advances in Information Security*, pages 115–137. Springer, 2010.

[25] Bishop, Matt, Sophie Engle, Sean Peisert, Sean Whalen, and Carrie Gates. We Have Met the Enemy and He Is Us. In *Proc. of the 2008 Workshop on New Security Paradigms*, NSPW '08, pages 1–12, New York, NY, USA, 2008. ACM.

[26] Bishop, Matt, Sophie Engle, Sean Peisert, Sean Whalen, and Carrie Gates. Case Studies of an Insider Framework. In *Proc. of the 42nd Hawaii International Conference on System Sciences*, pages 1–10, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[27] Bishop, Matt and Carrie Gates. Defining the Insider Threat. In *Proc. of the 4th Annual Workshop on Cyber Security and Information Intelligence Research: Developing Strategies to Meet the Cyber Security and Information Intelligence Challenges Ahead*, CSIIRW '08, pages 15:1–15:3, New York, NY, USA, 2008. ACM.

[28] Brackney, Richard C. and Robert H. Anderson. Understanding the Insider Threat. In *Proceedings of a March 2004 Workshop*. National Security Research Division, RAND Corporation, January 2005.

[29] Burgoon, Judee K., Matthew L. Jensen, John Kruse, Thomas O. Meservy, and Jay F. Nunamaker, Jr. Chapter 8, Deception and Intention Detection. In H. Chen, Th. Raghu, R. Ramesh, A. Vinze, and D. Zeng, editors, *National Security*, volume 2 of *Handbook in Information Systems*, pages 187–208. Elsevier, 2007.

[30] Burke, David A. Towards a Game Theory Model of Information Warfare. Msc Thesis, Faculty of the Graduate School of Engineering and Management of the Air Force Institute of Technology, Air University, November 1999.

[31] Burszstein, Elie and Jean Goubault-Larrecq. A Logical Framework for Evaluating Network Resilience Against Faults and Attacks. In Iliano Cervesato, editor, *Proc. of the 12th ASIAN Computing Science Conference*, pages 212–227, Doha, Qatar, December 9-11 2007. Springer-Verlag Berlin Heidelberg.

[32] Bursztein, Elie. NetQi: A Model Checker for Anticipation Game. In Sungdeok Cha, Jin-Young Choi, Moonzoo Kim, Insup Lee, and Mahesh Viswanathan, editors, *Automated Technology for Verification and Analysis*, volume 5311 of *Lecture Notes in Computer Science*, pages 246–251. Springer Berlin / Heidelberg, 2008. $doi : 10.1007978 - 3 - 540 - 88387 - 6\_22$.

[33] Camerer, Colin F. *Behavioral Game Theory.* Princeton University Press, 2003.

[34] Cappelli, Dawn, Andrew Moore, Randall Trzeciak, and Timothy J. Shimeall. Common Sense Guide to Prevention and Detection of Insider Threats. Technical Report, CERT, Software Engineering Institute, Carnegie Mellon University, January 2009. 3rd Edition - Version 3.1.

[35] Cavusoglu, Huseyin and Srinivasan Raghunathan. Configuration of Intrusion Detection Systems: A Comparison of Decision and Game Theoretic Approaches. *Decision Analysis*, 1(3):131–148, September 2004. Addison Wesley, doi: 10.1287/*deca*.1040.0022.

[36] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. Anomaly Detection: A Survey. *ACM Computing Surveys*, 41(3):15:1–15:58, July 2009. doi: 10.1145/1541880.1541882.

[37] Charness, Gary and David I. Levine. Intention and Stochastic Outcomes: An Experimental Study. *The Economic Journal*, 117(522):1051•1072, July 2007. doi: 10.1111/j.1468-0297.2007.02066.x.

[38] Cho, Sung-Bae. Incorporating Soft Computing Techniques into a Probabilistic Intrusion Detection System. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 32(2):154–160, May 2002.

[39] Clemen, Robert T. and Robert L. Winkler. Combining Probability Distributions from Experts in Risk Analysis. *Risk Analysis*, 19(2), 1999.

[40] CompTIA Survey Reveals Human Error Most Likely Cause of IT Security Breaches. Computing Technology Industry Association (CompTIA). Technical report, March 18, 2003.

[41] Dalvi, Nilesh, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial Classification. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 99–108, New York, NY, USA, 2004. ACM.

[42] Damme, E. van. *Game Theory: the Next Stage.* in Economics beyond the Millennium. Oxford University Press, 1999.

[43] Daskalakis, Constantinos. *The Complexity of Nash Equilibria.* PhD thesis, University of California, Berkeley, Fall 2008.

[44] Daskalakis, Constantinos, Paul W. Goldberg, and Christos H. Papadimitriou. The Complexity of Computing a Nash Equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.

[45] Debar, Hervé. An Introduction to Intrusion-Detection Systems. In *Proc. of Connect'2000*, Doha, Qatar, April 29th - May 1st 2000.

[46] Debar, Hervé, Monique Becker, and Didier Siboni. A Neural Network Component for an Intrusion Detection System. In *Proc. of the IEEE Computer Society Symposium on Research in Security and Privacy*, pages 240–250, May 1992. doi:10.1109/RISP.1992.213257.

[47] Debar, Hervé, Marc Dacier, and Andreas Wespi. A Revised Taxonomy for Intrusion-Detection Systems. *Annals of Telecommunications*, 55(7):361–378, 2000. doi:10.1007/BF02994844.

[48] Debar, Hervé and Benjamin Morin. Evaluation of the Diagnostic Capabilities of Commercial Intrusion Detection Systems. In Andreas Wespi, Giovanni Vigna, and Luca Deri, editors, *Recent Advances in Intrusion Detection*, volume 2516 of *Lecture Notes in Computer Science*, pages 177–198. Springer Berlin / Heidelberg, 2002. doi: $10.1007/3-540-36084-0\_10$.

[49] Denault, Michel, Dimitris Gritzalis, Dimitris Karagiannis, and Paul Spirakis. Intrusion Detection: Approach and Performance Issues of the SECURENET System. *Computers & Security*, 13(6):495 – 508, 1994.

[50] Denning, Dorothy E. An Intrusion-Detection Model. *IEEE Transactions on Software Engineering*, 13(2):222–232, 1987.

[51] Dixit, Avinash and Susan Skeath. *Games of Strategy*. W. W. Norton & Company, Inc., 1999.

[52] Dowell, Cheri and Paul Ramstedt. The ComputerWatch Data Reduction Tool. In *Proc. of the 13th National Computer Security Conference*, pages 99–108, Washington DC, USA, October 1990. National Institute of Standards and Technology/National Computer Security Center.

[53] Durst, Robert, Terrence Champion, Brian Witten, Eric Miller, and Luigi Spagnuolo. Testing and Evaluating Computer Intrusion Detection Systems. *Communications of the ACM*, 42(7):53–61, July 1999.

[54] E-Crime, Watch Survey. CSO Magazine, CERT Program, Microsoft Copr., 2007. accessed 14/4/2011, www.cert.org/archive/pdf/ecrimesummary07.pdf.

[55] Erev, Ido and Alvin E. Roth. Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique Mixed Strategy Equilibria. *The American Economic Review*, 88(4):848–881, September 1998.

[56] Fallah, Mehran. A Puzzle-Based Defense Strategy Against Flooding Attacks Using Game Theory. *Dependable and Secure Computing, IEEE Transactions on*, 7(1):5–19, Jan - March 2010.

[57] Flower, Jean and John Howse. Generating Euler Diagrams. In *Proc. of Diagrams 2002*, pages 61–75. Springer Verlag, 2002.

[58] Frank, Robert H. *Passions within Reason: The Strategic Role of Emotions*. W.W. Norton, 1988.

[59] Gill, Rupinder, Jason Smith, and Andrew Clark. Specifcation-based Intrusion Detection in WLANs. In *Proc. of the 22nd Annual Computer Security Applications Conference (ACSAC'06)*, pages 141–152, Washington, DC, USA, 2006. IEEE Computer Society.

[60] Gintis, Herbert. *Game Theory Evolving - A Problem-Centered Introduction to Modeling Strategic Interaction*. Princeton University Press, 2000.

[61] Goeree, Jacob K., Charles A. Holt, and Thomas R. Palfrey. Quantal Response Equilibrium. In Steven N. Durlauf and Lawrence E. Blume (Eds.), editors, *The New Palgrave: A Dictionary of Economics*. Palgrave Macmillan, 2008.

[62] Habra, Naji, Baudouin Le Charlier, Abdelaziz Mounji, and Isabelle Mathieu. ASAX: Software Architecture and Rule-Based Language for Universal Audit Trail Analysis. In Yves Deswarte et. al., editor, *Computer Security - Proc. of ESORICS*, volume 648 of *LNCS*, pages 435–450. Springer Verlag, Toulouse, France, November 23-25, 1992.

[63] Haile, Philip A., Ali Hortaçsu, and Grigory Kosenok. On the Empirical Content of Quantal Response Equilibrium. *The American Economic Review*, 98(1):180–200, March 2008.

[64] Halpern, Joseph Y. A Computer Scientist Looks at Game Theory. *Games and Economic Behavior*, 45(1):114 – 131, 2003.

[65] Halpern, Joseph Y. Beyond Nash Equilibrium: Solution Concepts for the 21st Century. In *Proc. of the 27th ACM Symposium on Principles of Distributed Computing*, PODC '08, pages 1–10, New York, NY, USA, 2008. ACM.

[66] Halpern, Joseph Y. Computer Science and Game Theory: A Brief Survey. In Steven N. Durlauf and Lawrence E. Blume, editors, *The New Palgrave: A Dictionary of Economics*. Palgrave Macmillan, 2008.

[67] Hamilton, Samuel N., Wendy L. Miller, Allen Ott, and O. Sami Saydjari. Challenges in Applying Game Theory to the Domain of Information Warfare. In *Proc. of the 4th Information Survivability Workshop (ISW-2001/2002)*, Vancouver, BC, Canada, March 2002.

[68] Hamilton, Samuel N., Wendy L. Miller, Allen Ott, and O. Sami Saydjari. The Role of Game Theory in Information Warfare. In *Proc. of the 4th Information Survivability Workshop (ISW-2001/2002)*, Vancouver, BC, Canada, March 2002.

[69] Harsanyi, John C. and Reinhard Selten. *A General Theory of Equilibrium Selection in Games*. Massachusetts Institute of Technology, 1988.

[70] Hart, Sergiu. An Interview with Robert Aumann. *Macroeconomic Dynamics*, 9(5):683–740, January 2005. doi:10.1017.S1365100505050078.

[71] Harvard Business School. Citation Guide: 2009-10 Academic Year. Online, October 2009. http://www.library.hbs.edu/guides/citationguide.pdf, accessed June 2010.

[72] Heberlein L. Todd, Gihan V. Dias, Karl N. Levitt, Biswanath Mukherjee, Jeff Wood, and David Wolber. A Network Security Monitor. In *Proc. of the IEEE Computer Society Symposium on Research in Security and Privacy*, pages 296–304, Los Alamitos, CA, USA, 1990. IEEE Computer Society. doi:10.1109/RISP.1990.63859.

[73] Ho, Yu-Chi, Qian-Chuan Zhao, and David L. Pepyne. The No Free Lunch Theorems: Complexity and Security. *IEEE Transactions on Automatic Control*, 48(5):783–793, May 2003.

[74] Holt, Charles A. and Alvin E. Roth. The Nash Equilibrium: A Perspective. *Proceedings of the National Academy of Sciences (PNAS) of the United States of America*, 101(12):3999–4002, 23 March 2004.

[75] Hubaux, Jean-Pierre. Designining Network Security and Privacy Mechanisms: How Game Theory Can Help. PowerPoint presentation to GameSec 2010 (Conference on Decision and Game Theory for Security), 22-23 November, Berlin, Germany 2010. accessed 27/12/2010, http://www.gamesec-conf.org/GameSecKeynoteFinal.pptx.

[76] Ilgun, Koral, Richard A. Kemmerer, and Phillip A. Porras. State Transition Analysis: A Rule-Based Intrusion Detection Approach. *IEEE Transactions on Software Engineering*, 21:181–199, 1995.

[77] Jackson, Kathleen, David DuBois, and Cathy Stallings. An Expert System Application for Network Intrusion Detection. In *Proc. of the 14th National Computer Security Conference*, pages 215–225, 1991.

[78] Jackson, Kathleen A. Intrusion Detection System (IDS) Product Survey. Research Report LA-UR-99-3883, Los Alamos National Laboratory, June 1999. Version 2.1.

[79] Jagannathan R., Teresa Lunt, Debra Anderson, Chris Dodd, Fred Gilham, Caveh Jalali, Hal Javitz, Peter Neumann, Ann Tamaru, and Alfonso Valdes. Next-Generation Intrusion Detection Expert System (NIDES). System Design Document: A007, A008, A009, A011, A012, A014, SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025, March 9, 1993.

[80] Johnson, Benjamin, Jens Grossklags, Nicolas Christin, and John Chuang. Are Security Experts Useful? Bayesian Nash Equilibria for Network Security Games with Limited Information. In Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou, editors, *Computer Security • ESORICS 2010*, volume 6345 of *Lecture Notes in Computer Science*, pages 588–606. Springer Berlin / Heidelberg, 2010. doi: $10.1007/978-3-642-15497-3\_36$.

[81] Jones, Anita K. and Robert S. Sielken. Computer System Intrusion Detection: A Survey. Technical Report, Computer Science Department, University of Virginia, 2000.

[82] Kabiri, Peyman and Ali A. Ghorbani. Research on Intrusion Detection and Response: A Survey. *International Journal of Network Security*, 1(2):84–102, September 2005.

[83] Kalai, Ehud. Games, Computers, and O.R. In *Proc. of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '96, pages 468–473, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics.

[84] Kantzavelou, Ioanna. An attack detection system for secure computer systems. Msc thesis, National University of Ireland, Computer Networks & Distributed Systems Re-

search Group, Department of Computer Science, University College Dublin, Belfield, Dublin 4, April 1994.

[85] Kantzavelou, Ioanna and Sokratis Katsikas. An Attack Detection System for Secure Computer Systems - Outline of the Solution. In L. Yngstrom and J. Carlsen, editors, *Information Security in Research and Business*, pages 123–135, Copenhagen, Denmark, May 1997. Chapman and Hall.

[86] Kantzavelou, Ioanna and Sokratis Katsikas. A Generic Intrusion Detection Game Model in IT Security. In *Proc. of the 5th International Conference on Trust, Privacy and Security in Digital Business (TrustBus '08)*, pages 151–162, Turin, Italy, September 2008. Springer-Verlag.

[87] Kantzavelou, Ioanna and Sokratis Katsikas. Playing Games with Internal Attackers Repeatedly. In *Systems, Signals and Image Processing, 2009. IWSSIP 2009. 16th International Conference on*, pages 1–6, Chalkida, Greece, 18-20 2009. IEEE. doi: 10.1109/IWSSIP.2009.5367708, (invited).

[88] Kantzavelou, Ioanna and Sokratis Katsikas. A Game-based Intrusion Detection Mechanism to Confront Internal Attackers. *Computers & Security*, 29(8):859 – 874, 2010.

[89] Kantzavelou, Ioanna and Ahmed Patel. An Attack Detection System for Secure Computer Systems - Desing of the ADS. In S. Katsikas and D. Gritzalis, editors, *Information Systems Security: Facing the Information Society of the 21st Century*, pages 337–347, Samos, Greece, May 1996. Chapman & Hall.

[90] Ko, Calvin, George Fink, and Karl Levitt. Automated Detection of Vulnerabilities in Privileged Programs by Execution Monitoring. In *Proc of the 10th Annual Computer Security Applications Conference*, volume XIII, pages 134–144, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.

[91] Ko, Calvin, Manfred Ruschitzka, and Karl Levitt. Execution Monitoring of Security-critical Programs in Distributed Systems: a Specification-based Approach. In *Proc. of the IEEE Symposium on Security and Privacy, S&P'97*, pages 175–187, Los Alamitos, CA, USA, 1997. IEEE Computer Society.

[92] Kodialam, Murali and T. V. Lakshman. Detecting Network Intrusions via Sampling: A Game Theoretic Approach. In *Proc. of the IEEE INFOCOM 2003*, San Fransisco, March 2003.

[93] Konorski, Jerzy. A Game-Theoretic Study of CSMA/CA Under a Backoff Attack. *IEEE/ACM Transactions on Networking*, 14(6):1167–1178, December 2006. IEEE Press.

[94] Koutsoupias, Elias and Christos Papadimitriou. Worst-Case Equilibria. In *Proc. of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413, Trier, Germany, 4–6 March 1999.

[95] Kreps, David. *Game Theory and Economic Modelling.* Oxford University Press, 1990.

[96] Kruegel, Christopher, Fredrik Valeur, and Giovanni Vigna. *Intrusion Detection and Correlation - Challenges and Solutions.* Advances in Information Security. Springer, 2005.

[97] Kumar, Sandeep and Eugene H. Spafford. A Pattern Matching Model for Misuse Intrusion Detection. In *Proc. of the 17th National Computer Security Conference*, pages 11–21, 1994.

[98] Lazarevic, Aleksandar, Vipin Kumar, and Jaideep Srivastava. Chapter 2, Intrusion Detection: A Survey. In Vipin Kumar, Jaideep Srivastava, and Aleksandar Lazarevic, editors, *Managing Cyber Threats: Issues, Approaches and Challenges*, volume 5 of *Massive Computing*, pages 19–78. Springer, 2005.

[99] Lee, Wenke and Salvatore J. Stolfo. Data Mining Approaches for Intrusion Detection. In *Proceedings of the 7th conference on USENIX Security Symposium - Volume 7*, pages 6–6, Berkeley, CA, USA, January 26-29 1998. USENIX Association.

[100] Levine, David K. Modeling Altruism and Spitefulness in Experiments. *Review of Economic Dynamics*, 1(3):593 – 622, July 1998.

[101] Levine, David K. Game theory. In *Encyclopedia of Cognitive Science*. Nature Publishing Group: Basingstoke, 2001.

[102] Levine, David K. Repeated Games Step-by-Step, May 2002. accessed 1/4/2011, http://www.dklevine.com/econ101/repeated-step.pdf.

[103] L'Huillier, Gaston, Richard Weber, and Nicolas Figueroa. Online Phishing Classification Using Adversarial Data Mining and Signaling Games. *SIGKDD Explor. Newsl.*, 11(2):92–99, December 2009. ACM.

[104] Linial, Nathan. Chapter 38 Game-Theoretic Aspects of Computing. In Robert Aumann and Sergiu Hart, editors, *Handbook of Game Theory with Economic Applications*, volume II, pages 1339–1395. Elsevier, North-Holland, Amsterdam, 1994.

[105] Lin,L., X. Wang, and S. Jajodia. Abstraction-Based Misuse Detection: High-Level Specifications and Adaptable Strategies. In *Proc. of the 11th IEEE Computer Security Foundations Workshop*, pages 190–202, Los Alamitos, CA, USA, 1998. IEEE Computer Society.

[106] Liu, Debin, XiaoFeng Wang, and Jean Camp. Game-theoretic Modeling and Analysis of Insider Threats. *International Journal of Critical Infrastructure Protection I*, 1(1):75–80, December 2008.

[107] Liu, Peng and Wanyu Zang. Incentive-Based Modeling and Inference of Attacker Intent, Objectives, and Strategies. In *Proc. of the 10th ACM Computer and Commu-*

nications Security Conference (CCS 2003), pages 179–189, Washington, DC, USA, October 2003. ACM.

[108] Liu, Peng, Wanyu Zang, and Meng Yu. Incentive-Based Modeling and Inference of Attacker Intent, Objectives, and Strategies. *ACM Transactions on Information and System Security*, 8(1):78–118, February 2005.

[109] Liu, Yu, Cristina Comaniciu, and Hong Man. Modelling Misbehaviour in Ad Hoc Networks: A Game Theoretic Approach for Intrusion Detection. *International Journal of Security and Networks*, 1(3/4):243–254, 2006.

[110] Lunt, Teresa F. A Survey of Intrusion Detection Techniques. *Computers & Security*, 12(4):405 – 418, 1993.

[111] Lunt, Teresa F., R. Jagannathan, Rosanna Lee, Sherry Listgarten, David L. Edwards, Peter G. Neumann, Harold S. Javitz, and Al Valdes. IDES: The Enhanced Prototype - A Real-Time Intrusion-Detection Expert System. Technical Report SRI-CSL-88-12, Computer Science Laboratory, SRI International, 333 Ravenswood Avenue, Menlo Park, CA, USA, October 1988.

[112] Lye, Kong-wei and Jeannette Wing. Game Strategies in Network Security. In *Proceedings of the Foundations of Computer Security Workshop*, Copenhagen, Denmark, July 2002.

[113] Lye, Kong-wei and Jeannette Wing. Game Strategies in Network Security. *International Journal of Information Security*, 4:71–86, February 2005. Springer, Special Issue of selected papers from FCS/VERIFY2002, doi: 10.1007/s10207-004-0053-9.

[114] Machado, Renita and Sirin Tekinay. A survey of game-theoretic approaches in wireless sensor networks. *Computer Networks*, 52(16):3047–3061, November 2008.

[115] Mailath, George J. and Larry Samuelson. *Repeated Games and Reputations*. Oxford University Press, 2006.

[116] Manshaei, Mohammad Hossein, Quanyan Zhu, Tansu Alpcan, Tamer Başar, Mohammad Hossein, and Jean-Pierre Hubaux. Game Theory Meets Network Security and Privacy. EPFL-REPORT-151965, Ecole Polytechnique Fédérale de Lausanne (EPFL), September 2010.

[117] McCabe, Kevin A., Vernon L. Smith, and Michael LePore. Intentionality Detection and "Mindreading": Why Does Game Form Matter. *Proceedings of the National Academy of Sciences (PNAS) of the United States of America*, 97(8):4404–4409, 11 April 2000.

[118] McHugh, John. Intrusion and Intrusion Detection. *International Journal of Information Security*, 1(1):14–35, August 2001. Springer Berlin / Heidelberg, doi:10.1007/s102070100001.

[119] McKelvey, Richard D., Andrew M. McLennan, and Theodore L. Turocy. Gambit: Software tools for game theory, version 0.2007.01.30, January 2007. accessed 20/1/2010, http://www.gambit-project.org.

[120] Mckelvey, Richard D. and Thomas R. Palfrey. Quantal Response Equilibria for Normal Form Games. *Games and Economic Behavior*, 10(1):6–38, July 1995.

[121] Mckelvey, Richard D. and Thomas R. Palfrey. Quantal Response Equilibria for Extensive Form Games. *Experimental Economics*, 1(1):9–41, June 1998.

[122] McMillan, John. Selling Spectrum Rights. *Journal of Economics Perspectives*, 8:145–162, 1994.

[123] Michiardi, Pietro and Refik Molva. CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile ad hoc Networks. In Borka Jerman-Blazic

and Tomaz Klobucar, editors, *Proceedings of the IFIF TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security*, pages 107–121, Portoroz, Slovenia, September 26-27 2002. Kluwer, B.V.

[124] Milgrom, Paul. *Putting Auction Theory to Work*. Cambridge University Press, 2004.

[125] Miller, Charles D., Vern E. Heeren, and John Hornsby. *Mathematical Ideas*. Pearson Addison-Wesley, 2007.

[126] Mohammed, Noman, Hadi Otrok, Lingyu Wang, Mourad Debbabi, and Prabir Bhattacharya. Mechanism Design-Based Secure Leader Election Model for Intrusion Detection in MANET. *IEEE Transactions on Dependable and Secure Computing*, 8(1):89–103, January-February 2011. doi:10.1109/TDSC.2009.22.

[127] Nasar, Sylvia. *A Beautiful Mind*. Faber and Faber, England, 1998.

[128] Nash, John F. Equilibrium Points in n-Person Games. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 36, no. 1, pages 48–49, January 15, 1950.

[129] Ning, Peng and Sushil Jajodia. Intrusion Detection Techniques. In H. Bidgoli, editor, *The Internet Encyclopedia*. John Wiley & Sons, December 2003.

[130] Ning, Peng, Sushil Jajodia, and Sean Wang. *Intrusion Detection in Distributed Systems: An Abstraction-Based Approach*. Advances in Information Security. Kluwer Academic Publishers, 2004.

[131] Ning, Peng, Sushil Jajodia, and Xiaoyang Sean Wang. Abstraction-Based Intrusion Detection in Distributed Environments. *ACM Transactions on Information and System Security*, 4(4):407–452, November 2001.

[132] Nisan, Noam, Tim Roughgarden, Eva Tardos, and Vijay Vazirani (Eds.). *Algorithmic Game Theory*. Cambridge University Press, 2007.

[133] Nobelprize.org. The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 1994. accessed 25 Nov 2010, http://nobelprize.org/nobel_prizes/economics-/laureates/1994/index.html.

[134] Nobelprize.org. The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2005. accessed 25 Nov 2010, http://nobelprize.org/nobel_prizes/economics-/laureates/2005/index.html.

[135] Osborn, Martin J. and Ariel Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

[136] Osborne, Martin J. *An Introduction to Game Theory*. Oxford University Press, New York, 2004.

[137] Otrok, Hadi, Mourad Debbabi, Chadi Assi, and Prabir Bhattacharya. A Cooperative Approach for Analyzing Intrusions in Mobile Ad hoc Networks. In *Proc. of the 27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*, Toronto, Canada, June 25-29 2007. IEEE Press. doi: 10.1109/ICDCSW.2007.91.

[138] Otrok, Hadi, Mona Mehrandish, Chadi Assi, Mourad Debbabi, and Prabir Bhattacharya. Game Theoretic Models for Detecting Network Intrusions. *Computer Communications*, 31(10):1934–1944, June 2008.

[139] Otrok, Hadi, Noman Mohammed, Lingyu Wang, Mourad Debbabi, and Prabir Bhattacharya. A Game-Theoretic Intrusion Detection Model for Mobile Ad Hoc Networks. *Computer Communications*, 31(4):708 – 721, March 2008. Algorithmic and Theoretical Aspects of Wireless ad hoc and Sensor Networks.

[140] Otrok, Hadi, Benwen Zhu, Hamdi Yahyaoui, and Prabir Bhattacharya. An Intrusion Detection Game Theoretical Model. *Information Security Journal: A Global Perspective*, 18(5):199–212, 2009.

[141] Panagiotou, George. Conjoining prescriptive and descriptive approaches: Towards an integrative framework of decision making. *Management Decision*, 46(4):553–564, 2008. Emerald.

[142] Papadimitriou, Christos. Algorithms, Games, and the Internet. In *Proc. of the 33rd Annual ACM Symposium on Theory of Computing*, STOC '01, pages 749–753, New York, NY, USA, 2001. ACM.

[143] Patcha, Animesh and Jung-Min Park. A Game Theoretic Approach to Modeling Intrusion Detection in Mobile Ad Hoc Networks. In *Proc. of the 2004 IEEE Workshop on Information Assurance and Security*, June 2004.

[144] Patcha, Animesh and Jung-Min Park. A Game Theoretic Formulation for Intrusion Detection in Mobile Ad Hoc Networks. *International Journal of Network Security*, 2(2):131–137, March 2006.

[145] Patcha, Animesh and Jung-Min Park. An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends. *Computer Networks*, 51(12):3448 – 3470, 2007.

[146] Paxson, Vern. Bro: A System for Detecting Network Intruders in Real-Time. *Computer Networks*, 31(23-24):2435–2463, December 1999. Elsevier.

[147] Pieprzyk, Josef, Thomas Hardjono, and Jennifer Seberry. *Fundamentals of Computer Security*. Springer-Verlag, 2003.

[148] Porras, Phil, Dan Schnackenberg, Stuart Staniford-Chen, Maureen Stillman, and Felix Wu. The Common Intrusion Detection Framework Architecture, 1998. accessed 15/10/2010, http://www.isi.edu/gost/cidf/drafts/architecture.txt.

[149] Porras, Phillip A. and Peter G. Neumann. EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In *Proc. of the 20th National Information Systems Security Conference*, pages 353–365, October 7-10, 1997.

[150] Prevelakis, Vassilis and Diomidis Spinellis. The Athens Affair. *IEEE Spectrum*, 44(7):26•33, July 2007.

[151] Randazzo, Marisa Reddy, Michelle Keeney, Eileen Kowalski, Dawn Cappelli, and Andrew Moore. Insider Threat Study: Illicit Cyber Activity in the Banking Sector. Technical Report CMU/SEI-2004-TR-021, June 2005.

[152] Riggs, Cliff. *Network Perimeter Security - Building Defense In-Depth.* Auerbach Publications, 2004.

[153] Roughgarden, Tim. Algorithmic Game Theory. *Communications of the ACM*, 53(7):78–86, July 2010.

[154] Roughgarden, Tim. Computing Equilibria: A Computational Complexity Perspective. *Economic Theory*, 42(1):193–236, 2010. doi:10.1007/s00199-009-0448-y.

[155] Roy, Sankardas, Charles Ellis, Sajjan Shiva, Dipankar Dasgupta, Vivek Shandilya, and Qishi Wu. A Survey of Game Theory as Applied to Network Security. In *Proc. of the 43rd Hawaii International Conference on System Sciences*, pages 1–10, Los Alamitos, CA, USA, 2010. IEEE Computer Society.

[156] Salem, Malek Ben, Shlomo Hershkop, and Salvatore J. Stolfo. A Survey of Insider Attack Detection Research. In Stolfo S. J., Bellovin S. M., Hershkop S., Keromytis A.,

Sinclair S., and Smith S. W., editors, *in Insider Attack and Cyber security - Beyond the Hacker*. Springer, June 2008.

[157] Sallhammar, Karin, Svein Johan Knapskog, and Bjarne E. Helvik. Using Stochastic Game Theory to Compute the Expected Behavior of Attackers. In *SAINT-W '05: Proceedings of the 2005 Symposium on Applications and the Internet Workshops*, pages 102–105, Washington, DC, USA, 2005. IEEE Computer Society.

[158] SANS. Top-20 2007 Internet Security Problems, Threats, and Risks. Technical report, SANS Institute, June 2005.

[159] Schultz, Eugene and Russell Shumway. *Incident Response: A Strategic Guide to Handling System and Network Security Breaches*. New Riders Publishing, 2002.

[160] Sebring, Michael M., Eric Shellhouse, Mary E. Hanna, and R. Alan Whitehurst. Expert Systems in Intrusion Detection: A Case Study. In *Proc. of the 11th National Computer Security Conference*, pages 74–81, Baltimore, Maryland, October 1988. National Institute of Standards and Technology/National Computer Security Center.

[161] Seleznyov, Alexandr and Seppo Puuronen. Anomaly Intrusion Detection Systems: Handling Temporal Relations between Events. In *Proc. of the 2nd International Workshop on Recent Advances in Intrusion Detection (RAID 1999)*, West Lafayette, Indiana, USA, September 7-9 1999.

[162] Selten, Reinhard and Thorsten Chmura. Stationary Concepts for Experimental 2x2-Games. *The American Economic Review*, 98(3):938–966, June 2008.

[163] Shin, Sun-Joo and Oliver Lemon. Diagrams. The Stanford Encyclopedia of Philosophy (Winter 2008 Edition), Edward N. Zalta (ed.), 2008. accessed 10/11/2010, http://plato.stanford.edu/archives/win2008/entries/diagrams/.

[164] Shiva, Sajjan, Sankardas Roy, and Dipankar Dasgupta. Game Theory for Cyber Security. In *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, CSIIRW '10, pages 34:1–34:4, New York, NY, USA, 2010. ACM.

[165] Shoham, Yoav. Computer Science and Game Theory. *Communications of the ACM*, 51(8):75–79, Auhust 2008.

[166] Shoham, Yoav and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*. Cambridge University Press, 2009.

[167] Skyrms, Brian and Peter Vanderschraaf. Game theory. In M. Gabbay, D. and P. Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, pages 391–439. Kluwer Academic Publishers, 1998.

[168] Smaha, Stephen E. Haystack: An Intrusion Detection System. In *Proc. of the 4th Aerospace Computer Security Applications Conference*, pages 37–44, October 1988.

[169] Snapp, Steven R., Stephen E. Smaha, Daniel M. Teal, and Tim Grance. The DIDS (Distributed Intrusion Detection System) Prototype. In *Proc. of the USENIX Technical Conference*, pages 227–233, San Antonio, Texas, June 8-12, 1992. USENIX Association.

[170] Spirakis, Paul, Sokratis K. Katsikas, Dimitris A. Gritzalis, F. Allegre, D. Androutsopoulos, John Darzentas, C. Gigante, D. Karagiannis, P. Kess, H. Putkonen, and Thomas Spyrou. SECURENET: A Network-oriented Intelligent Intrusion Prevention and Detection System. In *Proc. of the 10th International Information Security Conference (IFIP SEC '94)*, Curacao, Dutch Caribbean, 1994.

[171] Spyrou, Thomas and John Darzentas. Intention modelling: Approximating computer user intentions for detection and prediction of intrusions. In S. Katsikas and

D. Gritzalis, editors, *Proc. of the 12th International Information Security Conference (IFIP SEC '96)*, pages 319–336, Samos, Greece, 1996.

[172] Staniford-Chen, S., S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagl, K. Levitt, C. Wee, R. Yip, and D. Zerkle. GrIDS - A Graph Based Intrusion Detection System for Large Networks. In *Proc. of the 19th National Information Systems Security Conference*, pages 361–370, 1996.

[173] Toosi, Adel Nadjaran and Mohsen Kahani. A New Approach to Intrusion Detection Based on an Evolutionary Soft Computing Model Using Neuro-Fuzzy Classifiers. *Computer Communications*, 30(10):2201–2212, July 2007.

[174] Turocy, Theodore L. A Dynamic Homotopy Interpretation of Quantal Response Equilibrium Correspondences. *Games and Economic Behavior*, 51:243–263, 2005.

[175] Uppuluri, Prem and R. Sekar. Experiences with Specification-based Intrusion Detection. In *Proc. of the 4th International Symposium on Recent Advances in Intrusion Detection, RAID '00*, pages 172–189. Springer, 2001.

[176] Vaccaro, M.S. and G. E. Liepins. Detection of Anomalous Computer Session Activity. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 280–289, Los Alamitos, CA, USA, 1989. IEEE Computer Society.

[177] Verwoerd, Theuns and Ray Hunt. Intrusion Detection Techniques and Approaches. *Computer Communications*, 25(15):1356 – 1365, 2002.

[178] Vigna, Giovanni and Richard A. Kemmerer. Netstat: A Network-based Intrusion Detection System. *Journal of Computer Security*, 7(1):37–71, January 1999.

[179] Vigna, Giovanni and Christopher Kruegel. Host-based Intrusion Detection Systems. In H. Bigdoli, editor, *Handbook of Information Security*, volume III. John Wiley & Sons, December 2005.

[180] Von Neumann, John and Oscar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1953.

[181] Watson, Joel. *Strategy: An Introduction to Game Theory*. W. W. Norton & Company, 2002.

[182] Winkler, Robert L. Combining Probability Distributions from Dependent Information Sources. *Management Science*, 27(4):479–488, April 1981.

Appendix A

Table 9.1 in the sequel refers to the detailed information derived from the calculations of the QRE algorithm. Only the significant steps of these calculations have been included, because the total number of steps is 191.

Quantal Response Equilibria (QRE) Calculations													
Step	λ	**Insider - 1^{st} Info Set**				**Insider - 2^{nd} Info Set**				**IDS Info Set**			
		Normal	Mistake	Pre-attack	Attack	Normal	Mistake	Pre-attack	Attack	Continue	Recommend	Warning	Stop
1	0.000	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250
2	0.004	0.250	0.248	0.247	0.255	0.250	0.248	0.250	0.252	0.248	0.252	0.250	0.250
3	0.009	0.250	0.246	0.243	0.261	0.250	0.245	0.251	0.254	0.245	0.254	0.250	0.251
4	0.015	0.251	0.243	0.239	0.267	0.251	0.243	0.251	0.256	0.243	0.256	0.250	0.251
5	0.020	0.251	0.240	0.235	0.274	0.251	0.240	0.251	0.258	0.240	0.258	0.251	0.252
6	0.027	0.251	0.237	0.231	0.281	0.252	0.236	0.252	0.260	0.236	0.260	0.251	0.252
7	0.034	0.250	0.234	0.226	0.290	0.252	0.232	0.252	0.263	0.233	0.263	0.251	0.253
8	0.041	0.250	0.230	0.221	0.299	0.253	0.228	0.253	0.266	0.228	0.266	0.252	0.254
9	0.049	0.249	0.225	0.216	0.309	0.254	0.223	0.254	0.269	0.223	0.269	0.253	0.255
10	0.058	0.248	0.221	0.211	0.320	0.255	0.217	0.255	0.272	0.218	0.272	0.253	0.257
...	...	...	...	...	...	...	...	...	...	...	...	...	...
57	3.317	0.035	0.000	0.000	0.965	0.273	0.000	0.384	0.343	0.000	0.000	0.000	1.000
58	3.589	0.027	0.000	0.000	0.973	0.282	0.000	0.376	0.342	0.000	0.000	0.000	1.000
59	3.892	0.020	0.000	0.000	0.980	0.292	0.000	0.368	0.340	0.000	0.000	0.000	1.000
...	...	...	...	...	...	...	...	...	...	...	...	...	...
66	7.269	0.001	0.000	0.000	0.999	0.331	0.000	0.336	0.334	0.000	0.000	0.000	1.000
67	7.985	0.000	0.000	0.000	1.000	0.332	0.000	0.335	0.334	0.000	0.000	0.000	1.000
68	8.775	0.000	0.000	0.000	1.000	0.333	0.000	0.334	0.333	0.000	0.000	0.000	1.000
69	9.645	0.000	0.000	0.000	1.000	0.333	0.000	0.334	0.333	0.000	0.000	0.000	1.000
...	...	...	...	...	...	...	...	...	...	...	...	...	...
189	888596.414	0.000	0.000	0.000	1.000	0.333	0.000	0.333	0.333	0.000	0.000	0.000	1.000
190	977456.050	0.000	0.000	0.000	1.000	0.333	0.000	0.333	0.333	0.000	0.000	0.000	1.000
191	1075201.649	0.000	0.000	0.000	1.000	0.333	0.000	0.333	0.333	0.000	0.000	0.000	1.000

Table 9.1: Step-by-step calculations of QRE

Appendix B

We present in pseudo-code the main parts of the Game-based Detection Algorithm, illustrated in Figure 8.3 in Section 8.2.2. We have chosen this flexible way, to describe it in lower level and supplement its picture.

Algorithm 1 The Game-based Detection Algorithm.

Require: This algorithm is triggered by a new event, *event(i)*, stored in *Filtered Data*.

Ensure: At the end of this algorithm, a counteraction will be activated if necessary.

1: *intrusion_alert* $\leftarrow 0$
2: Get *event(i)* from the *Filtered Data* store
3: *intrusion_alert* $\leftarrow$ **Intrusion_Detection_Engine**(*event*(i))
4: **if** *intrusion_alert* $== 0$ **then**
5: **print** "INTRUSION ALARM is OFF"
6: **else**
7: **print** "INTRUSION ALARM is ON"
8: **return** **Counteraction**(*IDS_result*)
9: Send counteraction information to SO
10: **end if**
11: Send event information to SO
12: $configuration\_assessement \leftarrow$ **Check_Configuration**($Detection\_Parameters$)
13: **if** *configuration_assessement* == 'YES' **then**
14: **return** **Configuration**(*Detection_Parameters*)
15: **end if**
16: *game*(u) $\leftarrow$ **Game_Construction**(*Game_Data*)
17: *action_probabilities*(u) $\leftarrow$ **QRE_Calculations**(*game*(u))
18: **if** *QRE_alert* $== 0$ **then**
19: **print** "QRE ALARM is OFF"
20: **else**
21: **print** "QRE ALARM is ON"
22: **return** **Counteraction**(*Prevent*)
23: Send counteraction information to SO
24: **end if**
25: Send *action_probabilities(u)* to SO

Algorithm 2 The **Game_Construction** Algorithm.

Require: *Game_Data, u, event(i).*

Ensure: An existing game or a new game (*game*).

1: **if** Check_Game_Existence(*Game_Data, event(i)*) == 0 **then**
2: $game \leftarrow Create\_New\_Game(event(i))$
3: **else**
4: Get *Game_Data(u)* from the *Stored Games* store
5: $game \leftarrow Update(Game\_Data(u))$
6: **end if**
7: **return** *game*

Algorithm 3 The **QRE_Calculations** Algorithm.

Require: *game.*

Ensure: QRE action probabilities.

1: Calculate the QRE for the current time period
2: **print** "QRE probabilities for the current time period"
3: Calculate the QRE for the next time period
4: **print** "QRE probabilities for the next time period"

Algorithm 4 The **Combining_Probabilities** Algorithm.

Require: IDS result and QRE action probabilities.

Ensure: Final decision result.

1: Calculate the average between the IDS result and the QRE action probabilities
2: **print** "Final Decision result"
3: Send *Final decision result* to SO
4: $Counteraction() \leftarrow$ *Final decision result*