2

An Introduction to Information Assurance

Clay Shields

2.1 Introduction

The intuitive function of computer security is to limit access to a computer system. With a perfect security system, information would never be compromised because unauthorized users would never gain access to the system. Unfortunately, it seems beyond our current abilities to build a system that is both perfectly secure and useful. Instead, the security of information is often compromised through technical flaws and through user actions.

The realization that we cannot build a perfect system is important, because it shows that we need more than just protection mechanisms. We should expect the system to fail, and be prepared for failures. As described in Sect. 2.2, system designers not only use mechanisms that *protect* against policy violations, but also *detect* when violations occur, and *respond* to the violation. This response often includes analyzing why the protection mechanisms failed and improving them to prevent future failures.

It is also important to realize that security systems do not exist just to limit access to a system. The true goal of implementing security is to protect the information on the system, which can be far more valuable than the system itself or access to its computing resources. Because systems involve human users, protecting information requires more than just technical measures. It also requires that the users be aware of and follow security policies that support protection of information as needed.

This chapter provides a wider view of information security, with the goal of giving machine learning researchers and practitioners an overview of the area and suggesting new areas that might benefit from machine learning approaches. This wider view of security is called *information assurance*. It includes the technical aspects of protecting information, as well as defining policies thoroughly and correctly and ensuring proper behavior of human users and operators. I will first describe the security process. I will then explain the standard model of information assurance and its components, and, finally, will describe common attackers and the threats they pose. I will conclude

with some examples of problems that fall outside much of the normal technical considerations of computer security that may be amenable to solution by machine learning methods.

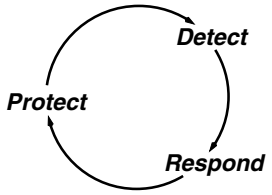

Fig. 2.1. The security cycle

2.2 The Security Process

Human beings are inherently fallible. Because we will make mistakes, our security process must reflect that fact and attempt to account for it. This recognition leads to the cycle of security shown in Fig. 2.1. This cycle is really very familiar and intuitive, and is common in everyday life, and is illustrated here with a running example of securing an automobile.

2.2.1 Protection

Protection mechanisms are used to enforce a particular policy. The goal is to prevent things that are undesirable from occurring. A familiar example is securing an automobile and its contents. A car comes with locks to prevent anyone without a key from gaining access to it, or from starting it without the key. These locks constitute the car's protection mechanisms.

2.2.2 Detection

Since we anticipate that our protection mechanisms will be imperfect, we attempt to determine when that occurs by adding detection mechanisms. These monitor the system, try to locate any policy violations that have occurred, and then provide an alert or alarm to that fact. Our familiar example is again a car. We know that a determined thief can gain entry to a car, so in many cases, cars have alarm systems that sound loudly to attract attention when they detect what might be a theft.

However, just as our protection mechanisms can fail or be defeated, so can detection mechanisms. Car alarms can operate correctly and sound the alarm when someone is breaking in. This is termed a *true positive*; the event that is looked for is detected. However, as many city residents know, car alarms can

also go off when there is no break-in in progress. This is termed a *false positive*, as the system is indicating it detected something when nothing was happening. Similarly, the alarm can fail to sound when there is an intrusion. This is termed a *false negative*, as the alarm is indicating that nothing untoward is happening when in fact it is. Finally, the system can indicate a *true negative* and avoid sounding when nothing is going on.

While these terms are certainly familiar to those in the machine learning community, it is worth emphasizing the fallibility of detection systems because the rate at which false results occur will directly impact whether the detection system is useful or not. A system that has a high false-positive rate will quickly become ignored. A system that has a high false-negative rate will be useless in its intended purpose.

2.2.3 Response

If, upon examination of an alert provided by our detection system, we find that a policy violation has occurred, we need to respond to the situation. Response varies, but it typically includes mitigating the current situation, analyzing what happened, recovering from any damage, and improving the protection and detection mechanisms to prevent similar future occurrences.

For example, if our car alarm sounds and we see someone breaking in, we might respond by summoning the police to catch or run off the thief. Some cars have devices that allow police to determine their location, so that if a car is stolen, it can be recovered. Afterwards, we might try to prevent future incidents by adding a locking device to the steering wheel or parking in a locked garage. If we find that the car was broken into and the alarm did not sound, we might choose also to improve the alarm system.

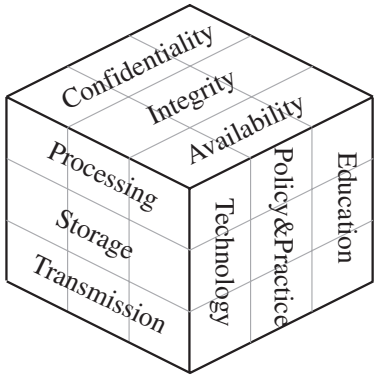

Fig. 2.2. The standard model of information assurance

2.3 Information Assurance

The standard model of information assurance is shown in Fig. 2.2 [4]. In this model, the security properties of confidentiality, integrity, and availability of information are maintained in the different locations of storage, transport, and processing by technological means, as well as through the process of educating users in the proper policies and practices. Each of these properties, location, and processes is described below.

The term *assurance* is used because we fully expect failures and errors to occur, as described above in Sect. 2.2. Recognizing this, we do not expect perfection and instead work towards a high level of confidence in the systems we build.

Though this model can apply to virtually any system which includes information flow, such as the movement of paper through an office, our discussion will naturally focus on computer systems.

2.3.1 Security Properties

The first aspects of this model we will examine are the security properties that can be maintained. The traditional properties that systems work towards are confidentiality, integrity, and availability, though other properties are sometimes included. Because different applications will have different requirements, a system may be designed to maintain all of these properties or only a chosen subset as needed, as described below.

Confidentiality

The confidentiality property specifies that only entities authorized to access some particular information are allowed to do so. This is the property that maintains the secrecy of information on a need-to-know basis, and is the most intuitive.

The most common mechanisms for protecting confidentiality are access control and encryption. Access control mechanisms prevent any reading of the information until the accessing entity, either a person or computer process acting on behalf of a person, prove that it is authorized to do so. Encryption does not prevent access to the information, but instead obfuscates the information so that even if it is read, it is not understandable.

The mechanisms for detecting violations of confidentiality and responding to them vary depending on the situation. In the most general case, public disclosure of the information would indicate loss of confidentiality. In an electronic system, violations might be detectable through audit and logging systems. In situations where the actions of others might be influenced by the release of confidential information, such changes in behavior might indicate a violation. For example, during World War II, an Allied effort broke the German Enigma encryption system, violating the confidentiality of German

communications. Concerned that unusual military success might indicate that Enigma had been broken, the Allies were careful to not exploit all information gained [5]. Though it will vary depending on the case, there may be learning situations that involve monitoring the actions of others to see if access to confidential information has been compromised.

There might be an additional requirement that the existence of information be kept confidential as well, in which case, encryption and access control might not be sufficient. This is a more subtle form of confidentiality.

Integrity

In the context of information assurance, *integrity* means that only authorized entities can alter information within a system. This is the property that keeps information from being changed when it should not be.

While we will use the above definition of *integrity*, it is an overloaded term and other meanings exist. *Integrity* can be used to describe the reliability of information. For example, a data source has integrity if it provides accurate data. This is sometimes referred to as *origin integrity*. *Integrity* can also be used to refer to a state that exists in several systems; if the state is consistent, then there is high integrity. If the distributed states are inconsistent, then there is low integrity.

Mechanisms exist to protect data integrity and to detect when it has been violated. In practice, protection mechanisms are similar to the access control mechanisms for confidentiality, and in implementation may share common components. Detecting integrity violations may involve comparing the data to a different copy, or the use of cryptographic hashes. Response typically involves repairing the changes by reverting to an earlier, archived copy.

Availability

Availability is the property that the information on a system is obtainable when needed. Information that is kept secret and unaltered might still be made unavailable by attackers conducting *denial-of-service* attacks.

The general approach to protecting availability is to limit the amount of system resources that can be consumed, either by rate-limiting or by requiring access control. Another common approach is to over-provision the system. Detection of availability is generally conducted by polling to see if the resources are there. It can be difficult to determine if some system is unavailable because of attack or because of some system failure. In some situations, there may be learning problems to be solved to differentiate between failure and attack conditions.

Response to availability problems generally includes reducing the system load, or adding more capacity to a system.

Other Components

The properties above are the classic components of security, and are sufficient to describe many situations. However, there has been some discussion within the security community for the need for other properties to fully capture requirements for other situations. Two of the commonly suggested additions, authentication and non-repudiation, are discussed below.

Authentication

Both the confidentiality properties and integrity properties include a notion of authorized entities. The implication is that the system can accurately identify entities in some manner and, given their identity, provide or deny access. The authentication property ensures that all entities in a system have their identities properly verified.

There are a number of ways to conduct authentication and protect against false identification. For individuals, the standard mnemonic for describing classes of authentication mechanisms is, *What you are*, *what you have*, and *what you know*.

- "What you are" refers to specific physical attributes of an individual that can serve to differentiate him or her from others. These are commonly biometric measurements of such things as fingerprints, hand size and shape, voice, or retinal patterns. Other attributes can be used as well, such as a person's weight, gait, face, or potentially DNA. It is important to realize that these systems are not perfect. They have false-positive and false-negative rates that can allow false authentication or prohibit legitimate users from accessing the system. Often the overall accuracy of a biometric system can be improved by measuring different attributes simultaneously. As an aside, many biometric systems have been shown to be susceptible to simple attacks, such as plastic bags of warm water placed on a fingerprint sensor to reactivate the prior latent print, or pictures held in front of a camera [6, 7]. Because these attacks are generally observable, it may be more appropriate for biometric authentication to take place under human observation. It might be a vision or machine learning problem to determine if this type of attack is occurring.
- "What you have" describes some token that is carried by a person that the system expects only that person to have. This token can take many forms. In a physical system, a key could be considered an access token. Most people have some form of identification, which is a token that can be used to show that the issuer of the identification has some confidence in the carrier's identity. For computer systems, there are a variety of authentication tokens. These commonly include devices that generate pass codes at set intervals. Providing the correct pass code indicates possession of the device.

- "What you know" is the most familiar form of authentication for computer users. In this form of authentication, users prove their identity by providing some information that only they would know that can be verified. The most common example of this is a password, which is a secret shared by the individual and the end system conducting the authentication. The private portion of a public/private key pair is also an example of what you know.

More recently, it has been shown that it is possible to use location as another form of authentication. With this "where you are" authentication, systems can use signals from the Global Positioning System to verify that the authentication attempt is coming from a particular location [8].

Authenticating entities across a network is a particularly subtle art. Because attackers can potentially observe, replay, and manipulate network traffic, designing protocols that are resistant to attack is very difficult to do correctly. This has been a significant area of research for some time [9].

The mechanisms outlined above provide the basis for authentication protection. Detecting authentication failures, which would be incorrectly identifying a user as a legitimate user, can often be done on the basis of behavior after authentication. There is a significant body of work addressing user profiling to detect aberrant behavior that might indicate an authentication failure. One appropriate response is to revoke the credentials gained through authentication. The intruder can also be monitored to better understand attacker behavior.

Non-repudiation

The non-repudiation property makes it difficult for any entity to deny that it performed some action. A system with non-repudiation will allow entities to be held responsible for what they do. Very few computer systems have effective non-repudiation mechanisms. In general, logging and audit data is recorded, but is often unreliable. More effective non-repudiation systems require the use of strong cryptographic mechanisms, though these require significant overhead for additional processing and key distribution.

System Security Requirements

Different systems have different security requirements, which might include some or all of the properties discussed above. A financial system might need all five: Confidentiality is required to protect the privacy of records; integrity is needed to maintain a proper balance; availability allows access to money when required; authentication keeps unauthorized users from withdrawing funds; and non-repudiation keeps users from arguing that they did not take funds out and keeps the institution from denying it received a deposit.

Other systems do not require that level of security. For example, a Web page may be publicly available and therefore not require any confidentiality.

The owner of the page might still desire that the integrity of the page be maintained and that the page be available. The owner of a wiki might allow anyone to edit the page and hence be unconcerned with integrity, but might require that users authenticate to prevent non-repudiation of what they edit.

2.3.2 Information Location

The model of information assurance makes a clear distinction about where information resides within a system. This is because the mechanisms used to protect, detect, and respond differ for each case.

Processing

While information is being processed in a computer system, it is loaded into memory, some of which may be virtual memory pages on a disk, and into the registers of the CPU. The primary protection mechanisms in this case are designed to prevent processes on the system from reading or altering each other's memory space. Modern computer systems contain a variety of hardware and software mechanisms to provide each process with a secure, independent memory space.

Confidentiality can also be lost through information leaking from a process. This can happen through a covert channel, which is a mechanism that uses shared system resources not intended for communication to transmit information between processes [10]. It is possible to prevent or rate-limit covert channels, though it can be difficult to detect them. Response varies, but includes closing the channel through system updates. Loss of confidentiality can also occur through electromagnetic radiation from system components, such as the CPU, bus, video card, and CRT. These produce identifiable signals that can be used to reconstruct information being processed on the system [11, 12]. Locations that work with highly classified information are often constructed to keep this radiation from escaping.

Storage

Information in storage resides on some media, either within the system or outside of it. The protection mechanisms for information stored on external media are primarily physical, so that the media cannot be stolen or accessed. It is also possible and frequently desirable to encrypt information that is stored externally. Detection often consists of alarm systems to detect illicit access, and inventory systems to detect missing media. To detect integrity violations, cryptographic hashes can be computed for the stored data and kept separately from the media, then periodically checked [13]. At the end of its useful lifetime, media should be destroyed instead of discarded.

Information that is stored within a system is protected by operating systems mechanisms that prevent unauthorized access to the data. These include

access control mechanisms and, increasingly, mechanisms that keep stored information encrypted. There are many methods of detecting unauthorized access. These generally fall under the classification of *intrusion detection*. Intrusion detection systems can further be classified as *signature-based*, which monitor systems for known patterns of attack, or as *anomaly detection*, which attempt to discern attacks by identifying abnormal activity.

Transport

Information can be transported either physically or electronically. While it is natural to think of transmitted data over a network, for large amounts of data it can be significantly faster to send media through the mail or via an express delivery service. Data transported in this manner can be protected using encryption or physical security, such as locked boxes.

Data being transported over the network is best protected by being encrypted, and this functionality is common in existing software. In the future, quantum cryptographic methods will increasingly be used to protect data in transmission. Using quantum cryptography, two communicating parties can agree on an encryption key in a way that inherently detects if their agreement has been eavesdropped upon [14].

2.3.3 System Processes

While most computer scientists focus on the technological processes involved in implementing security, technology alone cannot provide a complete security solution. This is because human users are integral in maintaining security. The model of information assurance recognizes this, and gives significant weight to human processes. This section provides more detail about the processes that are used to provide assurance.

Technology

Every secure information system requires some technological support to be secure. In our discussion thus far, we have mentioned a number of technological mechanisms that exist to support the protect, detect, and respond cycle. These include systems that provide authentication; access control mechanisms that limit what authenticated users can view and change; and intrusion detection systems that identify when these prior mechanisms have failed.

There are other technological controls that protect information security that are not part of computer systems, however, and which are often forgotten. The foremost of these are physical security measures. Access control on a computer system is of little use if an attacker has physical access and can simply steal the computer or its archive media and off-load the data later. Large corporations are typically more aware of this than universities, and often implement a number of controls designed to limit physical access. The efficacy

of these devices can vary, however. Some systems use cards with magnetic stripes that encode an employee number that is also shown on the front of the card, which may be worn around the neck. Anyone who is able to read this number can then duplicate the card with a $600 card writer. Radio frequency identification (RFID) tags are also becoming popular. These frequently respond to a particular radio-frequency query with a static ID. Because there is no control over who can query the tag, anyone can read and potentially duplicate the tag. Impersonation in these cases may be relatively simple for someone who feels comfortable that they might not be noticed as out of place within a secure area.

Policy and Practice

While technological controls are important, they are not sufficient simply because they are designed to allow some access to the system. If the people who are permitted to access systems do not behave properly, they can inadvertently weaken the security of the system. A common example is users who open or run attachments that they receive over e-mail. Because users are allowed to run processes on the system, the access control mechanisms prove ineffective.

Organizations that do security well therefore create policies that describe how they expect their users to act, and provide best-practice documents that detail what can be done to meet these policies. Again, these policies must go beyond the computer system. They should include physical security as well as policies that govern how to answer phones, how to potentially authenticate a caller, and what information can be provided. These policies are directed towards stopping *social engineering*, in which an outside attacker tries to manipulate people into providing sufficient information to access the system.

Education

Having defined policies and practices is not sufficient. Users must know them, accept them, and follow them. Therefore, user education is a necessity. Proper education includes new-user orientation and training, as well as recurring, periodic training to keep the material fresh. Some organizations include security awareness and practice as part of job performance evaluation.

2.4 Attackers and the Threats Posed

It is difficult to determine what security measures are needed without an understanding of what capabilities different types of attackers possess. In this section, we will examine different classes of attackers, what unique threats each might pose, and how those threats are addressed in the information assurance model.

It is important to note that attackers will generally attempt to compromise the system the easiest way that they can, given their capabilities. For example, an attacker might have access to an encrypted password file and to network traffic. In this case, it might be easier to "sniff" unencrypted passwords off the network instead of making the effort to decrypt the password file. A similar attack for someone with physical access to the system might be to place a hardware device to capture keystrokes instead of making the effort of guessing an encryption key. Other attackers might find it easier to attack the encryption; for example, government intelligence agencies might want to limit their exposure to detection. In this case, given their desire for secrecy and massive computing facilities, it might be easiest to attack the encryption.

2.4.1 Worker with a Backhoe

While they hardly seem like fearsome hackers and appear quite comical, construction workers might be one of the most damaging accidental attackers. Given the prevalence of underground power and network wiring, it is a common occurrence for lines to be severed. This can easily rob a large area of power and network access, making services unavailable. It can also take a significant amount of time to make repairs. The best defense is over-provisioning through geographically separate lines for networking or power, or possession of a separate power generator.

As a military tactic, the equivalent of an attacker with a backhoe has proven quite effective in the past, and could be again in the future. In the early days of World War I, British sailors located, raised, and severed an underwater telephone line that was used to transmit orders to the German Navy. Without the telephone line, the Germans had to transmit orders over radio, allowing the British to attack the encryption, eventually with significant success [5]. It is easy to believe that similar actions could occur today to force broadcast communication.

2.4.2 Ignorant Users

Many modern security problems are caused by otherwise well-intentioned users who make mistakes that weaken, break, or bypass security mechanisms. Users who open or run attachments received by e-mail are a clear example of this. Similarly, users who are helpful when contacted over the phone and provide confidential internal information, such as the names of employees and their phone numbers or even passwords, pose a threat. These types of employees are best prevented using proper policies, practices, and education.

2.4.3 Criminals

While most criminals lack any significant computer savvy, they are a serious threat because of the value of computer equipment. Theft of electronics is a

common occurrence, because of the potential resale value. Small items, such as laptops and external drives, are easy to steal and can contain significant amounts of information. Such information might have inherent value – passwords and account numbers are examples. Theft or misplacement could also cause financial loss as a result of legal action, especially if the lost data are like medical records, which should be kept private. Unfortunately, there is no way to know if equipment has been stolen for its value or to gain access to its information, and generally the worst case should be assumed.

2.4.4 Script Kiddies

The attackers discussed thus far have not been specifically targeting information systems. The somewhat denigrating term *script kiddie* applies to attackers who routinely attempt to remotely penetrate the security of networked computer systems, but lack the skills or knowledge to do so in a sophisticated way. Instead, they use a variety of tools that have been written by more capable and experienced people.

While they generally do not have a specific target in mind, script kiddies tend to be exceptionally persistent, and will scan hundreds of computers looking for vulnerabilities that they are able to exploit. They do not present a severe threat individually, but will eventually locate any known security hole that is presented to the network. As an analogy, imagine a group of roving youths who go from house to house trying the doors and windows. If the house is not properly secured, they will eventually find a way in. Fortunately, script kiddies are relatively easy to stop with good technological security practice.

2.4.5 Automated Agents

While script kiddies are often actively looking for security vulnerabilities, the scope of their efforts pale compared to the number of automated agents in the current Internet. These are programs, often called *malware*, that run with the sole purpose of spreading themselves to as many computers as possible. Many of these then provide their creator the ability to access information within a system, or to use its resources for other attacks. While there are many types of malware, there are a few specific types that merit mention.

Worm

A worm is a self-propagating piece of code that exploits vulnerabilities in remote systems to spread itself. Typically, a worm will infect a system and then start scanning to find other vulnerable systems and infect those. A worm might also have other functionality in its payload, including notifying its creator that it has compromised a new host and allowing access to it. It might also scan the compromised machine for interesting information and then send it to its creator.

Virus

Though the term *virus* has fallen into common use to describe any type of malware which spreads between computers, a more precise definition is that it is a piece of code which gets added to existing programs that only runs when they run. At that time, the virus adds its code to other programs on the system.

Trojan

Named after the famous Trojan horse, a *Trojan* is a piece of code that purports to do one thing but actually does another, or does what it says while also maliciously doing something else.

It should be immediately evident that a clear classification of malware into these separate categories may not be possible because one piece of malicious code may exhibit more than one of these characteristics. Many recent worms, for example, were also Trojans. They spread over the network directly, but also would search each machine compromised for e-mail addresses and then falsify e-mail that included a Trojan version of the worm. If the recipient were to open and run the attachment, the worm would continue from there.

These agents are stopped by common technological measures, the existence of which indicate how large the problem is. Unfortunately, it can be time-consuming and expensive to apply the proper patches to a large network of computers. Additionally, new malware variants are appearing that target new operating systems, like those in cellular phones, which do not have the same wealth of protection mechanisms.

2.4.6 Professional System Crackers

Unlike script kiddies, who lack the skills and experience to penetrate a specific target, professional crackers master a broad set of tools and have the intelligence and sophistication to pick and penetrate a particular target. They might do so on behalf of a government, or for financial gain, either independently or as part of an organized crime ring. While part of the attack might be conducted remotely over the network, they might also attempt to gain physical access to a particular target; to go through trash looking for useful information; or to gain the assistance of a helpful but ignorant user.

These attackers can be subtle and patient. There is no simple solution to mitigating the threat they present; instead, the full range of security measures is required.

2.4.7 Insiders

While the most popular image of a computer attacker is that of the professional cracker, they account for only a very small percentage of all attacks.

Instead, the most common attacker, and the one who is most often successful, is the insider [15]. An insider is someone who has access to some or all parts of the computer system, then misuses that access. Note that access may not be electronic; an insider may simply step over to someone else's desk while they are away and use their computer.

The insider is the most subtle and difficult attacker to identify. There is perhaps significant room for detecting insider attacks.

2.5 Opportunities for Machine Learning Approaches

It is evident from the other chapters in this book that machine learning and data mining are naturally most applicable to the detection phase of the security cycle. This section contains suggestions for other areas that might be amenable to machine learning approaches.

- When an attacker manages to acquire data without being detected, the information often ends up publicly available on the Internet. It might be possible to detect successful intrusions by making queries to search engines, such as Google. The difficulty here might not be a machine learning problem, but a data retrieval one: How is it possible to find information through queries without revealing what the information is to an attacker observing the queries?
- Many biometric authentication systems are subject to attacks that lead to false positives in identification. Most of these attacks are easily detected by human observers. A vision or machine learning problem might be to perform automated observation of biometric systems to detect these attacks.
- Education is an important part of the security process. While not all failures of proper user education will result in loss of confidentiality, integrity, or availability of data, problems short of these bad results might indicate the potential for future problems. Depending on the system, it might be possible to identify user behavior that does not result in a security violation but indicates that the user is not aware of good security practice.
- Insiders are the most insidious attackers, and the hardest to detect. One approach to detecting and identifying insiders might be to correlate user idle times between machines that are located in close proximity. A user becoming idle shortly before some other system ceases its idle time could indicate a user walking over to and using another unlocked system.
- Similarly, many companies use authentication systems that allow the physical location of employees to be known to some degree. Using data from these systems, it might be possible to identify insider attackers by finding odd movements or access patterns within a building or campus.
- Some insiders do not need to move to conduct attacks; instead, they are given broad access to a data processing system and trusted to limit the

data they examine to what they need to do their job. Without knowing what particular subset of data they should have access to, it might be possible to detect insider attackers based on the patterns of data access that are different than others who have similar responsibilities.

- Many outside attackers succeed by exploiting the trust and helpfulness of people within an organization. It might be possible to detect social engineering attacks by tracking patterns of phone calls coming into an organization. This data would likely be available in phone records.
- It can be difficult to classify availability failures as accidental or intentional. For example, a sudden increase in network consumption can indicate a denial-of-service attack, or simply a suddenly popular Web link. It might be possible to differentiate between them by examination of the network traffic.
- Automated agents, such as worms or Trojans, might be detectable based on patterns of outgoing network traffic.

2.6 Conclusion

Most machine learning work has focused on detecting technical attacks that originate from outside a particular network or system. This is actually a very small part of the security space. The ideas above touch on some aspects of security that seem to have appropriate data available, but that do not seem to have been as closely examined. There are certainly many existing and emerging areas where machine learning approaches can bring new improvements in security.

http://www.springer.com/978-1-84628-029-0

Machine Learning and Data Mining for Computer Security
Methods and Applications
Maloof, M.A. (Ed.)
2006, XVI, 210 p., Hardcover
ISBN: 978-1-84628-029-0