

Chapter 2

AN INTRODUCTION TO INTRUSION DETECTION

1. Intrusion Prevention

Several methods are available to protect a computer system or network from attack. A good introduction to such methods is [HB95], from which this section borrows heavily. The paper lists six general, non-exclusive approaches to anti-intrusion techniques: pre-emption, prevention, deterrence, detection, deflection, and countermeasures (see Figure 2.1):

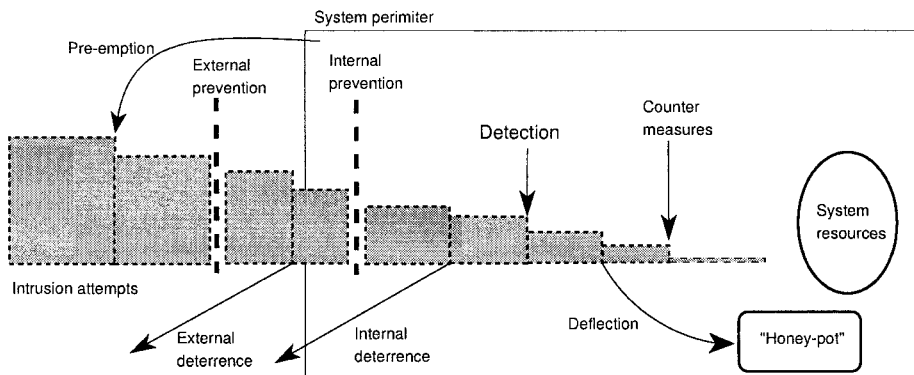


Figure 2.1. Anti-intrusion techniques (from [HB95])

- 1 *Pre-emption* To strike against the threat before it has had a chance to mount its attack, in the spirit of: “Do unto others, before they do unto you.” In a civilian setting, this is a dangerous and possibly unlawful approach, where innocent—and indeed not so innocent—bystanders may be harmed.

- 2 *Prevention* To preclude or severely limit the likelihood of a particular intrusion succeeding. One can, for example, elect to not be connected to the Internet if one is afraid of being attacked by that route, or choose to be connected via some restriction mechanism such as a firewall. Proving your software free of security defects also falls under this heading. Unfortunately, this can be an expensive and awkward approach, since it is easy to throw the baby out with the bath water in the attempt to prevent attacks. Internal prevention comes under the control of the system owner, while external prevention takes place in the environment surrounding the system, such as a larger organization, or society as a whole.
- 3 *Deterrence* To persuade an attacker to hold off his attack, or to break off an ongoing attack. Typically this is accomplished by increasing the perceived risk of negative consequences for the attacker. Of course, if the value of the protected resource is great, the determined attacker may not be scared off so easily. Internal deterrence can take the form of login banners warning potential internal and external attackers of dire consequences should they proceed. External deterrence could be effected by the legal system, with laws against computer crime and the strict enforcement of the same.
- 4 *Detection* To identify intrusion attempts, so that the proper response can be evoked. This most often takes the form of notifying the proper authority. The problems are obvious: the difficulty of defending against a hit-and-run attack, and the problem of false alarms, or failing to sound the alarm when someone surreptitiously gains, or attempts to gain, access.
- 5 *Deflection* To lure an intruder into thinking that he has succeeded when in fact he has been herded away from areas where he could do real damage. The main problem is that of managing to fool an experienced attacker, at least for a sufficient period of time.
- 6 *Countermeasures* To counter actively and autonomously an intrusion while it is in progress. This can be done without the need for detection, since the countermeasure does not have to discriminate—although it is preferable if it can—between a legitimate user who makes a mistake and an intruder who sets off a predetermined response, or “booby trap”.

The reasons for our desire to employ the principle of surveillance are much the same as in the physical security arena: we wish to deploy a defence in depth; we do not believe in the infallibility of the perimeter defence; when someone manages to slip through or even attempts to attack we do not want them to have undetected free reign of the system; for technical reasons we perhaps cannot strengthen our perimeter defences (lack of source code etc.); we wish to defend not only against outsiders, but also against insiders, those that already operate within the perimeter, etc.

2. Intrusion Detection

As the principle of surveillance stems from the application of intrusion detection systems to computer security it is fitting to start with a few definitions and introduction to that area of study. Research in intrusion detection is the study of systems that automatically detect intrusions into computer systems. They are designed to detect computer security violations made by the following important types of attackers:

- Attackers using prepackaged exploit scripts. Primarily outsiders.
- Automated attacks originating from other computers, so called *worms*.
- Attackers operating under the identity of a legitimate user, for example by having stolen that user's authentication information (password). Outsiders and insiders.
- Insiders abusing legitimate privileges, etc.

Giving satisfactory definitions to these terms turns out to be problematic. Although most computer users could easily describe what they do not want to happen with their computers, finding strict definitions of these actions is often surprisingly difficult. Furthermore, many security problems arise between the ordinary every day definitions that we use to communicate security, and the strict definitions that are necessary to research. For example the simple phrase "Alice speaks to Bob on the freshly authenticated channel" is very difficult to interpret in a packet-sending context, and indeed severe security problems have arisen from confusion arising from the application of such simple models such as "speaking" in a computer communications context [Gol00]. That numerous, spectacular mistakes have been made by computer security researchers and professionals only serves to demonstrate the difficulty of the subject.

2.1 Definitions

That said, a definition of what we mean by *intrusion* and other related terms remains essential, at least in the context of *intrusion detection*:

Intrusion The *malicious* violation of a *security policy* (implied or otherwise) by an *unauthorized agent*.

Intrusion detection The automated detection and alarm of any situation where an intrusion has taken, or is about to take place. (The detection must be complemented with an alert to the proper authority if it is to act as a useful security measure.)

We will consider these definitions in greater detail in the following paragraphs:

Malicious. The person who breaks into or otherwise unduly influences our computer system is deemed not have our best interests at heart. This is an interesting point, for in general it is impossible for the intrusion detection system to decide whether the agent of the security violation has malicious intent or not, even after the fact. Thus we may expect the intrusion detection system to raise the alarm whenever there is sufficient evidence of an activity that *could* be motivated by malice. By this definition this will result in a false alarm, but in most cases a benign one, since most people do not mind the alarm being raised about a potentially dangerous situation that has arisen from human error rather than malicious activity.

Security Policy. This stresses that the violations against which we wish to protect are, to a large extent, in the eyes of the owner of the resource being protected (in western law at least). Other legitimate demands on security may in future be made by the state legislature. Some branches of the armed services are already under such obligations, but in the civilian sector few (if any) such demands are currently made. In practice security policies are often weak, however, and in a civilian setting we often do not know what to classify as a violation until after the fact. Thus it is beneficial if our intrusion detection system can operate in circumstances where the security policy is weakly defined, or even non-existent. One way of circumventing this inherent problem is for the supplier of the intrusion detection system to define a *de facto* security policy that contains elements with which she hopes all users of her system will agree. This situation may be compared with the law of the land, only a true subset of which is agreed by most citizens to define *real* criminal acts. It goes without saying that a proper security policy is preferable. This ought to be defined as the set of actions (or rather principles) of operation that are allowed, instead of in the negative for best security.

Unauthorized Agent. The definition is framed to address the threat that comes from an *unauthorized agent*, and should not be interpreted too narrowly. The term singles out all those who are not legitimate owners of the system, i.e., who are not allowed to make decisions that affect the security policy. This does not specifically exclude *insiders* i.e. people who are authorized to use the system to a greater or lesser extent, but not authorized to perform all possible actions. The point of this distinction is that we do not attempt to encompass those violations that would amount to protecting the owner from himself. To accomplish this is, of course, both simple and impossible: simple in the sense that if the owner makes a simple legitimate mistake, a timely warning may make him see his error and take corrective action; impossible, in that if the person who legally commands the system wishes to destroy or otherwise influence the system, there is no way to prevent him, short of taking control of the system away from him,

in which case he no longer “legally commands the system.” When all is said and done, trust has to be placed in an entity, and our only defense against this trust being abused is to use risk management activities external to the intrusion detection system. It is a difficult question as to whether we should consider non-human attackers such as other computers to be agents in themselves, or merely tools acting on the behalf of some other agent. We will not delve more deeply into such questions here.

Automated Detection and Alarm. The research into intrusion detection has almost exclusively considered systems that operate largely without human supervision. An interesting class of systems that has not been studied to any significant degree (the present book excepted) are those that operate with a larger degree of human supervision, placing so much responsibility on the human operator that *she* can be thought of as the detection element proper (or at least a significant part of it). Such systems would support the human in observing and making decisions about the security state of the supervised system; a ‘security camera’ for computer systems. Continued reliance solely on fully automated systems may turn out to be less than optimal.

Delivered to the Proper Authority. It cannot be overemphasized that the alarm must be *delivered* to the *proper authority*—henceforth referred to as the Site Security Officer or SSO—in such a manner that the SSO can take action. The ubiquitous car alarm today arouses little, if any, response from the public, and hence does not act as an effective deterrent to would-be car thieves. Thus the SSO’s response, which may or may not be aided by automatic systems within the intrusion detection system itself, is a crucial component in the fielding of intrusion detection systems. There has been little research, even in the simpler field of automated alarms, into how to present information to the SSO so that she can make the correct decision and take the correct action. It is important that the authority that is expected to take corrective action in the face of computer security violations—keeping in mind that such violations often originate “in house”—really *has* the authority to take the appropriate action. This is not always the case in a civilian setting.

Intrusion has Taken Place. The phrase “any situation where an *intrusion has taken place*” may seem self-evident. However, there are important questions over the exact moment when the intrusion detection system *can* detect the intrusion. It is clearly impossible in the general case to sound the alarm when mere intent is present. There is a better chance of raising the alarm when preparatory action is taking place, while the best chance comes when a bona fide violation has taken place, or is ongoing. The case where we consider an intrusion which is “about to take place” is interesting enough to warrant special

treatment. In military circles this falls under the heading of *indication and warning*; there are sufficient signs that something is imminent to ensure that our level of readiness is affected. In a computer security context, the study of such clues, many of which are of course not “technological” in nature, is not far advanced. It is an important subject, however, since it actually gives us the opportunity to ward off or otherwise hinder an attack. Without such possibilities, an alarm can only help to reduce the damage after the fact, or can only function as a deterrent.

2.2 Intrusion Detection Systems

The study of intrusion detection is today some twenty five years old. The possibility of automatic intrusion detection was first put forward in James Anderson’s classic paper [And80], in which he states that a certain class of intruders—the so-called *masqueraders*, or intruders who operate with stolen identities—could probably be detected by their departures from the set norm for the original user. Later the idea of checking all activities against a set security policy was introduced.

We can group intrusion detection systems into two overall classes: those that detect anomalies, hereafter termed *anomaly detection systems*, and those that detect the signatures of known attacks, hereafter termed *signature based systems*. Often the former automatically forms an opinion on what is ‘normal’ for the system, for example by constructing a profile of the commands issued by each user and then sounding the alarm when the subject deviates sufficiently from the norm. Signature systems, on the other hand, are most often programmed beforehand to detect the signatures of intrusions known of in advance.

These two techniques are still with us today, and with the exception of hybrid approaches nothing essentially new has been put forward in this area. Section 2.4 will explain these two approaches in terms of detection and estimation theory.

2.3 An Architectural Model of Intrusion Detection Systems

Since the publication of Anderson’s seminal paper [And80], several intrusion detection systems have been invented. Today there exists a sufficient number of systems in the field for one to be able to form some sort of notion of a ‘typical’ intrusion detection system, and its constituent parts. Figure 2.2 depicts such a system. Please note that not all possible data/control flows have been included in the figure, but only the most important ones.

Any generalised architectural model of an intrusion detection system would contain at least the following elements:

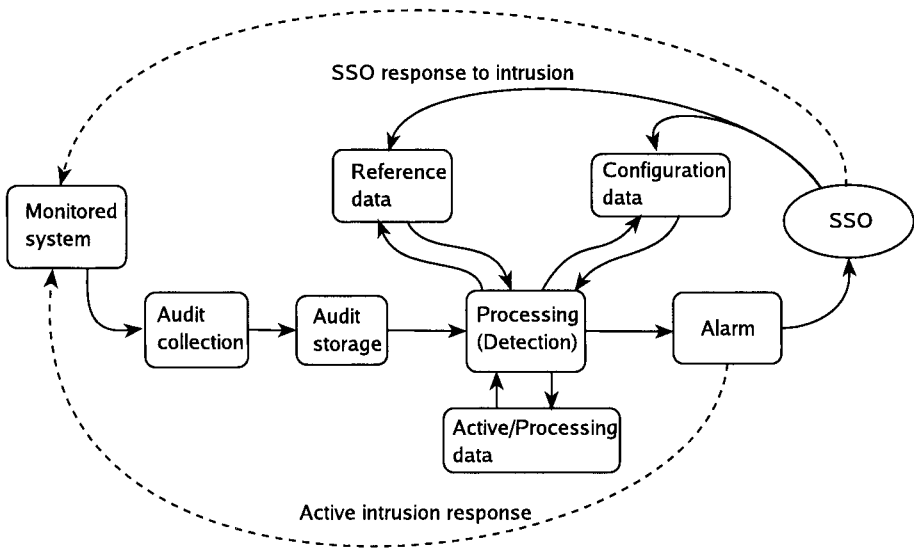


Figure 2.2. Organisation of a generalised intrusion detection system

Audit collection Audit data must be collected on which to base intrusion detection decisions. Many different parts of the monitored system can be used as sources of data: keyboard input, command based logs, application based logs, etc. In most cases network activity or host-based security logs, or both, are used.

Audit storage Typically, the audit data is stored somewhere, either indefinitely¹ for later reference, or temporarily awaiting processing. The volume of data is often exceedingly large², making this a crucial element in any intrusion detection system, and leading some researchers to view intrusion detection as a problem in audit data reduction [Fra94, ALGJ98]

Processing The processing block is the heart of the intrusion detection system. It is here that one or many algorithms are executed to find evidence (with some degree of certainty) in the audit trail of suspicious behavior. More will be said about the detector proper in Section 2.4.

Configuration data This is the state that affects the operation of the intrusion detection system: how and where to collect audit data, how to respond

¹Or at least for a long time—perhaps several months or years—compared to the processing turn around time.

²The problem of collecting sufficient but not excessive amounts of audit data has been described as “You either die of thirst, or you are allowed a drink from a fire hose.”

to intrusions, etc. This is therefore the SSO's main means of controlling the intrusion detection system. This data can grow surprisingly large and complex in a real world intrusion detection installation. Furthermore, it is relatively sensitive, since access to this data would give the competent intruder information on which avenues of attack are likely to go undetected.

Reference data The reference data storage stores information about known intrusion signatures—for misuse systems—or profiles of normal behavior—for anomaly systems. In the latter case the processing element updates the profiles as new knowledge about the observed behavior becomes available. This update is often performed at regular intervals in batches. Stored intrusion signatures are most often updated by the SSO, as and when new intrusion signatures become known. The analysis of novel intrusions is a highly skilled task. More often than not, the only realistic mode for operating the intrusion detection system is one where the SSO subscribes to some outside source of intrusion signatures. At present these are proprietary. In practice it is difficult, if not impossible, to make intrusion detection systems operate with signatures from an alternate source, even though it is technically feasible [LMPT98].

Active/processing data The processing element must frequently store intermediate results, for example information about partially fulfilled intrusion signatures. The space needed to store this active data can grow quite large.

Alarm This part of the system handles all output from the system, whether it be an automated response to suspicious activity, or more commonly the notification of a SSO.

2.4 Explaining Intrusion Detection From the Perspective of Detection and Estimation Theory⁴

Research into the automated detection of computer security violations is hardly in its infancy, yet little comparison has been made with the established field of detection and estimation theory (one exception being [LMS00]) the results of which have been found applicable to a wide range of problems in other disciplines. In order to explain the two major approaches behind intrusion detection principles we will attempt such a comparison, studying the problem of intrusion detection by the use of the introductory models of detection and estimation theory.

⁴This section is based on [Axc00b].

Classical Detection Theory

The problem of detecting a signal transmitted over a noisy channel is one of great technical importance, and has consequently been studied thoroughly for some time now. An introduction to detection and estimation theory is given in [Tre68], from which this section borrows heavily.

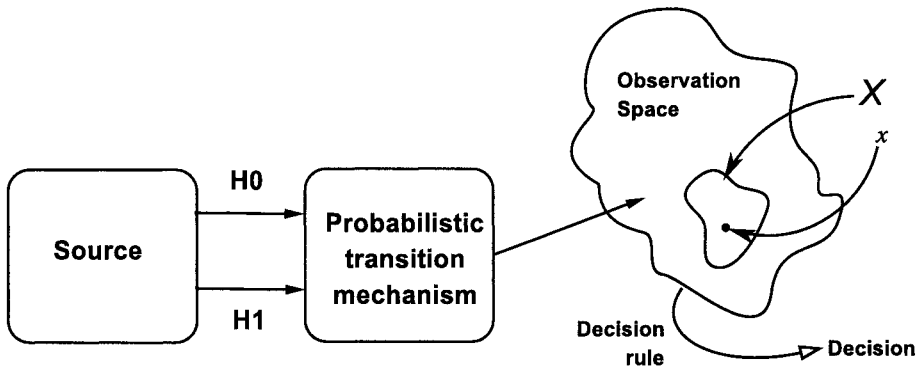


Figure 2.3. Classical detection theory model

In classical binary detection theory (see Figure 2.3) we should envisage a system that consists of a source from which originates one of two signals, $H0$ or $H1$, for *hypothesis zero* and *one* respectively. This signal is transmitted via some channel that invariably adds noise and distorts the signal according to a probabilistic transition mechanism. The output—what we receive—can be described as a point in a finite (multidimensional) observation space, for example x in Figure 2.3. Since this is a problem that has been studied by statisticians for some time, we have termed it the *classical detection model*. Based on an observation of the output of the source as transmitted through the probabilistic transition mechanism, we arrive at a decision. Our decision is based on a decision rule; for example: ‘Is or is not x in X ,’ where X is the region in the observation space that defines the set of observations that we believe to be indicative of $H0$ (or $H1$) (see Figure 2.3). We then make a decision as to whether the source sent $H0$ or $H1$ based on the outcome of the comparison of x and X .

Note that the source and signal model $H0$ and $H1$ could represent any of a number of interesting problems, and not only the case of transmitting a one or a zero. For example, $H1$ could represent the presence of a disease (and conversely $H0$ its absence), and the observation space could be any number of measurable physiological parameters such as blood count. The decision would then be one

of ‘sick’ or ‘healthy.’ In our case it would be natural to assign the symbol $H1$ to some form of intrusive activity, and $H0$ to its absence.

The problem is then one of deciding the nature of the probabilistic transition mechanism. We must choose what data should be part of our observation space, and on this basis derive a decision rule that maximizes the detection rate and minimizes the false alarm rate, or settle for some desirable combination of the two.

When deciding on the decision rule the *Bayes criterion* is a useful measurement of success [Tre68, pp. 24]. In order to conduct a Bayes test, we must first know the a priori probabilities of the source output (see Chapter 3 for further discussion). Let us call these P_0 and P_1 for the probability of the source sending a zero or a one respectively. Second, we assign a cost to each of the four possible courses of action. These costs are named C_{00} , C_{10} , C_{11} , and C_{01} , where the first subscript indicates the output from our decision rule—what we thought had been sent—and the second what was actually sent. Each decision or experiment then incurs a cost, in as much as we can assign a cost or value to the different outcomes. For example, in the intrusion detection context, the detection of a particular intrusion could potentially save us an amount that can be deduced from the potential cost of the losses if the intrusion had gone undetected. We aim to design our decision rule so that the *average* cost will be minimized. The expected value— R for *risk*—of the cost is then [Tre68, p. 9]:

$$\begin{aligned}
 R = & C_{00}P_0P(\text{say } H0|H0 \text{ is true}) \\
 & + C_{10}P_0P(\text{say } H1|H0 \text{ is true}) \\
 & + C_{11}P_1P(\text{say } H1|H1 \text{ is true}) \\
 & + C_{01}P_1P(\text{say } H0|H1 \text{ is true})
 \end{aligned} \tag{2.1}$$

It is natural to assume that $C_{10} > C_{00}$ and $C_{01} > C_{11}$, in other words the cost associated with an incorrect decision or misjudgment is higher than that of a correct decision. Given knowledge of the a priori possibilities and a choice of C parameter values, we can then construct a Bayes optimal detector.

Though Figure 2.3 may lead one to believe that this is a multidimensional problem, it can be shown [Tre68, p. 29] that a *sufficient statistic* can always be found whereby a coordinate transform from our original problem results in a new point that has the property that only one of its coordinates contains all the information necessary for making the detection decision. Figure 2.4 depicts such a case, where the only important parameter of the original multidimensional problem is named L .

It can furthermore be shown that the two main approaches to maximizing the desirable properties of the detection—the Bayes or Neyman-Pearson criteria—amount to the same thing; the detector finds a likelihood ratio (which will be a function only of the sufficient statistic above) and then compares this ratio with

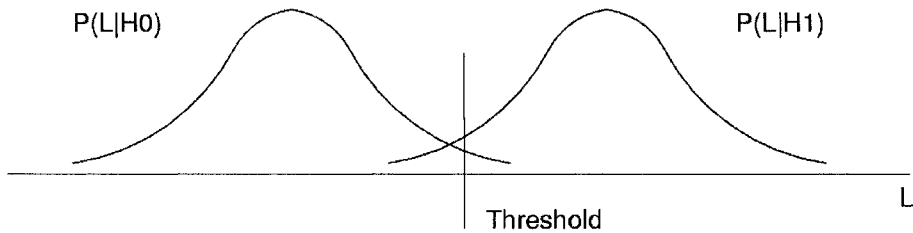


Figure 2.4. One dimensional detection model

a pre-set threshold. By varying the threshold in Figure 2.4, it can be seen that the detection ratio (where we correctly say $H1$) and the false alarm rate (where we incorrectly say $H1$) will vary in a predictable manner. Hence, if we have complete knowledge of the probability densities of $H0$ and $H1$ we can construct an optimal detector, or at least calculate the properties of such a detector. We will later apply this theory to explain anomaly and signature detection.

Application to the Intrusion Detection Problem

This section is a discussion of the way in which the intrusion detection problem may be explained in light of the classical model described above.

Source Starting with the *source*, ours is different from that of the ordinary radio transmitter because it is human in origin. Our source is a human computer user who issues commands to the computer system using any of a number of input devices. In the vast majority of cases, the user is benevolent and non-malicious, and he is engaged solely in non-intrusive activity. The user sends only $H0$, that is, non-intrusive activity. Even when the user is malicious, his activity will still mostly consist of benevolent activity. Some of his activity will however be malicious, that is, he will send $H1$. Note that *malicious* has to be interpreted liberally, and can arise from a number of different types of activities such as those described by the taxonomies in for example [LBMC94, LJ97]. Thus, for example, the use of a pre-packed exploit script is one such source of intrusive activity. A *masquerading*⁵ intruder can be another source of intrusive activity. In this case the activity that he initiates differs from the activity that the proper user would have originated.

It should be noted that we have only treated the binary case here, differentiating between ‘normal’ behavior and one type of intrusion. In reality there are many different types of intrusions, and different detectors are needed to detect

⁵A *masquerader* is an intruder that operates under false identity. The term was first used by Anderson in [And80].

them. Thus the problem is really a multi-valued problem, that is, in an operational context we must differentiate between $H0$ and $H1$, $H2$, $H3$... , where $H1-Hn$ are different types of intrusions. To be able to discriminate between these different types of intrusions, some statistical difference between a parameter in the $H0$ and $H1$ situation must be observable. This is simple, almost trivial, in some cases, but difficult in others where the observed behavior is similar to benevolent behavior. Knowledge, even if incomplete, of the statistical properties of the ‘signals’ that are sent is crucial to make the correct detection decision.

It should be noted that earlier classifications of computer security violations that exist [LBMC94, NP89, LJ97] are not directed at intrusion detection, and on closer study appear to be formulated on too high a level of representation to be directly applicable to the problem in hand. There are now a handful of studies that links the classification of different computer security violations to the problem of detection, in this case the problem of what traces are necessary to detect intrusions after the fact [ALGJ98, Bar04a, KMT04, Max03].

Probabilistic Transition Mechanism In order to detect intrusive behavior we have first to observe it. In a computer system context it is rare to have the luxury of observing user behavior directly, looking over the user’s shoulder while he provides a running commentary on what he is doing and intends to do. Instead we have to observe the user by other means, often by some sort of security logging mechanism, although it is also possible by observing the network traffic emanating from the user. Other more direct means have also been proposed, such as monitoring the user’s keystrokes.

In the usual application of detection theory, the probabilistic transition mechanism, or “channel”, often adds noise of varying magnitude to the signal. This noise can be modeled and incorporated into the overall model of the transmission system. The same applies to the intrusion detection case, although our “noise” is of a different nature and does not in general arise from nature, as described by physics. In our case we observe the subject by some (imperfect) means where several sources of noise can be identified. One such source is where other users’ behavior is mixed with that of the user under study, and it is difficult to identify the signal we are interested in.

If, for example, our user proves to be malicious, and sends TCP-syn packets from a PC connected to a network of PCs to a target host, intended to execute a SYN-flooding denial-of-service attack on that host. Since the source host is on a network of PCs—the operating systems of which are known to suffer from flaws that make them prone to sending packet storms that look like SYN-flooding attacks to the uninitiated⁶—it may be difficult to detect the malicious

⁶Or at least *were* prone to ten years ago.

user. This is because he operates from under the cover of the noise added by the poorly implemented TCP/IP stacks of the computers on the same source network. It can thus be much more difficult to build a model of our ‘channel’ than when the noise arises as a result of a purely physical process.

Observation Space Given that the action has taken place, and that it has been ‘transmitted’ through the logging system/channel, we can make observations. The set of possible observations, given a particular source and channel model, makes up our *observation space*. As said earlier, some results suggest that we can always make some sort of coordinate transformation that transforms all available information into one coordinate in the observation space. Thus in every detection situation we need to find this transform.

In most cases the computer security we are presented with will be discrete in nature, not continuous. This is different from the common case in detection theory where the signals are most often continuous in nature. In our case a record from a host-based security log will contain information such as commands or system calls that were executed, who initiated them, any arguments such as files read, written to, or executed, what permissions were utilized to execute the operation, and whether it succeeded or not. In the case of network data we will typically not have such high quality since the data may not contain all security relevant information; for example, we will not know exactly how the attacked system will respond to the data that it is sent, or whether the requested operation succeeded or not [PN98]. The question of what data to log in order to detect intrusions of varying kinds is central, but for a long time this question was largely unaddressed. We also know little of the way different intrusions manifest themselves when logged by different means.

Once again the literature is hardly extensive, although for example [ALGJ98, HL93, LB98] and more recently [Bar04b] have addressed the issues presented in this section, albeit from different angles.

Decision Rule Having made the coordinate transformation in the previous step we then need to decide on a threshold to distinguish between $H0$ and $H1$.

Thus our hope when we apply anomaly detection is that all that is not normal behavior for the source in question—that cannot be construed as $H0$ —is some sort of intrusive behavior. The question is thus to what degree abnormal equates to intrusive. This is perhaps most likely in the case of a *masquerader* who one may presume is not trained to emulate the user whose identity he has assumed. There are some studies that suggest that different users indeed display sufficiently different behavior for them to be told apart [LB98].

Existing Approaches to Intrusion Detection

For a survey of existing approaches to intrusion detection see [BAJ03]. Here we will only outline the two major methods of intrusion detection: *anomaly detection* and *signature detection*. These have been with us since the inception of the field. In short, *anomaly detection* can be defined as looking for the unexpected—that which is unusual is suspect—at which point the alarm should be raised. *Signature detection*, on the other hand, relies on the explicit codifying of ‘illegal’ behavior, and when traces of such behavior is found the alarm is raised.

Anomaly Detection Taking the basic outline of detection and estimation theory laid out in the beginning of this section, we can elaborate upon it in describing these methods. In contrast to the model in Figure 2.4, where we have knowledge of both $H0$ and $H1$, here we operate without any knowledge of $H1$. Thus we choose a region in our observation space— X in Figure 2.3. To do so, we must transform the observed, normal behavior in such a manner that it makes sense in our observation space context. The region X will contain the transformed normal behavior, and typically also behavior that is ‘close’ to it, in such a way as to provide some leeway in the decision, trading off some of the detection rate to lower the false alarm rate. The detector proper then flags all occurrences of x in X as no alarm, and all occurrences of x not in X as an alarm. Note that X may be a disjoint region in the observation space.

Signature Detection The signature detector detects evidence of intrusive activity irrespective of the model of the background traffic; these detectors have to be able to operate no matter what the background traffic, looking instead for patterns or signals that are thought by the designers to stand out against any possible background traffic. Thus we choose a region in our observation space, but in this instance we are only interested in known intrusive behavior. Thus X will here only encompass observations that we believe stem from intrusive behavior plus the same leeway as before, in this case trading off some of the false alarm rate to gain a greater detection rate in the face of ‘modified’ attacks. During detector operation we flag all occurrences of x in X as an alarm, and all other cases as no alarm. X here may also consist of several disjoint regions, of course.

Comparison with Bayes Optimal Detectors It is an open question to what degree detectors in these classes can be made to, or are, approximate Bayes optimal detectors. In the case of non-parametric intrusion detectors—detectors where we cannot trade off detection rate for false alarm rate by varying some parameter of the detector—merely studying the receiver operating characteristics (ROC) curve cannot give us any clue as to the similarity to a Bayes optimal

detector. This is because the ROC curve in this case only contains one point, and it is impossible to ascertain the degree to which the resulting curve follows the optimal Bayes optimal detector. (See Chapter 3 for a brief introduction to ROC curves, and [Tre68] for a thorough treatment).

Summary

The dichotomy between *anomaly detection* and *signature detection* that is present in the intrusion detection field, vanishes (or is at least weakened) when we study the problem from the perspective of classical detection theory. If we wish to classify our source behavior correctly as either $H0$ or $H1$, knowledge of both distributions of behavior will help us greatly when making the intrusion detection decision. Interestingly, early on only few research prototypes took this view [Lee99, BAJ03]; all others were firmly entrenched in either the $H0$ or $H1$ camp. It may be that further study of this class of detectors will yield more accurate detectors, especially in the face of attackers who try to modify their behavior to escape detection. A detector that operates with a strong source model, taking both $H0$ and $H1$ behavior into account, will most probably be better able to qualify its decisions by stating strongly that this behavior is not only known to occur in relation to certain intrusions, and further is not a known benign or common occurrence in the supervised system.

The detectors we have developed in connection with this book (except for the one in Chapter 4) all take both $H0$ and $H1$ into account.

Understanding Intrusion Detection through
Visualization

Axelsson, S.; Sands, D.

2006, XX, 145 p. 34 illus., Hardcover

ISBN: 978-0-387-27634-2