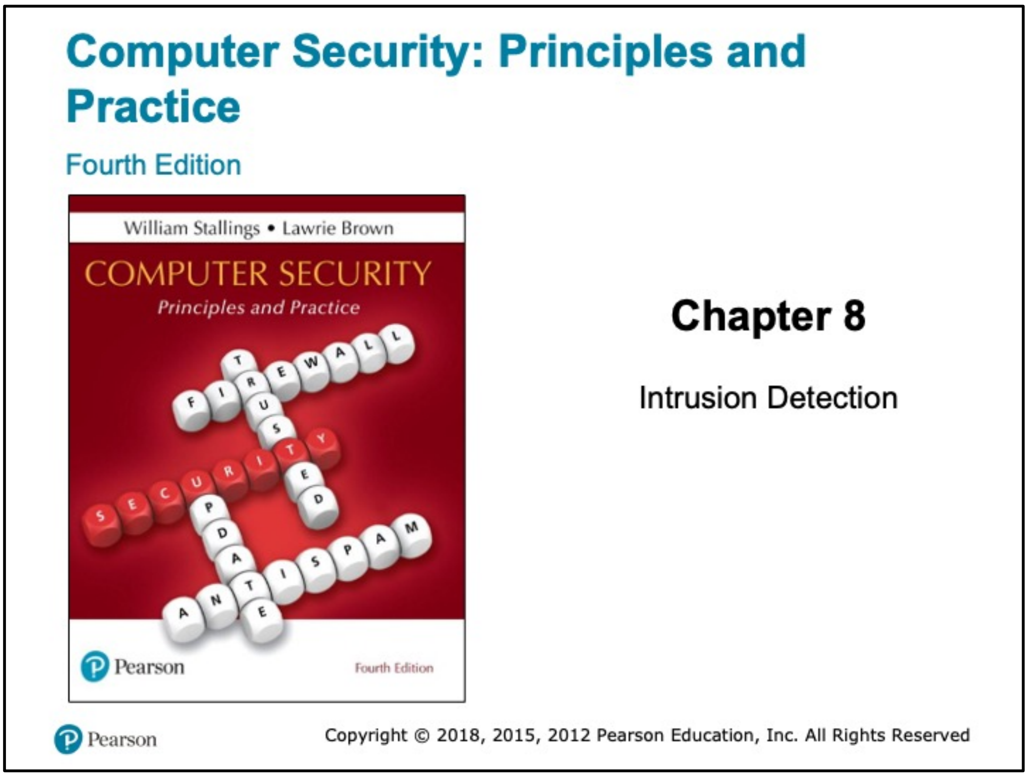


Computer Security: Principles and Practice

Fourth Edition

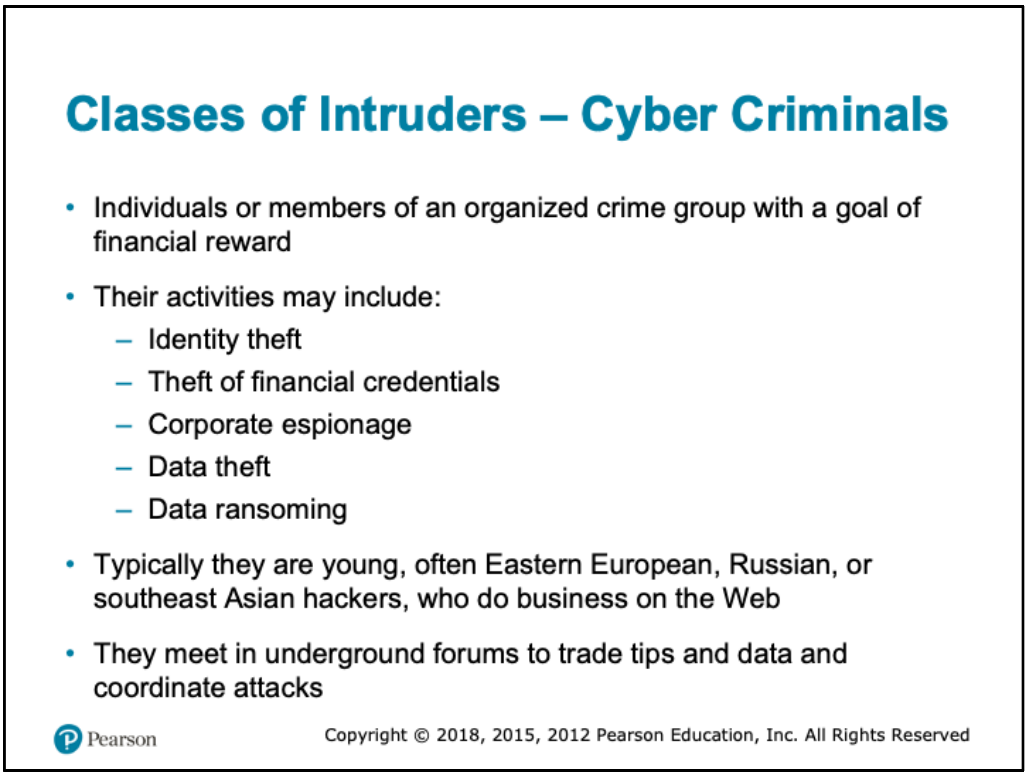
Chapter 8

Intrusion Detection

If this PowerPoint presentation contains mathematical equations, you may need to check that your computer has the following installed:
1) MathType Plugin
2) Math Player (free versions available)
3) NVDA Reader (free versions available)

A significant security problem for networked systems is hostile, or at least unwanted, trespass by users or software. User trespass can take the form of unauthorized logon or other access to a machine or, in the case of an authorized user, acquisition of privileges or performance of actions beyond those that have been authorized. Software trespass includes a range of malware variants as we discuss in Chapter 6.

This chapter covers the subject of intrusions. First, we examine the nature of intruders and how they attack, then look at strategies for detecting intrusions.



Classes of Intruders – Cyber Criminals

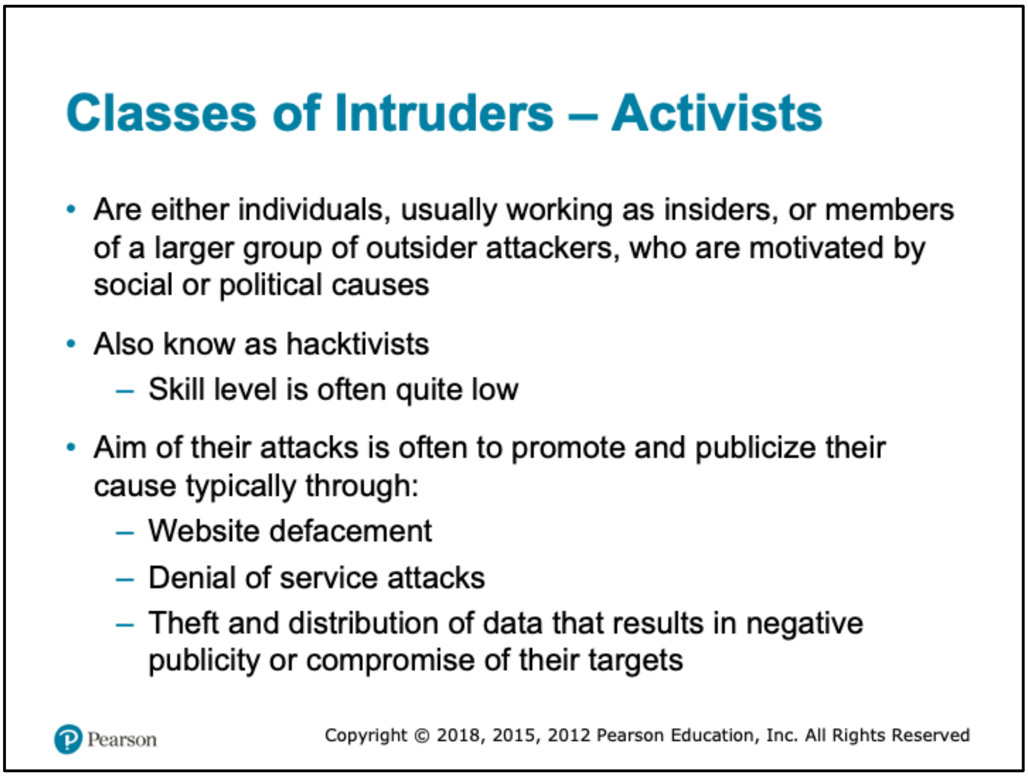
- Individuals or members of an organized crime group with a goal of financial reward
- Their activities may include:
  - Identity theft
  - Theft of financial credentials
  - Corporate espionage
  - Data theft
  - Data ransoming
- Typically they are young, often Eastern European, Russian, or southeast Asian hackers, who do business on the Web
- They meet in underground forums to trade tips and data and coordinate attacks

Pearson

One of the key threats to security is the use of some form of hacking by an intruder, often referred to as a hacker or cracker. Verizon [VERI16] indicates that 92% of the breaches they investigated were by outsiders, with 14% by insiders, and with some breaches involving both outsiders and insiders. They also noted that insiders were responsible for a small number of very large dataset compromises. Both Symantec [SYMA16] and Verizon [VERI16] also comment that not only is there a general increase in malicious hacking activity, but also an increase in attacks specifically targeted at individuals in organizations and the IT systems they use. This trend emphasizes the need to use defense-in-depth strategies, since such targeted attacks may be designed to bypass perimeter defenses such as firewalls and network-based Intrusion detection systems (IDSs).

As with any defense strategy, an understanding of possible motivations of the attackers can assist in designing a suitable defensive strategy. Again, both Symantec [SYMA16] and Verizon [VERI16] comment on the following broad classes of intruders:

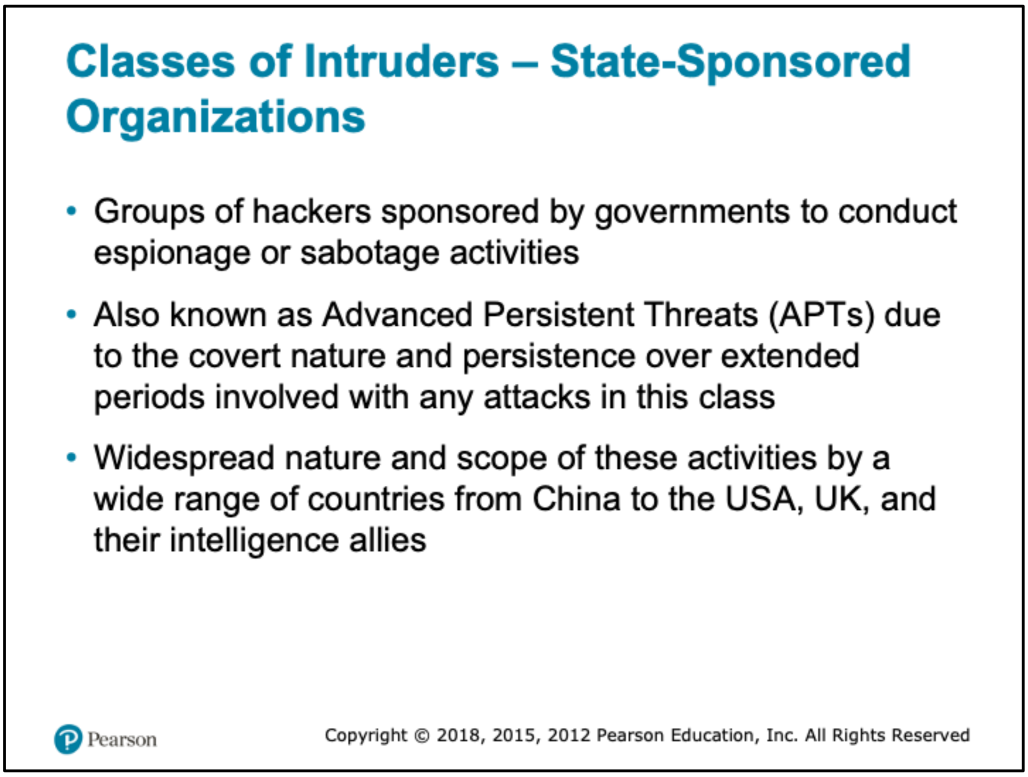
• **Cyber criminals**: Are either individuals or members of an organized crime group with a goal of financial reward. To achieve this, their activities may include identity theft, theft of financial credentials, corporate espionage, data theft, or data ransoming. Typically, they are young, often Eastern European, Russian, or southeast Asian hackers, who do business on the Web [ANTE06]. They meet in underground forums with names like DarkMarket.org and theftservices.com to trade tips and data and coordinate attacks. For some years reports such as [SYMA16] have quoted very large and increasing costs resulting from cyber-crime activities, and hence the need to take steps to mitigate this threat.



Classes of Intruders – Activists

- Are either individuals, usually working as insiders, or members of a larger group of outsider attackers, who are motivated by social or political causes
- Also know as hacktivists
  - Skill level is often quite low
- Aim of their attacks is often to promote and publicize their cause typically through:
  - Website defacement
  - Denial of service attacks
  - Theft and distribution of data that results in negative publicity or compromise of their targets

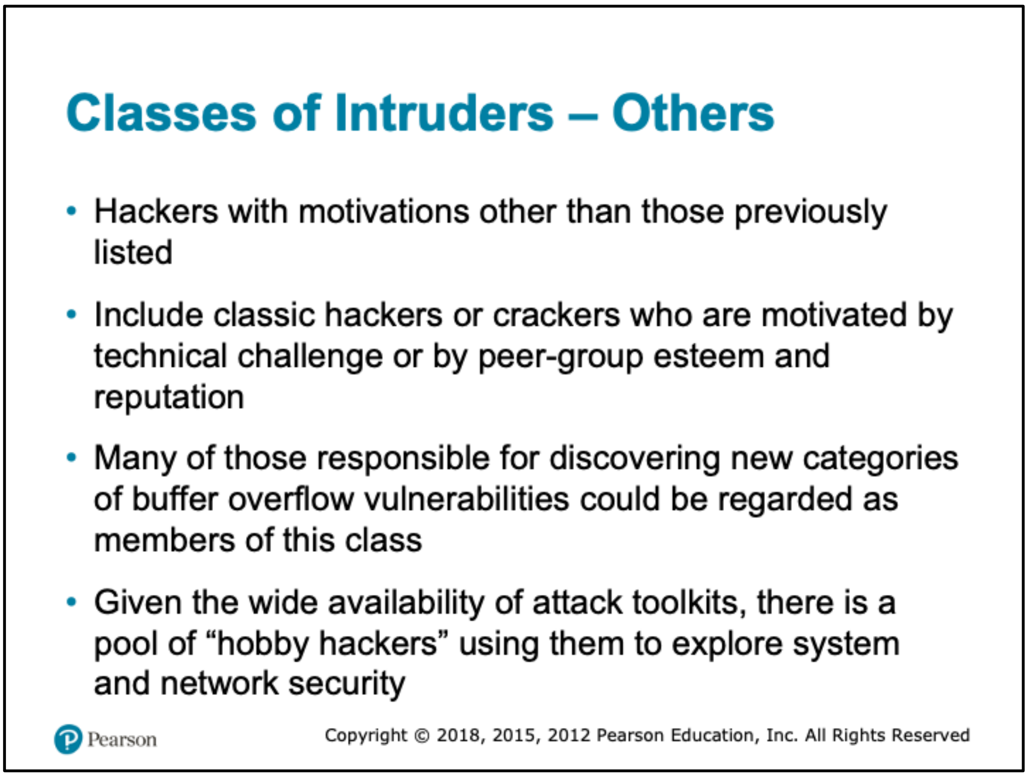
• **Activists**: Are either individuals, usually working as insiders, or members of a larger group of outsider attackers, who are motivated by social or political causes. They are also known as hacktivists, and their skill level is often quite low. The aim of their attacks is often to promote and publicize their cause, typically through website defacement, denial of service attacks, or the theft and distribution of data that results in negative publicity or compromise of their targets. Well-known recent examples include the activities of the groups Anonymous and LulzSec, and the actions of Chelsea (born Bradley) Manning and Edward Snowden.



Classes of Intruders – State-Sponsored Organizations

- Groups of hackers sponsored by governments to conduct espionage or sabotage activities
- Also known as Advanced Persistent Threats (APTs) due to the covert nature and persistence over extended periods involved with any attacks in this class
- Widespread nature and scope of these activities by a wide range of countries from China to the USA, UK, and their intelligence allies

• **State-sponsored organizations**: Are groups of hackers sponsored by governments to conduct espionage or sabotage activities. They are also known as Advanced Persistent Threats (APTs), due to the covert nature and persistence over extended periods involved with many attacks in this class. Recent reports such as [MAND13], and information revealed by Edward Snowden, indicate the widespread nature and scope of these activities by a wide range of countries from China to the USA, UK, and their intelligence allies.

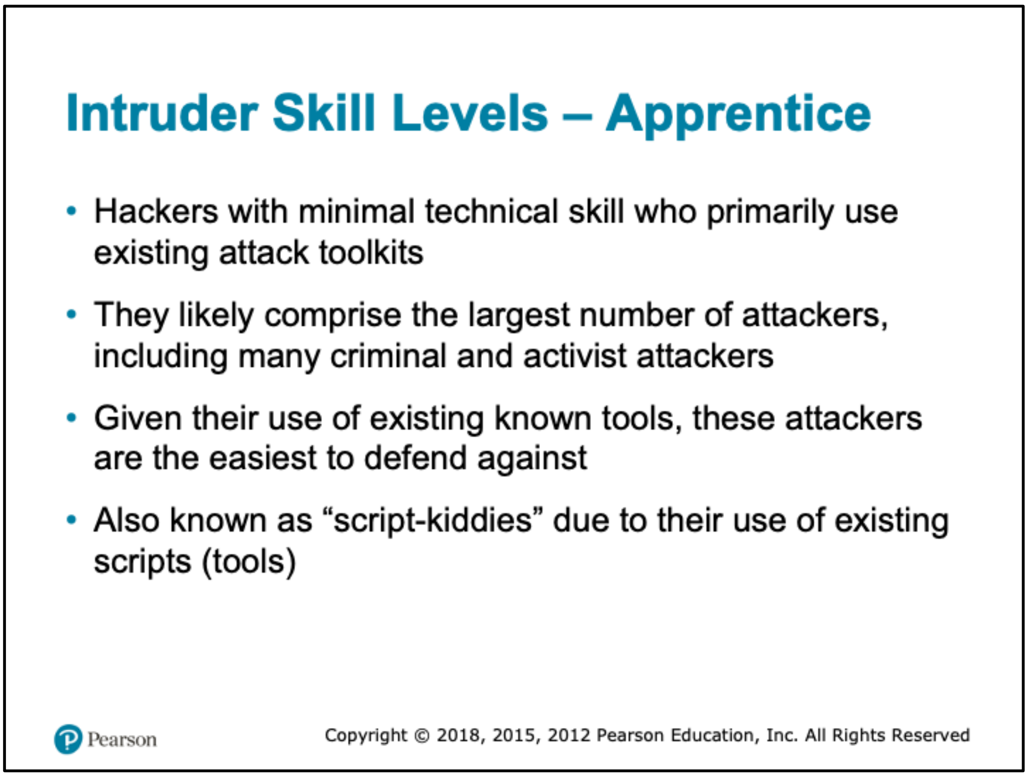


Classes of Intruders – Others

- Hackers with motivations other than those previously listed
- Include classic hackers or crackers who are motivated by technical challenge or by peer-group esteem and reputation
- Many of those responsible for discovering new categories of buffer overflow vulnerabilities could be regarded as members of this class
- Given the wide availability of attack toolkits, there is a pool of “hobby hackers” using them to explore system and network security

Pearson

• **Others**: Are hackers with motivations other than those listed above, including classic hackers or crackers who are motivated by technical challenge or by peer-group esteem and reputation. Many of those responsible for discovering new categories of buffer overflow vulnerabilities [MEER10] could be regarded as members of this class. Also, given the wide availability of attack toolkits, there is a pool of “hobby hackers” using them to explore system and network security, who could potentially become recruits for the above classes



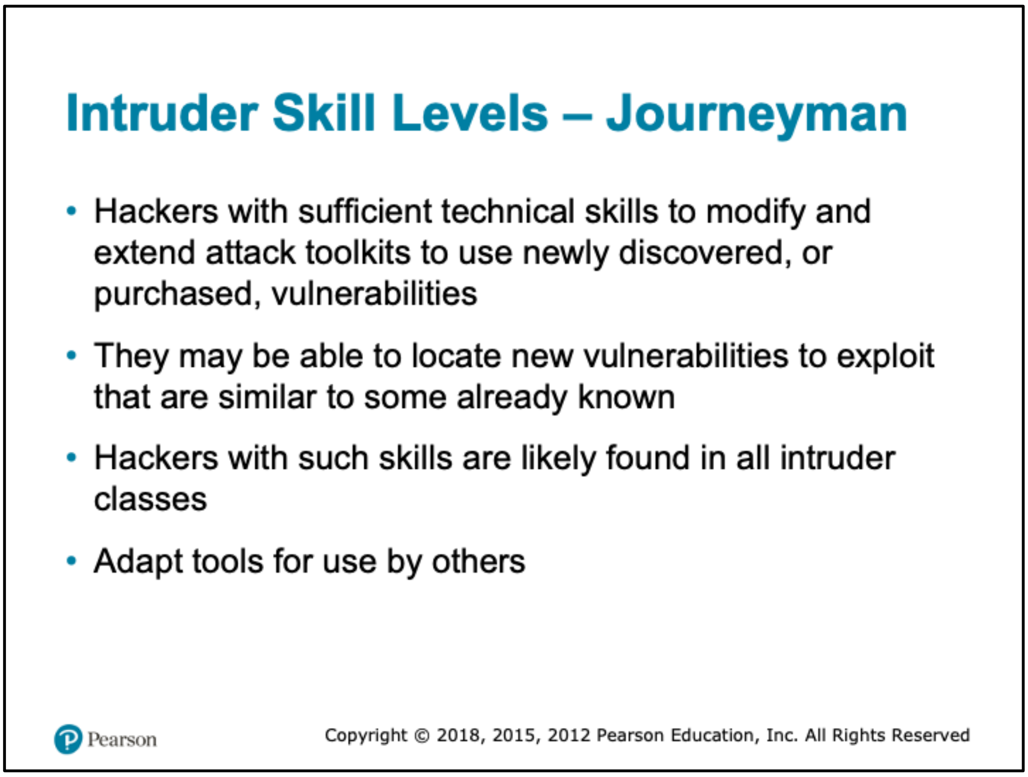
Intruder Skill Levels – Apprentice

- Hackers with minimal technical skill who primarily use existing attack toolkits
- They likely comprise the largest number of attackers, including many criminal and activist attackers
- Given their use of existing known tools, these attackers are the easiest to defend against
- Also known as “script-kiddies” due to their use of existing scripts (tools)

Pearson

Across these classes of intruders, there is also a range of skill levels seen. These can be broadly classified as:

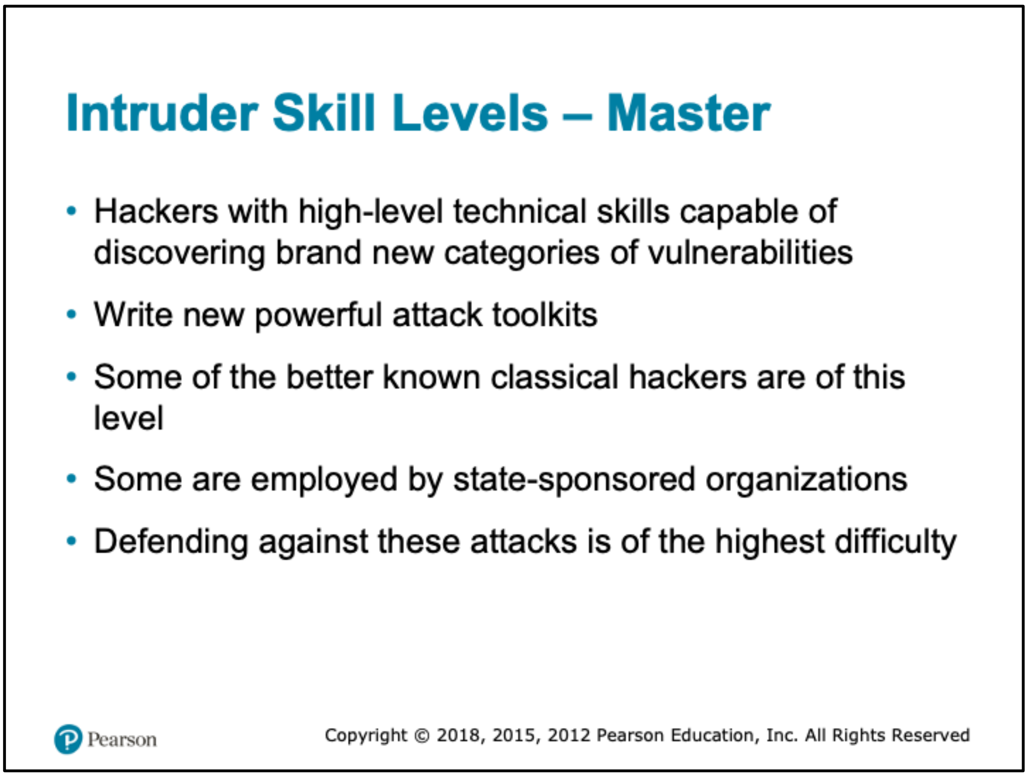
• **Apprentice**: Hackers with minimal technical skill who primarily use existing attack toolkits. They likely comprise the largest number of attackers, including many criminal and activist attackers. Given their use of existing known tools, these attackers are the easiest to defend against. They are also known as “script-kiddies” due to their use of existing scripts (tools).



Intruder Skill Levels – Journeyman

- Hackers with sufficient technical skills to modify and extend attack toolkits to use newly discovered, or purchased, vulnerabilities
- They may be able to locate new vulnerabilities to exploit that are similar to some already known
- Hackers with such skills are likely found in all intruder classes
- Adapt tools for use by others

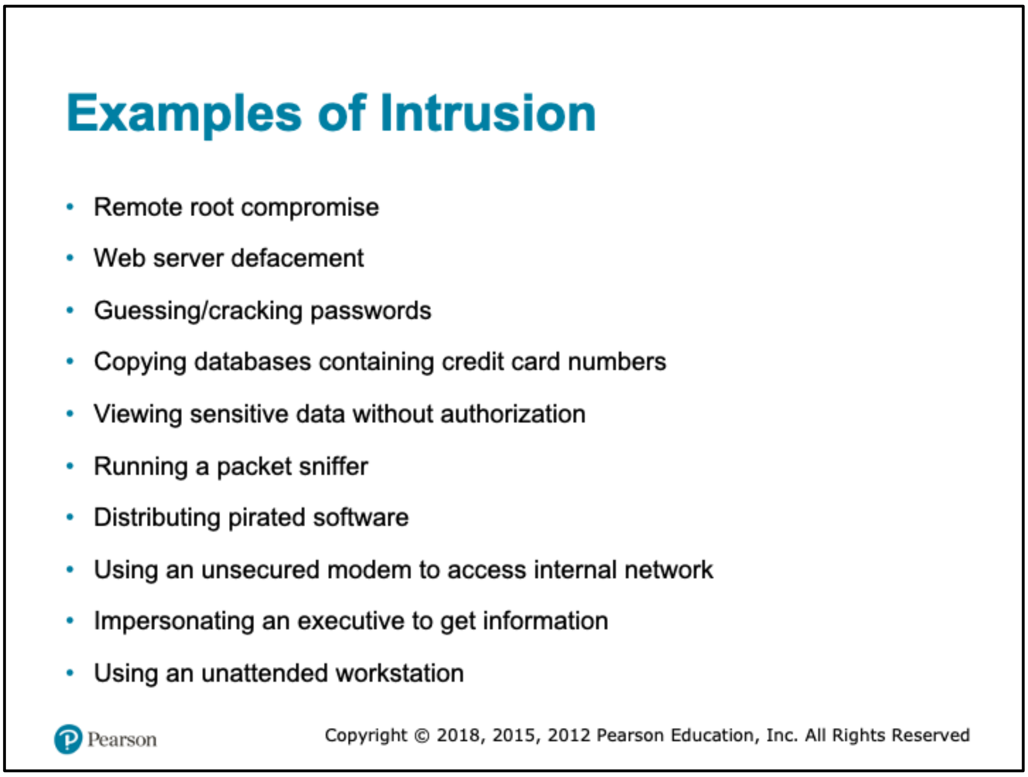
- **Journeyman**: Hackers with sufficient technical skills to modify and extend attack toolkits to use newly discovered, or purchased, vulnerabilities; or to focus on different target groups. They may also be able to locate new vulnerabilities to exploit that are similar to some already known. A number of hackers with such skills are likely found in all intruder classes listed above, adapting tools for use by others. The changes in attack tools make identifying and defending against such attacks harder.



Intruder Skill Levels – Master

- Hackers with high-level technical skills capable of discovering brand new categories of vulnerabilities
- Write new powerful attack toolkits
- Some of the better known classical hackers are of this level
- Some are employed by state-sponsored organizations
- Defending against these attacks is of the highest difficulty

Hackers with high-level technical skills capable of discovering brand new categories of vulnerabilities, or writing new powerful attack toolkits. Some of the better-known classical hackers are of this level, as clearly are some of those employed by some state-sponsored organizations, as the designation APT suggests. This makes defending against these attacks of the highest difficulty.



Examples of Intrusion

- Remote root compromise
- Web server defacement
- Guessing/cracking passwords
- Copying databases containing credit card numbers
- Viewing sensitive data without authorization
- Running a packet sniffer
- Distributing pirated software
- Using an unsecured modem to access internal network
- Impersonating an executive to get information
- Using an unattended workstation

Intruder attacks range from the benign to the serious. At the benign end of the scale, there are people who simply wish to explore internets and see what is out there. At the serious end are individuals or groups that attempt to read privileged data, perform unauthorized modifications to data, or disrupt systems.

NIST SP 800-61 (*Computer Security Incident Handling Guide* , August 2012) lists the following examples of intrusion:

- Performing a remote root compromise of an e-mail server

- Defacing a Web server

- Guessing and cracking passwords

- Copying a database containing credit card numbers

- Viewing sensitive data, including payroll records and medical information, without authorization

- Running a packet sniffer on a workstation to capture usernames and passwords

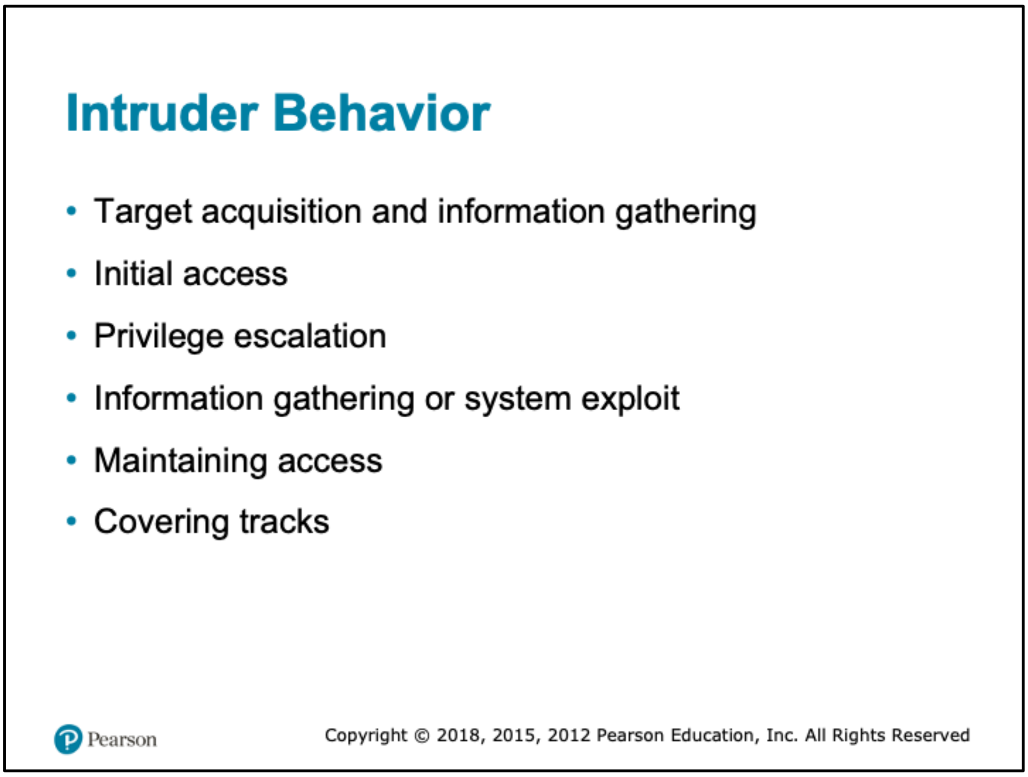
- Using a permission error on an anonymous FTP server to distribute pirated software and music files

- Dialing into an unsecured modem and gaining internal network access

- Posing as an executive, calling the help desk, resetting the executive’s e-mail password, and learning the new password

- Using an unattended, logged-in workstation without permission

Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs), of the type described in this chapter and Chapter 9, respectively, are designed to aid countering these types of threats. They can be reasonably effective against known, less sophisticated attacks, such as those by activist groups or large-scale email scams. They are likely less effective against the more sophisticated, targeted attacks by some criminal or state-sponsored intruders, since these attackers are more likely to use new, zero-day exploits, and to better obscure their activities on the targeted system. Hence they need to be part of a defense-in-depth strategy that may also include encryption of sensitive information, detailed audit trails, strong authentication and authorization controls, and active management of operating system and application security.



Intruder Behavior

- Target acquisition and information gathering
- Initial access
- Privilege escalation
- Information gathering or system exploit
- Maintaining access
- Covering tracks

Pearson

The techniques and behavior patterns of intruders are constantly shifting, to exploit newly discovered weaknesses and to evade detection and countermeasures. However, intruders typically use steps from a common attack methodology. [VERI16] in their “Wrap up” section illustrate a typical sequence of actions, starting with a phishing attack that results in the installation of malware that steals login credentials that eventually result in the compromise of a Point-of-Sale terminal. They note that while this is one specific incident scenario, the components are commonly seen in many attacks. [MCCL12] discuss in detail activities associated with the following steps:

• **Target Acquisition and Information Gathering**: Where the attacker identifies and characterizes the target systems using publicly available information, both technical and non-technical, and the use network exploration tools to map target resources.

Initial Access: The initial access to a target system, typically by exploiting a remote network vulnerability as we discuss in Chapters 10 and 11, by guessing weak authentication credentials used in a remote service as we discussed in Chapter 3, or via the installation of malware on the system using some form of social engineering or drive-by-download attack as we discuss in Chapter 6.

- **Privilege Escalation**: Actions taken on the system, typically via a local access vulnerability as discussed in Chapters 10 and 11, to increase the privileges available to the attacker to enable their desired goals on the target system.

- **Information Gathering or System Exploit:** Actions by the attacker to access or modify information or resources on the system, or to navigate to another target system.

- **Maintaining Access**: Actions such as the installation of backdoors or other malicious software as we discuss in Chapter 6, or through the addition of covert authentication credentials or other configuration changes to the system, to enable continued access by the attacker after the initial attack.

- **Covering Tracks:** Where the attacker disables or edits audit logs such as we discuss in Chapter 18, to remove evidence of attack activity, and uses rootkits and other measures to hide covertly installed files or code as we discuss in Chapter 6.

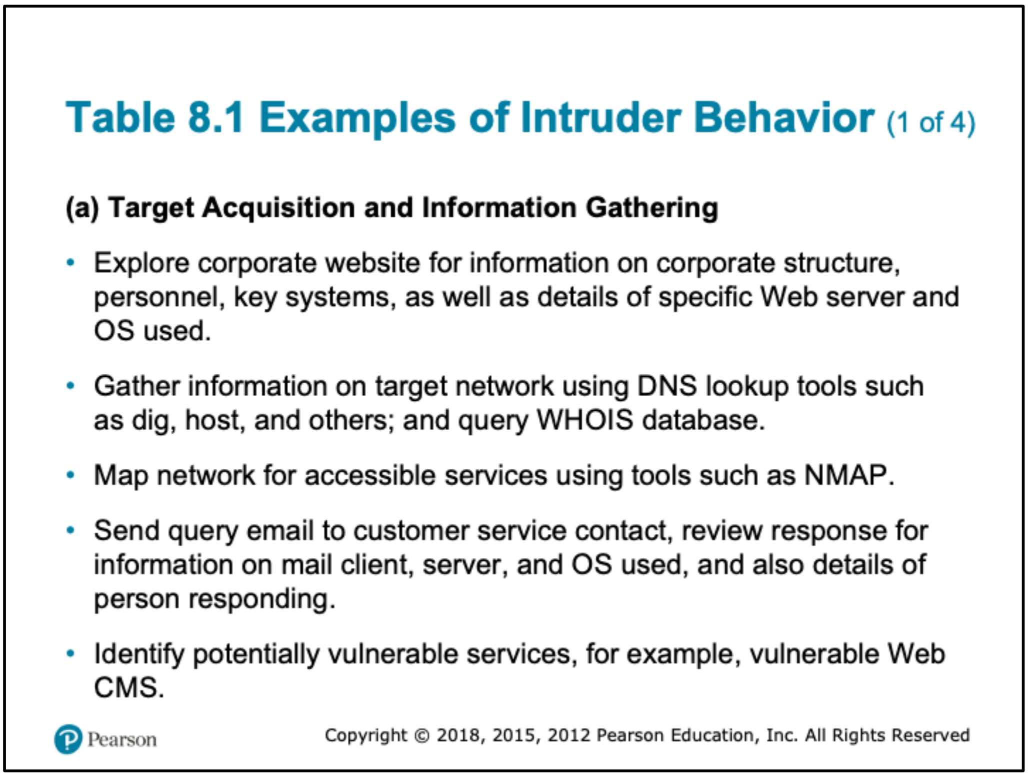


Table 8.1 Examples of Intruder Behavior (1 of 4)

(a) Target Acquisition and Information Gathering

- Explore corporate website for information on corporate structure, personnel, key systems, as well as details of specific Web server and OS used.
- Gather information on target network using DNS lookup tools such as dig, host, and others; and query WHOIS database.
- Map network for accessible services using tools such as NMAP.
- Send query email to customer service contact, review response for information on mail client, server, and OS used, and also details of person responding.
- Identify potentially vulnerable services, for example, vulnerable Web CMS.

Table 8.1 lists examples of activities associated with the above steps.

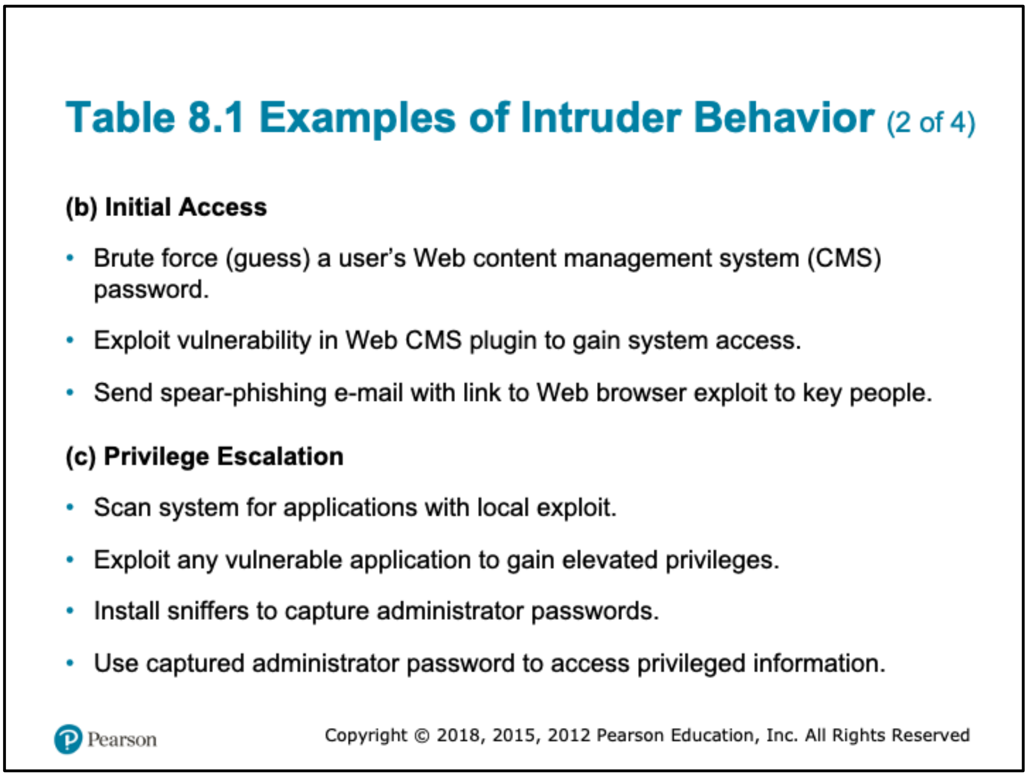


Table 8.1 Examples of Intruder Behavior (2 of 4)

(b) Initial Access

- Brute force (guess) a user's Web content management system (CMS) password.
- Exploit vulnerability in Web CMS plugin to gain system access.
- Send spear-phishing e-mail with link to Web browser exploit to key people.

(c) Privilege Escalation

- Scan system for applications with local exploit.
- Exploit any vulnerable application to gain elevated privileges.
- Install sniffers to capture administrator passwords.
- Use captured administrator password to access privileged information.

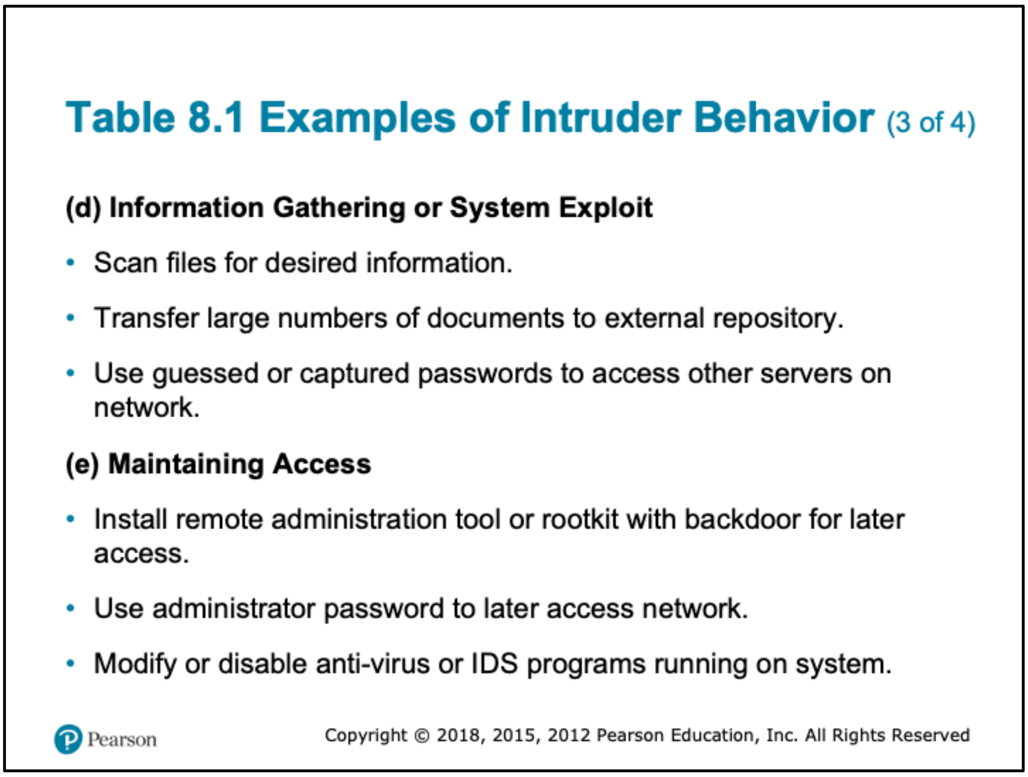


Table 8.1 Examples of Intruder Behavior (3 of 4)

(d) Information Gathering or System Exploit

- Scan files for desired information.
- Transfer large numbers of documents to external repository.
- Use guessed or captured passwords to access other servers on network.

(e) Maintaining Access

- Install remote administration tool or rootkit with backdoor for later access.
- Use administrator password to later access network.
- Modify or disable anti-virus or IDS programs running on system.

Pearson

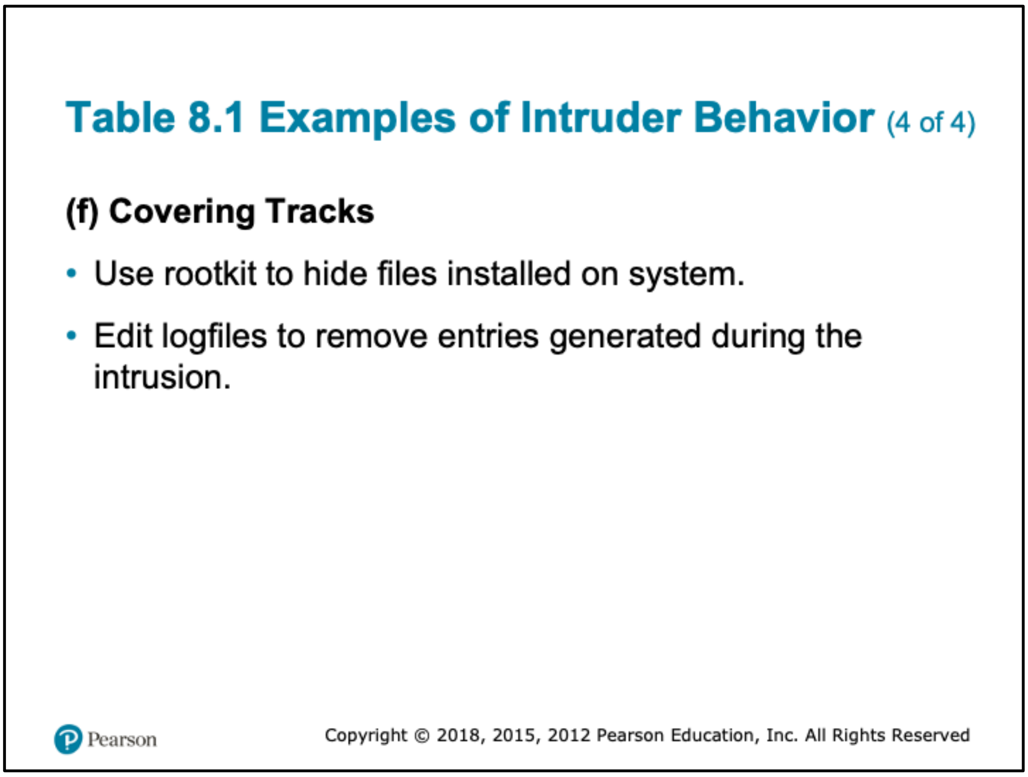
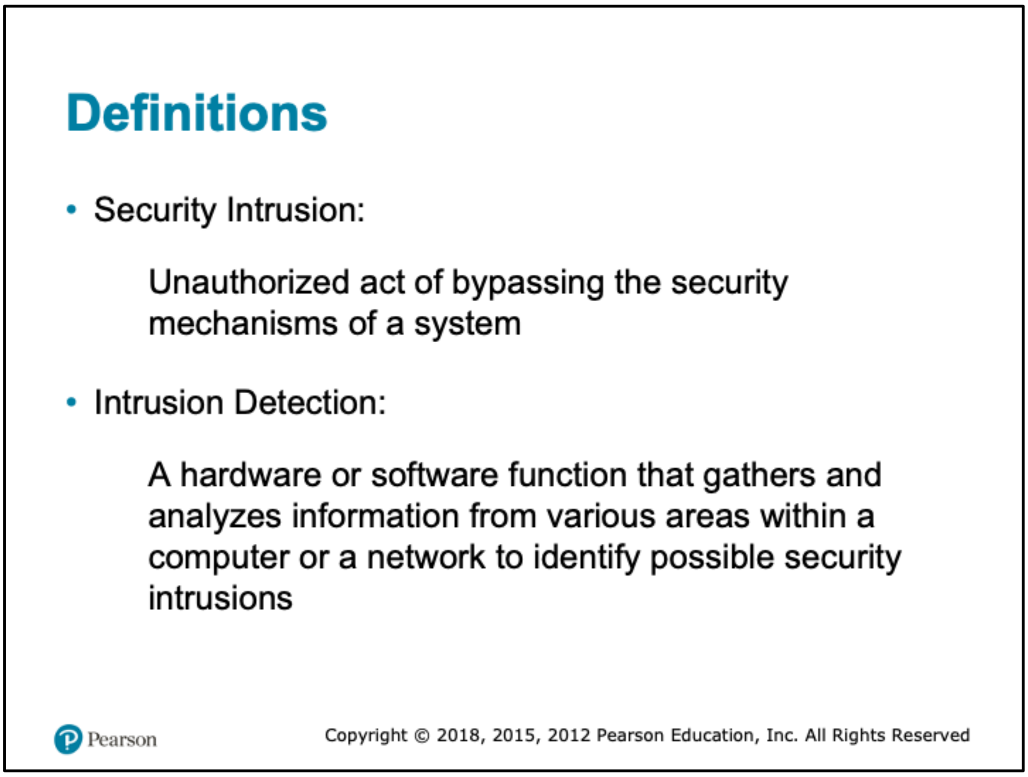


Table 8.1 Examples of Intruder Behavior (4 of 4)

(f) Covering Tracks

- Use rootkit to hide files installed on system.
- Edit logfiles to remove entries generated during the intrusion.



Definitions

- Security Intrusion:

  Unauthorized act of bypassing the security mechanisms of a system

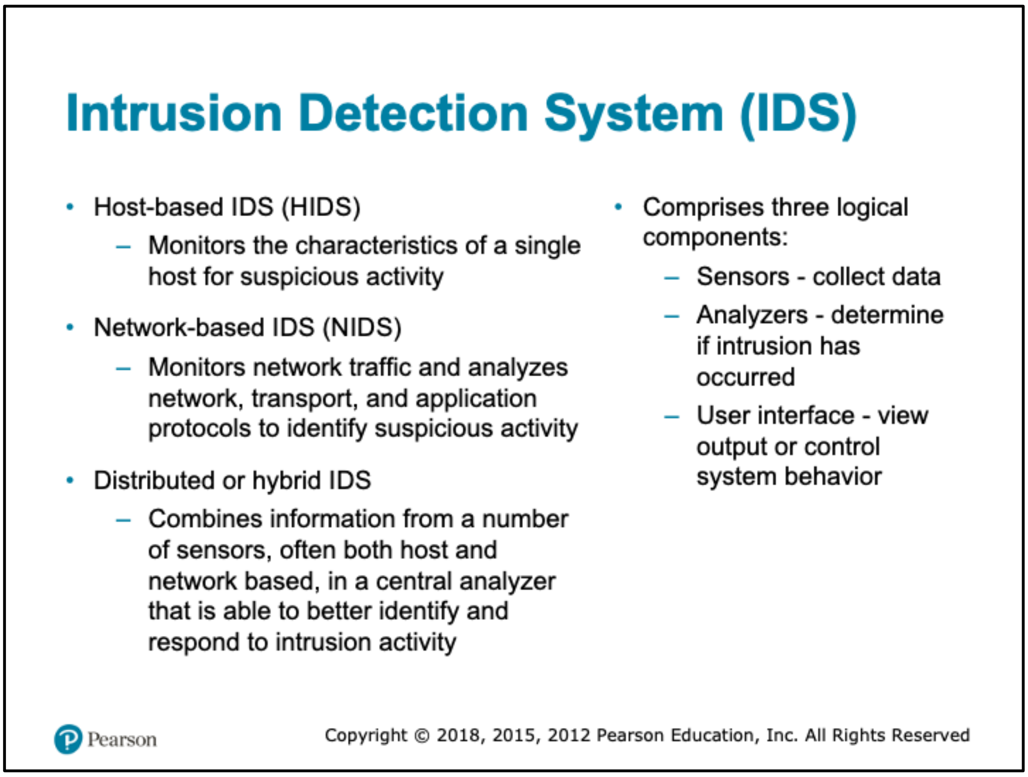
- Intrusion Detection:

  A hardware or software function that gathers and analyzes information from various areas within a computer or a network to identify possible security intrusions

Pearson

security intrusion: Unauthorized act of bypassing the security mechanisms of a system.

intrusion detection: A hardware or software function that gathers and analyzes information from various areas within a computer or a network to identify possible security intrusions.



Intrusion Detection System (IDS)

- Host-based IDS (HIDS)
  - Monitors the characteristics of a single host for suspicious activity
- Network-based IDS (NIDS)
  - Monitors network traffic and analyzes network, transport, and application protocols to identify suspicious activity
- Distributed or hybrid IDS
  - Combines information from a number of sensors, often both host and network based, in a central analyzer that is able to better identify and respond to intrusion activity

- Comprises three logical components:
  - Sensors - collect data
  - Analyzers - determine if intrusion has occurred
  - User interface - view output or control system behavior

Pearson

An IDS comprises three logical components:

• **Sensors** : Sensors are responsible for collecting data. The input for a sensor may be any part of a system that could contain evidence of an intrusion. Types of input to a sensor includes network packets, log files, and system call traces. Sensors collect and forward this information to the analyzer.

• **Analyzers :** Analyzers receive input from one or more sensors or from other analyzers. The analyzer is responsible for determining if an intrusion has occurred. The output of this component is an indication that an intrusion has occurred. The output may include evidence supporting the conclusion that an intrusion occurred. The analyzer may provide guidance about what actions to take as a result of the intrusion. The sensor inputs may also be stored for future analysis and review in a storage or database component.

• **User interface**: The user interface to an IDS enables a user to view output from the system or control the behavior of the system. In some systems, the user interface may equate to a manager, director, or console component.

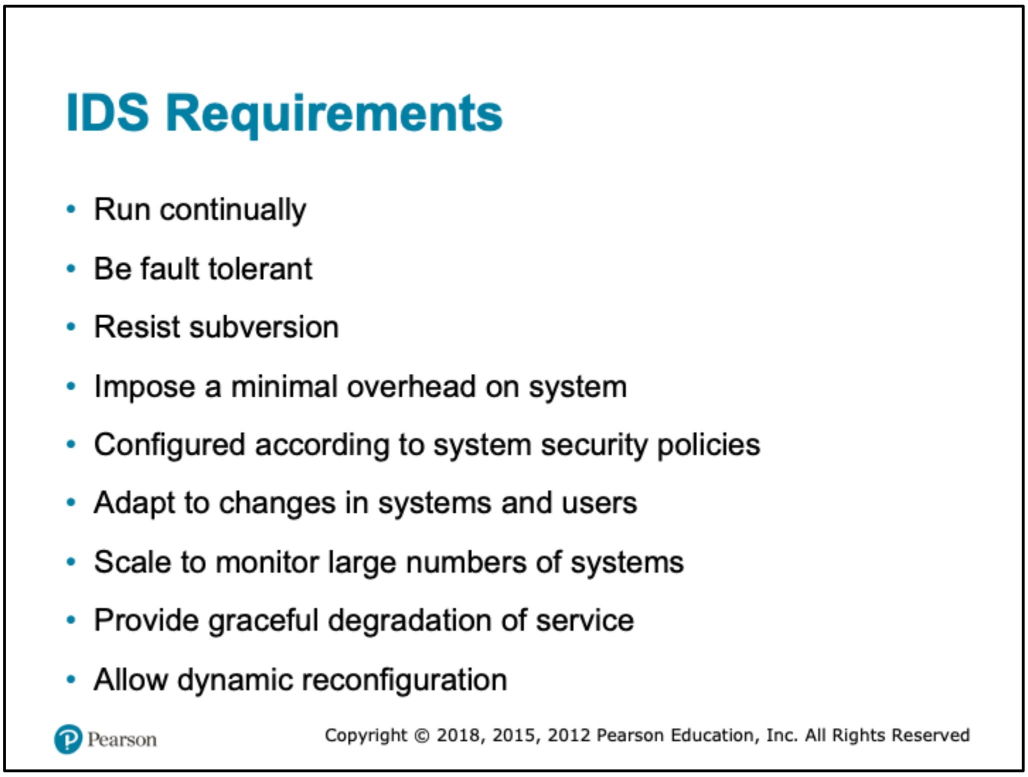
An IDS may use a single sensor and analyzer, such as a classic HIDS on a host or NIDS in a firewall device. More sophisticated IDSs can use multiple sensors, across a range of host and network devices, sending information to a centralized analyzer and user interface in a distributed architecture.

IDSs are often classified based on the source and type of data analyzed, as:

• **Host-based IDS (HIDS):** Monitors the characteristics of a single host and the events occurring within that host, such as process identifiers and the system calls they make, for evidence of suspicious activity.

• **Network-based IDS (NIDS):** Monitors network traffic for particular network segments or devices and analyzes network, transport, and application protocols to identify suspicious activity.

• **Distributed or hybrid IDS:** Combines information from a number of sensors, often both host and network-based, in a central analyzer that is able to better identify and respond to intrusion activity.



IDS Requirements

- Run continually
- Be fault tolerant
- Resist subversion
- Impose a minimal overhead on system
- Configured according to system security policies
- Adapt to changes in systems and users
- Scale to monitor large numbers of systems
- Provide graceful degradation of service
- Allow dynamic reconfiguration

To be of practical use, an IDS should detect a substantial percentage of intrusions while keeping the false alarm rate at an acceptable level. If only a modest percentage of actual intrusions are detected, the system provides a false sense of security. On the other hand, if the system frequently triggers an alert when there is no intrusion (a false alarm), then either system managers will begin to ignore the alarms, or much time will be wasted analyzing the false alarms.

Unfortunately, because of the nature of the probabilities involved, it is very difficult to meet the standard of high rate of detections with a low rate of false alarms. In general, if the actual numbers of intrusions is low compared to the number of legitimate uses of a system, then the false alarm rate will be high unless the test is extremely discriminating. This is an example of a phenomenon known as the base-rate fallacy . A study of existing IDSs, reported in [AXEL00], indicated that current systems have not overcome the problem of the base-rate fallacy. See Appendix I for a brief background on the mathematics of this problem.

[BALA98] lists the following as desirable for an IDS. It must

• Run continually with minimal human supervision.

• Be fault tolerant in the sense that it must be able to recover from system crashes and reinitializations.

• Resist subversion. The IDS must be able to monitor itself and detect if it has been modified by an attacker.

• Impose a minimal overhead on the system where it is running.

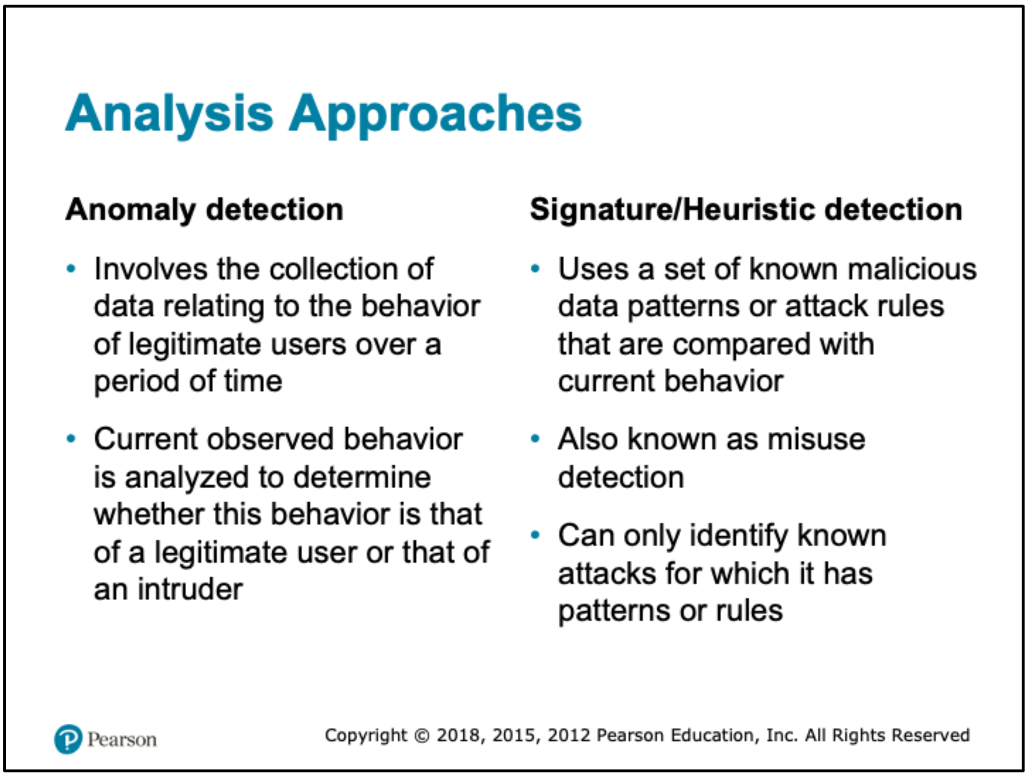
• Be able to be configured according to the security policies of the system that is being monitored.

• Be able to adapt to changes in system and user behavior over time.

• Be able to scale to monitor a large number of hosts.

• Provide graceful degradation of service in the sense that if some components of the IDS stop working for any reason, the rest of them should be affected as little as possible.

• Allow dynamic reconfiguration; that is, the ability to reconfigure the IDS without having to restart it.



Analysis Approaches

Anomaly detection

- Involves the collection of data relating to the behavior of legitimate users over a period of time
- Current observed behavior is analyzed to determine whether this behavior is that of a legitimate user or that of an intruder

Signature/Heuristic detection

- Uses a set of known malicious data patterns or attack rules that are compared with current behavior
- Also known as misuse detection
- Can only identify known attacks for which it has patterns or rules

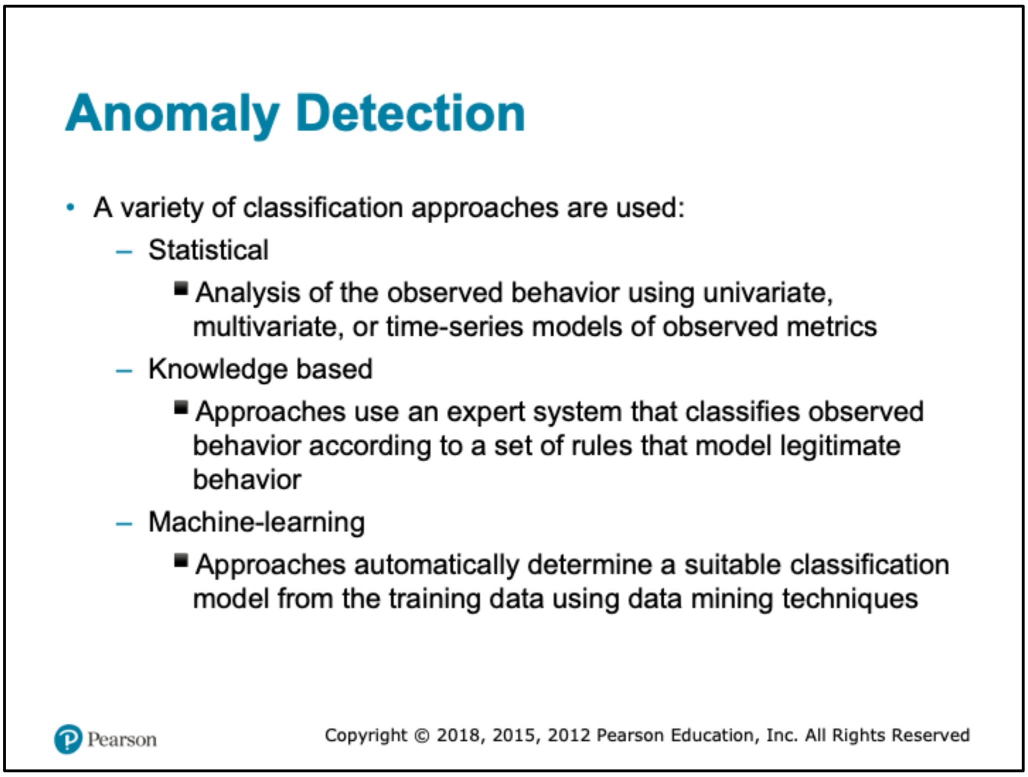
Pearson

IDSs typically use one of the following alternative approaches to analyze sensor data to detect intrusions:

1. **Anomaly detection**: Involves the collection of data relating to the behavior of legitimate users over a period of time. Then current observed behavior is analyzed to determine with a high level of confidence whether this behavior is that of a legitimate user or alternatively that of an intruder.

2. **Signature or Heuristic detection**: Uses a set of known malicious data patterns (signatures) or attack rules (heuristics) that are compared with current behavior to decide if is that of an intruder. It is also known as misuse detection. This approach can only identify known attacks for which it has patterns or rules.

In essence, anomaly approaches aim to define normal, or expected, behavior, in order to identify malicious or unauthorized behavior. Signature or heuristic-based approaches directly define malicious or unauthorized behavior. They can quickly and efficiently identify known attacks. However only anomaly detection is able to detect unknown, zero-day attacks, as it starts with known good behavior and identifies anomalies to it. Given this advantage, clearly anomaly detection would be the preferred approach, were it not for the difficulty in collecting and analyzing the data required, and the high level of false alarms, as we discuss in the following sections.



Anomaly Detection

- A variety of classification approaches are used:
  - Statistical
    - Analysis of the observed behavior using univariate, multivariate, or time-series models of observed metrics
  - Knowledge based
    - Approaches use an expert system that classifies observed behavior according to a set of rules that model legitimate behavior
  - Machine-learning
    - Approaches automatically determine a suitable classification model from the training data using data mining techniques

Pearson

The anomaly detection approach involves first developing a model of legitimate user behavior by collecting and processing sensor data from the normal operation of the monitored system in a training phase. This may occur at distinct times, or there may be a continuous process of monitoring and evolving the model over time. Once this model exists, current observed behavior is compared with the model in order to classify it as either legitimate or anomalous activity in a detection phase.

A variety of classification approaches are used, which [GARC09] broadly categorized as:

• **Statistical:** Analysis of the observed behavior using univariate, multivariate, or time-series models of observed metrics.

• **Knowledge based:** Approaches use an expert system that classifies observed behavior according to a set of rules that model legitimate behavior.

• **Machine-learning**: Approaches automatically determine a suitable classification model from the training data using data mining techniques.

They also note two key issues that affect the relative performance of these alternatives, being the efficiency and cost of the detection process.

The monitored data is first parameterized into desired standard metrics that will then be analyzed. This step ensures that data gathered from a variety of possible sources is provided in standard form for analysis.

Statistical approaches use the captured sensor data to develop a statistical profile of the observed metrics. The earliest approaches used univariate models, where each metric was treated as an independent random variable. However this was too crude to effectively identify intruder behavior. Later, multivariate models considered correlations between the metrics, which better levels of discrimination observed. Time-series models use the order and time between observed events to better classify the behavior. The advantages of these statistical approaches include their relative simplicity and low computation cost, and lack of assumptions about behavior expected. Their disadvantages include the difficulty in selecting suitable metrics to obtain a reasonable balance between false positives and false negatives, and that not all behaviors can be modeled using these approaches.

Knowledge based approaches classify the observed data using a set of rules. These rules are developed during the training phase, usually manually, to characterize the observed training data into distinct classes. Formal tools may be used to describe these rules, such as a finite-state machine or a standard description language. They are then used to classify the observed data in the detection phase. The advantages of knowledge-based approaches include their robustness and flexibility. Their main disadvantage is the difficulty and time required to develop high-quality knowledge from the data, and the need for human experts to assist with this process.

Machine-learning approaches use data mining techniques to automatically develop a model using the labeled normal training data. This model is then able to classify subsequently observed data as either normal or anomalous. A key disadvantage is that this process typically requires significant time and computational resources. Once the model is generated however, subsequent analysis is generally fairly efficient.

A variety of machine-learning approaches have been tried, with varying success. These include:

- **Bayesian networks**: Encode probabilistic relationships among observed metrics.

- **Markov models**: Develop a model with sets of states, some possibly hidden, interconnected by transition probabilities.

- **Neural networks**: Simulate human brain operation with neurons and synapse between them, that classify observed data.

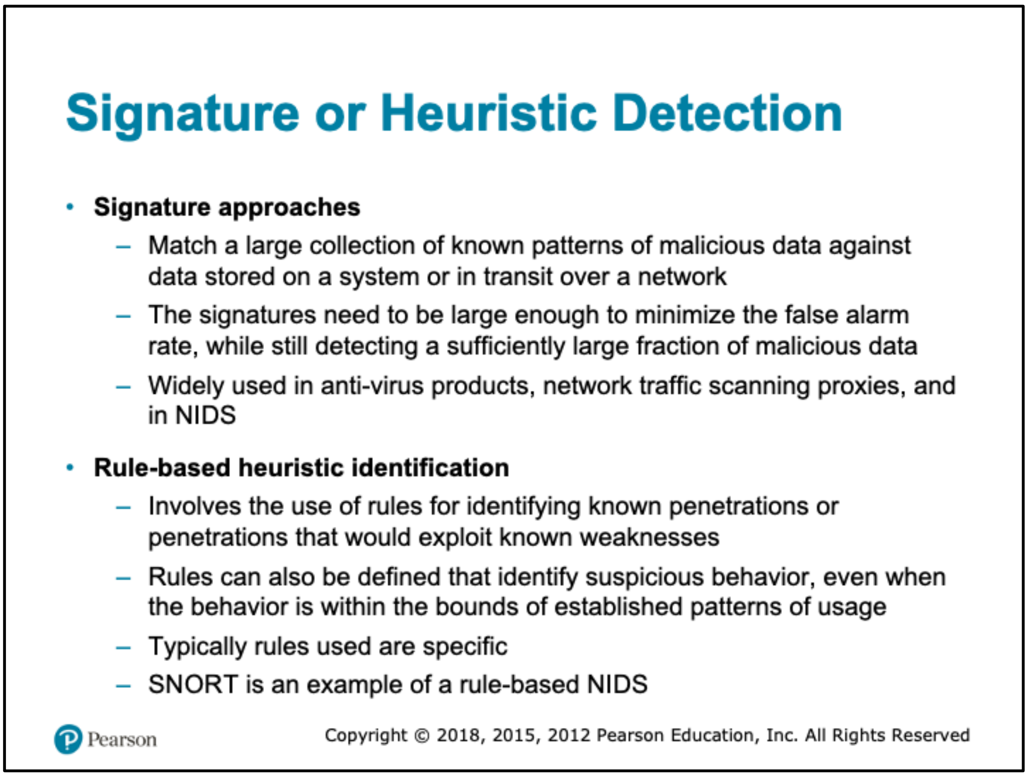
- **Fuzzy logic:** Uses fuzzy set theory where reasoning is approximate, and can accommodate uncertainty.

- **Genetic algorithms:** Uses techniques inspired by evolutionary biology, including inheritance, mutation, selection and recombination, to develop classification rules.

- **Clustering and outlier detection:** Group the observed data into clusters based on some similarity or distance measure, and then identify subsequent data as either belonging to a cluster or as an outlier.

The advantages of the machine-learning approaches include their flexibility, adaptability, and ability to capture interdependencies between the observed metrics. Their disadvantages include their dependency on assumptions about accepted behavior for a system, their currently unacceptably high false alarm rate, and their high resource cost.

A key limitation of anomaly detection approaches used by IDSs, particularly the machine-learning approaches, is that they are generally only trained with legitimate data, unlike many of the other applications surveyed in [CHAN09] where both legitimate and anomalous training data is used. The lack of anomalous training data, which occurs given the desire to detect currently unknown future attacks, limits the effectiveness of some of the techniques listed above.



Signature or Heuristic Detection

- **Signature approaches**
  - Match a large collection of known patterns of malicious data against data stored on a system or in transit over a network
  - The signatures need to be large enough to minimize the false alarm rate, while still detecting a sufficiently large fraction of malicious data
  - Widely used in anti-virus products, network traffic scanning proxies, and in NIDS
- **Rule-based heuristic identification**
  - Involves the use of rules for identifying known penetrations or penetrations that would exploit known weaknesses
  - Rules can also be defined that identify suspicious behavior, even when the behavior is within the bounds of established patterns of usage
  - Typically rules used are specific
  - SNORT is an example of a rule-based NIDS

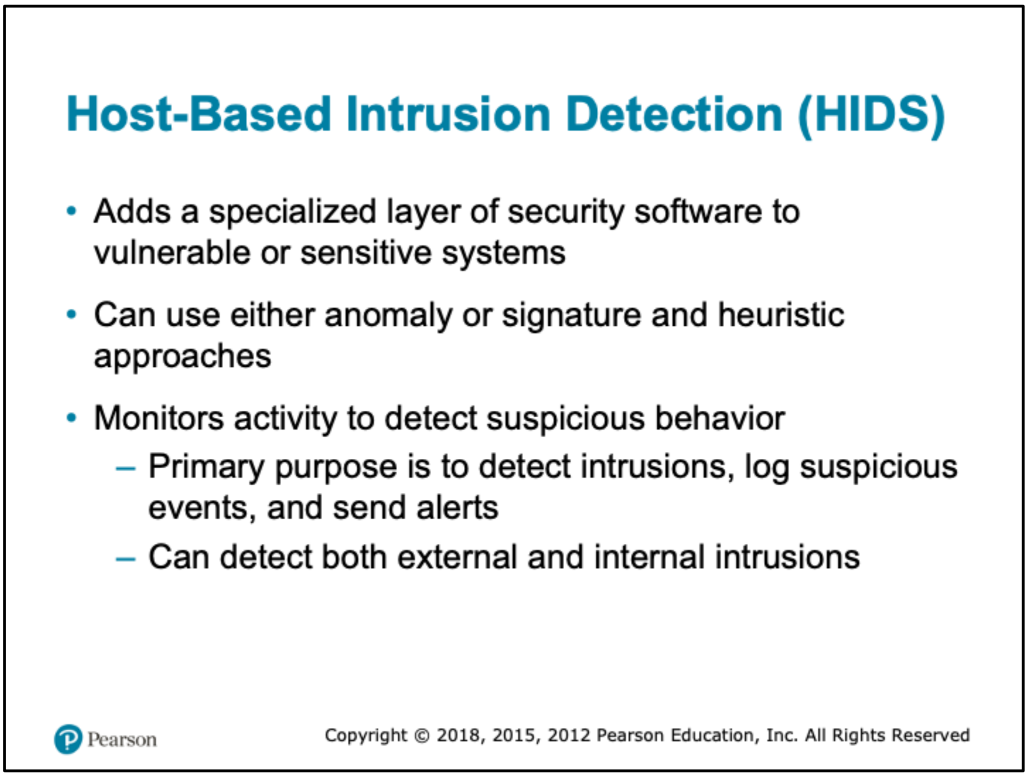
Signature or heuristic techniques detect intrusion by observing events in the system and applying either a set of signature patterns to the data, or a set of rules that characterize the data, leading to a decision regarding whether the observed data indicates normal or anomalous behavior.

Signature approaches match a large collection of known patterns of malicious data against data stored on a system or in transit over a network. The signatures need to be large enough to minimize the false alarm rate, while still detecting a sufficiently large fraction of malicious data. This approach is widely used in antivirus products, in network traffic scanning proxies, and in NIDS. The advantages of this approach include the relatively low cost in time and resource use, and its wide acceptance. Disadvantages include the significant effort required to constantly identify and review new malware to create signatures able to identify it, and the inability to detect zero-day attacks for which no signatures exist.

Rule-based heuristic identification involves the use of rules for identifying known penetrations or penetrations that would exploit known weaknesses. Rules can also be defined that identify suspicious behavior, even when the behavior is within the bounds of established patterns of usage. Typically, the rules used in these systems are specific to the machine and operating system.

The most fruitful approach to developing such rules is to analyze attack tools and scripts collected on the Internet. These rules can be supplemented with rules generated by knowledgeable security personnel. In this latter case, the normal procedure is to interview system administrators and security analysts to collect a suite of known penetration scenarios and key events that threaten the security of the target system.

The SNORT system, which we discuss later in Section 8.9 is an example of a rule-based NIDS. A large collection of rules exists for it to detect a wide variety of network attacks.



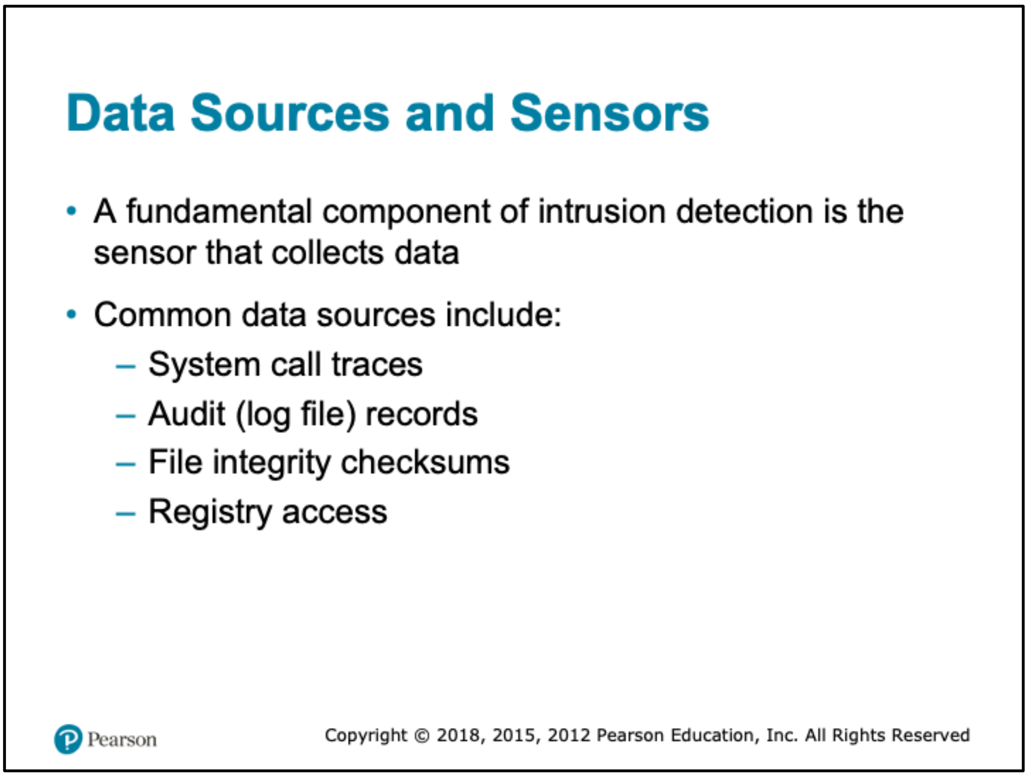
Host-Based Intrusion Detection (HIDS)

- Adds a specialized layer of security software to vulnerable or sensitive systems
- Can use either anomaly or signature and heuristic approaches
- Monitors activity to detect suspicious behavior
  - Primary purpose is to detect intrusions, log suspicious events, and send alerts
  - Can detect both external and internal intrusions

Pearson

Host-based IDSs (HIDSs) add a specialized layer of security software to vulnerable or sensitive systems; such as database servers and administrative systems. The HIDS monitors activity on the system in a variety of ways to detect suspicious behavior. In some cases, an IDS can halt an attack before any damage is done, as we discuss in Section 9.6, but its main purpose is to detect intrusions, log suspicious events, and send alerts.

The primary benefit of a HIDS is that it can detect both external and internal intrusions, something that is not possible either with network-based IDSs or firewalls. As we discuss in the previous section, host-based IDSs can use either anomaly or signature and heuristic approaches to detect unauthorized behavior on the monitored host. We now review some common data sources and sensors used in HIDS, and then continue with a discussion of how the anomaly, signature and heuristic approaches are used in HIDS, and then consider distributed HIDS.



Data Sources and Sensors

- A fundamental component of intrusion detection is the sensor that collects data
- Common data sources include:
  - System call traces
  - Audit (log file) records
  - File integrity checksums
  - Registry access

As noted previously, a fundamental component of intrusion detection is the sensor that collects data. Some record of ongoing activity by users must be provided as input to the analysis component of the IDS. Common data sources include:

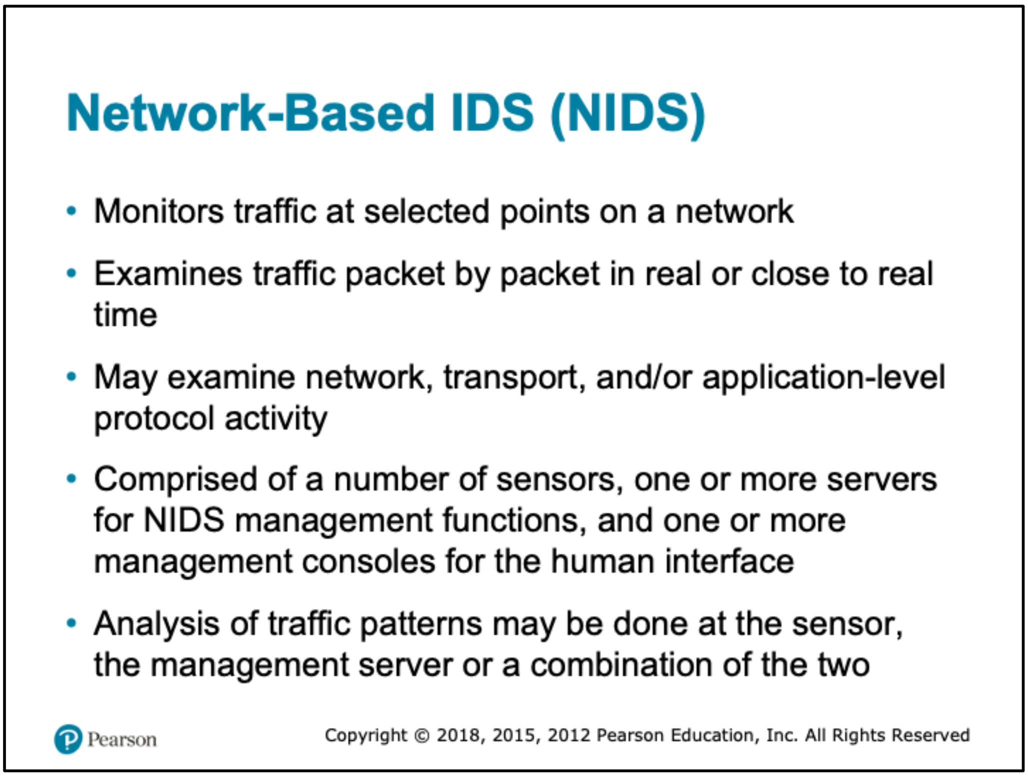
• **System call traces**: A record of the sequence of systems calls by processes on a system, is widely acknowledged as the preferred data source for HIDS since the pioneering work of Forrest [CREE13]. While these work well on Unix and Linux systems, they are problematic on Windows systems due to the extensive use of DLLs that obscure which processes use specific system calls.

• **Audit (log file) records:** Most modern operating systems include accounting software that collects information on user activity. The advantage of using this information is that no additional collection software is needed. The disadvantages are that the audit records may not contain the needed information or may not contain it in a convenient form, and that intruders may attempt to manipulate these records to hide their actions.

• **File integrity checksums**: A common approach to detecting intruder activity on a system is to periodically scan critical files for changes from the desired baseline, by comparing a current cryptographic checksums for these files, with a record of known good values. Disadvantages include the need to generate and protect the checksums using known good files, and the difficulty monitoring changing files. Tripwire is a well-known system using this approach.

• **Registry access**: An approach used on Windows systems is to monitor access to the registry, given the amount of information and access to it used by programs on these systems. However this source is very Windows specific, and has recorded limited success.

The sensor gathers data from the chosen source, filters the gathered data to remove any unwanted information and to standardize the information format, and forwards the result to the IDS analyzer, which may be local or remote.



Network-Based IDS (NIDS)

- Monitors traffic at selected points on a network
- Examines traffic packet by packet in real or close to real time
- May examine network, transport, and/or application-level protocol activity
- Comprised of a number of sensors, one or more servers for NIDS management functions, and one or more management consoles for the human interface
- Analysis of traffic patterns may be done at the sensor, the management server or a combination of the two

A network-based IDS (NIDS) monitors traffic at selected points on a network or interconnected set of networks. The NIDS examines the traffic packet by packet in real time, or close to real time, to attempt to detect intrusion patterns. The NIDS may examine network-, transport-, and/or application-level protocol activity. Note the contrast with a host-based IDS; a NIDS examines packet traffic directed toward potentially vulnerable computer systems on a network. A host-based system examines user and software activity on a host.

A typical NIDS facility includes a number of sensors to monitor packet traffic, one or more servers for NIDS management functions, and one or more management consoles for the human interface. The analysis of traffic patterns to detect intrusions may be done at the sensor, at the management server, or some combination of the two.

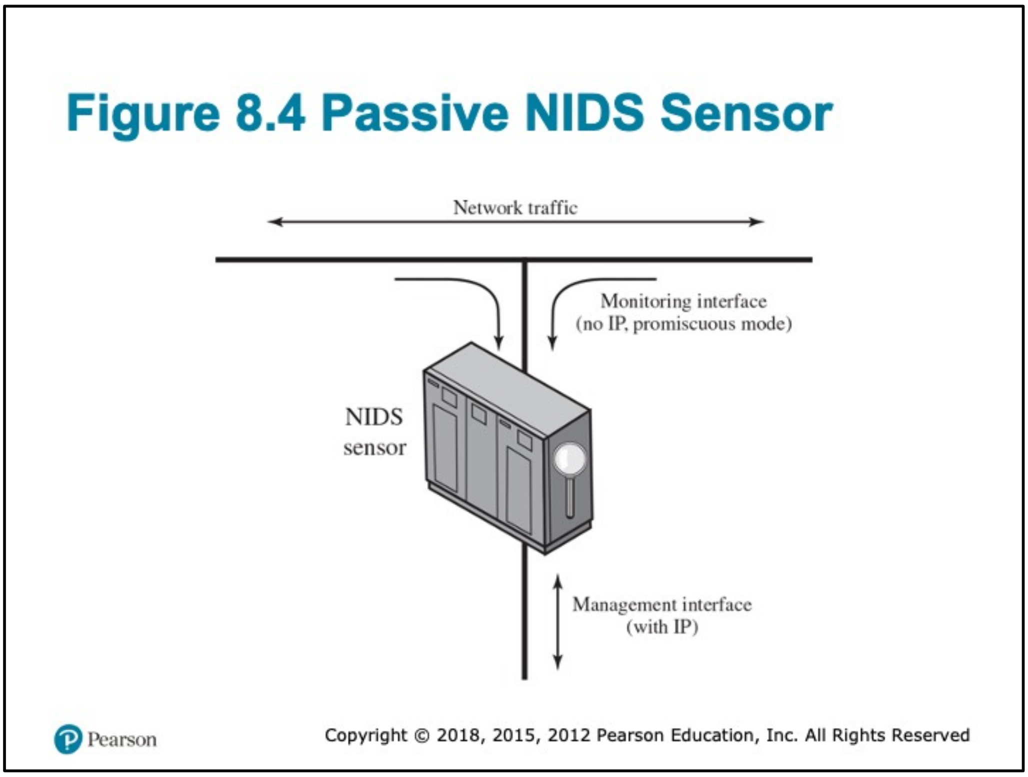


Figure 8.4 Passive NIDS Sensor

Pearson

Sensors can be deployed in one of two modes: inline and passive. An inline sensor is inserted into a network segment so that the traffic that it is monitoring must pass through the sensor. One way to achieve an inline sensor is to combine NIDS sensor logic with another network device, such as a firewall or a LAN switch. This approach has the advantage that no additional separate hardware devices are needed; all that is required is NIDS sensor software. An alternative is a stand-alone inline NIDS sensor. The primary motivation for the use of inline sensors is to enable them to block an attack when one is detected. In this case the device is performing both intrusion detection and intrusion prevention functions.

More commonly, passive sensors are used. A passive sensor monitors a copy of network traffic; the actual traffic does not pass through the device. From the point of view of traffic flow, the passive sensor is more efficient than the inline sensor, because it does not add an extra handling step that contributes to packet delay.

Figure 8.4 illustrates a typical passive sensor configuration. The sensor connects to the network transmission medium, such as a fiber optic cable, by a direct physical tap. The tap provides the sensor with a copy of all network traffic being carried by the medium. The network interface card (NIC) for this tap usually does not have an IP address configured for it. All traffic into this NIC is simply collected with no protocol interaction with the network. The sensor has a second NIC that connects to the network with an IP address and enables the sensor to communicate with a NIDS management server.

The sensor connects to the network transmission medium by a direct physical tap. The tap provides the sensor with a copy of all network traffic being carried by the medium. The monitoring interface for this tap with no I P, promiscuous mode configured for N I D S sensor. The sensor has a management interface that connects to the network with an IP address and enables the sensor to communicate with a N I D S management server.

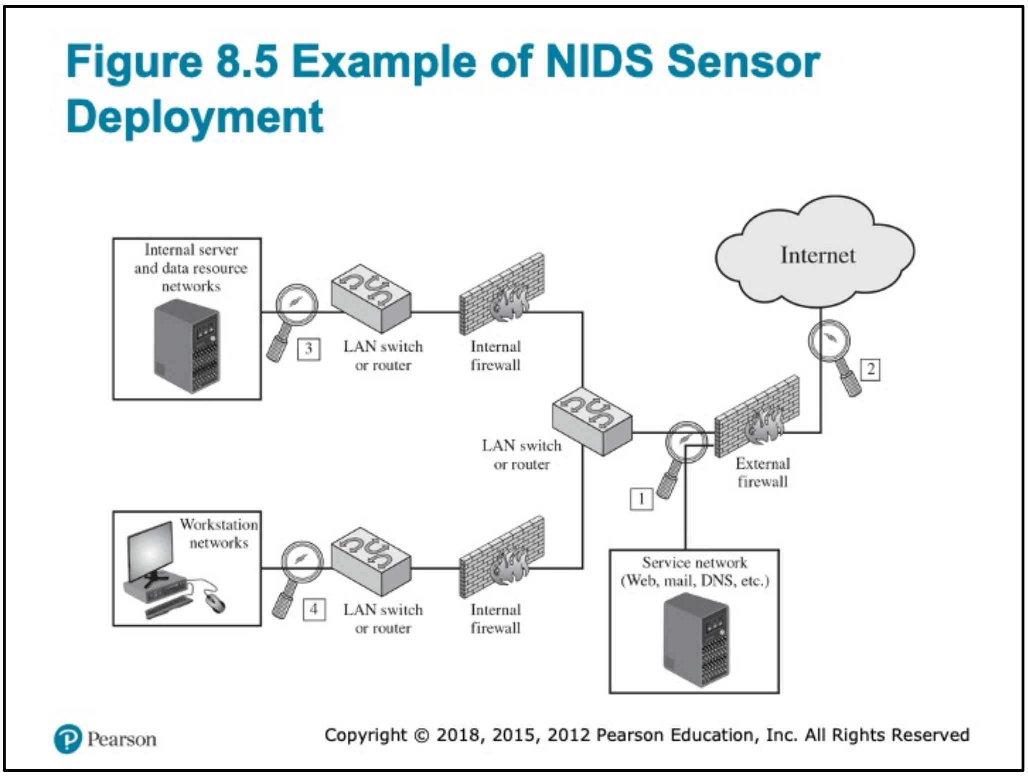


Figure 8.5 Example of NIDS Sensor Deployment

Consider an organization with multiple sites, each of which has one or more LANs, with all of the networks interconnected via the Internet or some other WAN technology. For a comprehensive NIDS strategy, one or more sensors are needed at each site. Within a single site, a key decision for the security administrator is the placement of the sensors.

Figure 8.5 illustrates a number of possibilities. In general terms, this configuration is typical of larger organizations. All Internet traffic passes through an external firewall that protects the entire facility. Traffic from the outside world, such as customers and vendors that need access to public services, such as Web and mail, is monitored. The external firewall also provides a degree of protection for those parts of the network that should only be accessible by users from other corporate sites. Internal firewalls may also be used to provide more specific protection to certain parts of the network.

A common location for a NIDS sensor is just inside the external firewall (location 1 in the figure). This position has a number of advantages:

• Sees attacks, originating from the outside world, that penetrate the network's perimeter defenses (external firewall).

- Highlights problems with the network firewall policy or performance.

- Sees attacks that might target the Web server or ftp server.

- Even if the incoming attack is not recognized, the IDS can sometimes recognize the outgoing traffic that results from the compromised server.

Instead of placing a NIDS sensor inside the external firewall, the security administrator may choose to place a NIDS sensor between the external firewall and the Internet or WAN (location 2). In this position, the sensor can monitor all network traffic, unfiltered. The advantages of this approach are as follows:

- Documents number of attacks originating on the Internet that target the network

- Documents types of attacks originating on the Internet that target the network

A sensor at location 2 has a higher processing burden than any sensor located elsewhere on the site network.

In addition to a sensor at the boundary of the network, on either side of the external firewall, the administrator may configure a firewall and one or more sensors to protect major backbone networks, such as those that support internal servers and database resources (location 3). The benefits of this placement include the following:

- Monitors a large amount of a network's traffic, thus increasing the possibility of spotting attacks

- Detects unauthorized activity by authorized users within the organization's security perimeter

Thus, a sensor at location 3 is able to monitor for both internal and external attacks. Because the sensor monitors traffic to only a subset of devices at the site, it can be tuned to specific protocols and attack types, thus reducing the processing burden.

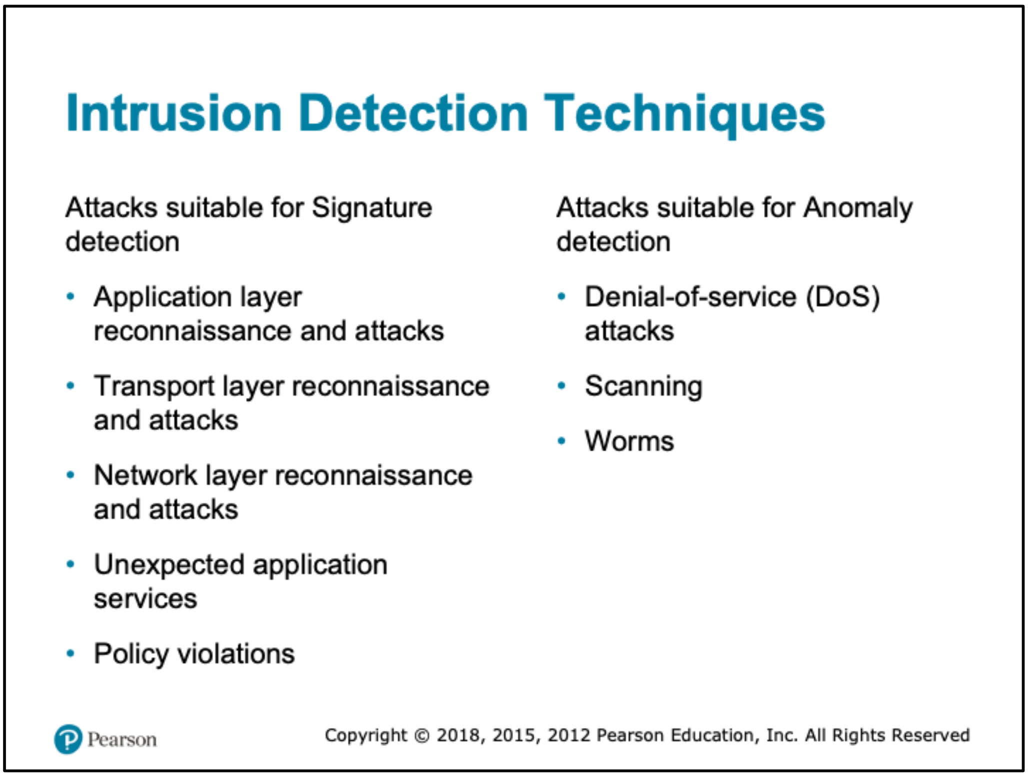
Finally, the network facilities at a site may include separate LANs that support user workstations and servers specific to a single department. The administrator could configure a firewall and NIDS sensor to provide additional protection for all of these networks or target the protection to critical subsystems, such as personnel and financial networks (location 4). A sensor used in this latter fashion provides the following benefits:

- Detects attacks targeting critical systems and resources

- Allows focusing of limited resources to the network assets considered of greatest value

As with a sensor at location 3, a sensor at location 4 can be tuned to specific protocols and attack types, thus reducing the processing burden.

A diagram illustrates the examples of N I D S senor deployment. All Internet traffic passes through an external firewall. Traffic from the service network such as web, mail, D N S, etcetera are monitored. The external firewall also provides a degree of protection for those parts of the network. A common location for a NIDS sensor is just inside the external firewall, location 1 in the diagram. A N I D S sensor is located between the external firewall and the Internet or WAN, location 2. In addition to a sensor at the boundary of the network, on either side of the external firewall, the administrator configures a firewall and one or more sensors that support internal servers and database resource network at location 3. Finally, the network facilities at a site may include separate L A N's that support user workstations and servers specific to a single department at location 4.



Intrusion Detection Techniques

Attacks suitable for Signature detection

- Application layer reconnaissance and attacks
- Transport layer reconnaissance and attacks
- Network layer reconnaissance and attacks
- Unexpected application services
- Policy violations

Attacks suitable for Anomaly detection

- Denial-of-service (DoS) attacks
- Scanning
- Worms

Pearson

As with host-based intrusion detection, network-based intrusion detection makes use of signature detection and anomaly detection. Unlike the case with HIDS, a number of commercial anomaly NIDS products are available [GARC09]. One of the best known is the Statistical Packet Anomaly Detection Engine (SPADE), available as a plug-in for the Snort system that we discuss later.

NIST SP 800-94 (*Guide to Intrusion Detection and Prevention Systems*, July 2012) lists the following as examples of that types of attacks that are suitable for signature detection:

• **Application layer reconnaissance and attacks**: Most NIDS technologies analyze several dozen application protocols. Commonly analyzed ones include Dynamic Host Configuration Protocol (DHCP), DNS, Finger, FTP, HTTP, Internet Message Access Protocol (IMAP), Internet Relay Chat (IRC), Network File System (NFS), Post Office Protocol (POP), rlogin/rsh, Remote Procedure Call (RPC), Session Initiation Protocol (SIP), Server Message Block (SMB), SMTP, SNMP, Telnet, and Trivial File Transfer Protocol (TFTP), as well as database protocols, instant messaging applications, and peer-to-peer file sharing software. The NIDS is looking for attack patterns that have been identified as targeting these protocols. Examples of attack include buffer overflows, password guessing, and malware transmission.

• **Transport layer reconnaissance and attacks**: NIDSs analyze TCP and UDP traffic and perhaps other transport layer protocols. Examples of attacks are unusual packet fragmentation, scans for vulnerable ports, and TCP-specific attacks such as SYN floods.

• **Network layer reconnaissance and attacks**: NIDSs typically analyze IPv4, IPv6, ICMP, and IGMP at this level. Examples of attacks are spoofed IP addresses and illegal IP header values.

• **Unexpected application services:** The NIDS attempts to determine if the activity on a transport connection is consistent with the expected application protocol. An example is a host running an unauthorized application service.

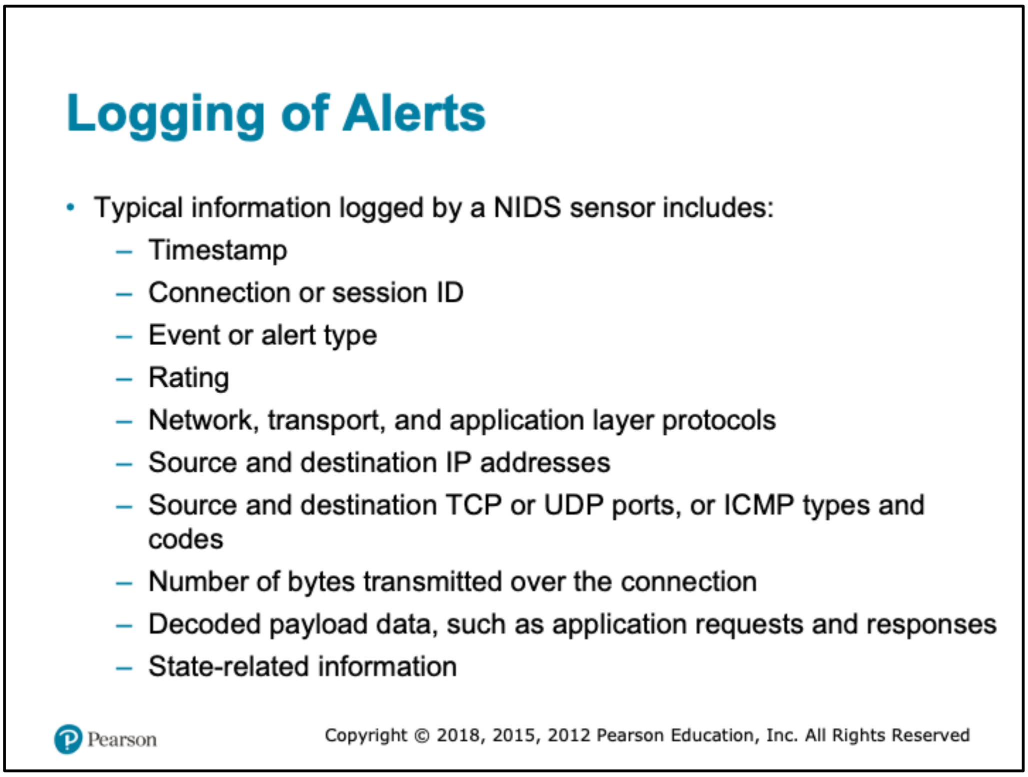
• **Policy violations:** Examples include use of inappropriate Web sites and use of forbidden application protocols.

NIST SP 800-94 lists the following as examples of the types of attacks that are suitable for anomaly detection:

• **Denial-of-service (DoS) attacks:** Such attacks involve either significantly increased packet traffic or significantly increase connection attempts, in an attempt to overwhelm the target system. These attacks are analyzed in Chapter 7. Anomaly detection is well suited to such attacks.

• **Scanning**: A scanning attack occurs when an attacker probes a target network or system by sending different kinds of packets. Using the responses received from the target, the attacker can learn many of the system's characteristics and vulnerabilities. Thus, a scanning attack acts as a target identification tool for an attacker. Scanning can be detected by atypical flow patterns at the application layer (e.g., banner grabbing3), transport layer (e.g., TCP and UDP port scanning), and network layer (e.g., ICMP scanning).

• **Worms**: Worms spreading among hosts can be detected in more than one way. Some worms propagate quickly and use large amounts of bandwidth. Worms can also be detected because they can cause hosts to communicate with each other that typically do not, and they can also cause hosts to use ports that they normally do not use. Many worms also perform scanning. Chapter 6 discusses worms in detail.



Logging of Alerts

- Typical information logged by a NIDS sensor includes:
  - Timestamp
  - Connection or session ID
  - Event or alert type
  - Rating
  - Network, transport, and application layer protocols
  - Source and destination IP addresses
  - Source and destination TCP or UDP ports, or ICMP types and codes
  - Number of bytes transmitted over the connection
  - Decoded payload data, such as application requests and responses
  - State-related information

Pearson

When a sensor detects a potential violation, it sends an alert and logs information related to the event. The NIDS analysis module can use this information to refine intrusion detection parameters and algorithms. The security administrator can use this information to design prevention techniques. Typical information logged by a NIDS sensor includes the following:

- Timestamp (usually date and time)

- Connection or session ID (typically a consecutive or unique number assigned to each TCP connection or to like groups of packets for connectionless protocols)

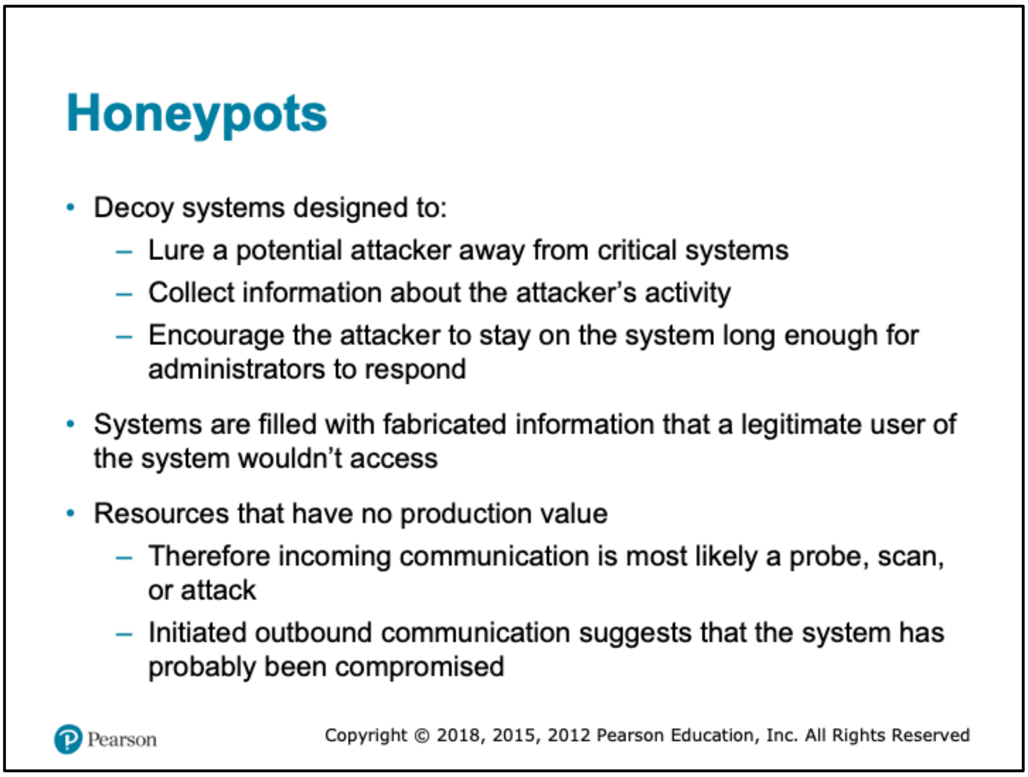
- Event or alert type

- Rating (e.g., priority, severity, impact, confidence)

- Network, transport, and application layer protocols

- Source and destination IP addresses

- Source and destination TCP or UDP ports, or ICMP types and codes
- Number of bytes transmitted over the connection
- Decoded payload data, such as application requests and responses
- State-related information (e.g., authenticated username)



Honeypots

- Decoy systems designed to:
  - Lure a potential attacker away from critical systems
  - Collect information about the attacker’s activity
  - Encourage the attacker to stay on the system long enough for administrators to respond
- Systems are filled with fabricated information that a legitimate user of the system wouldn’t access
- Resources that have no production value
  - Therefore incoming communication is most likely a probe, scan, or attack
  - Initiated outbound communication suggests that the system has probably been compromised

Pearson

A further component of intrusion detection technology is the honeypot. Honeypots are decoy systems that are designed to lure a potential attacker away from critical systems. Honeypots are designed to:

- Divert an attacker from accessing critical systems.

- Collect information about the attacker’s activity.

- Encourage the attacker to stay on the system long enough for administrators to respond.

These systems are filled with fabricated information designed to appear valuable but that a legitimate user of the system would not access. Thus, any access to the honeypot is suspect. The system is instrumented with sensitive monitors and event loggers that detect these accesses and collect information about the attacker’s activities. Because any attack against the honeypot is made to seem successful, administrators have time to mobilize and log and track the attacker without ever exposing productive systems.

The honeypot is a resource that has no production value. There is no legitimate reason for anyone outside the network to interact with a honeypot.

Thus, any attempt to communicate with the system is most likely a probe, scan, or attack. Conversely, if a honeypot initiates outbound communication, the system has probably been compromised.

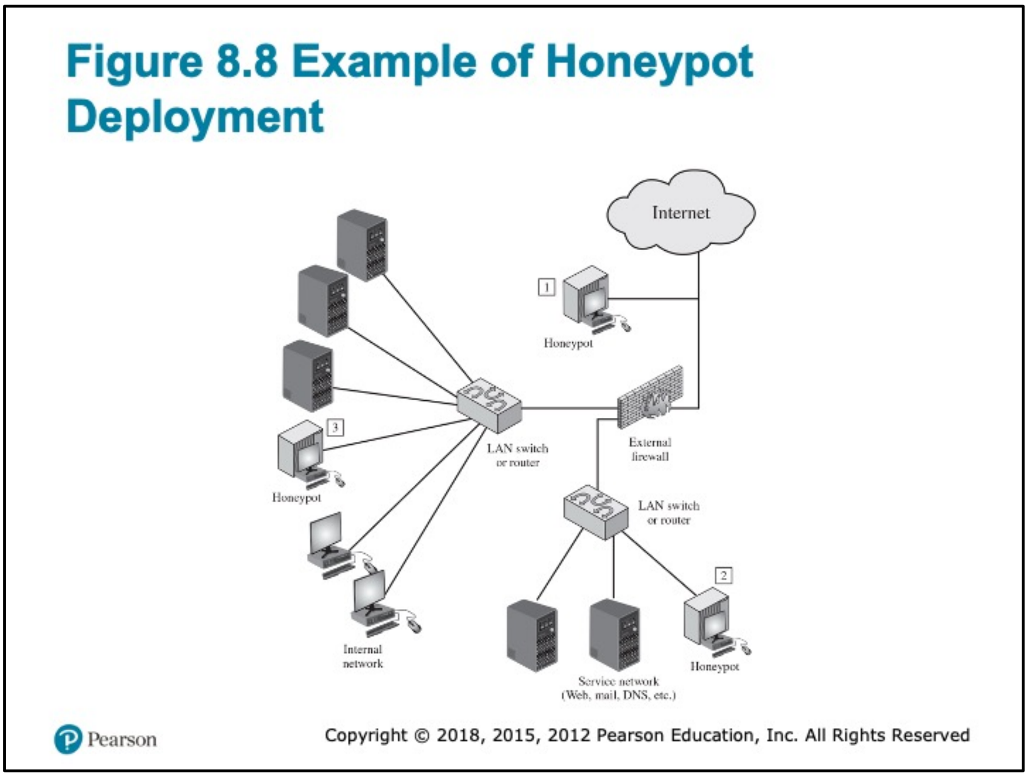


Figure 8.8 Example of Honeypot Deployment

Honeypots can be deployed in a variety of locations. Figure 8.8 illustrates some possibilities. The location depends on a number of factors, such as the type of information the organization is interested in gathering and the level of risk that organizations can tolerate to obtain the maximum amount of data.

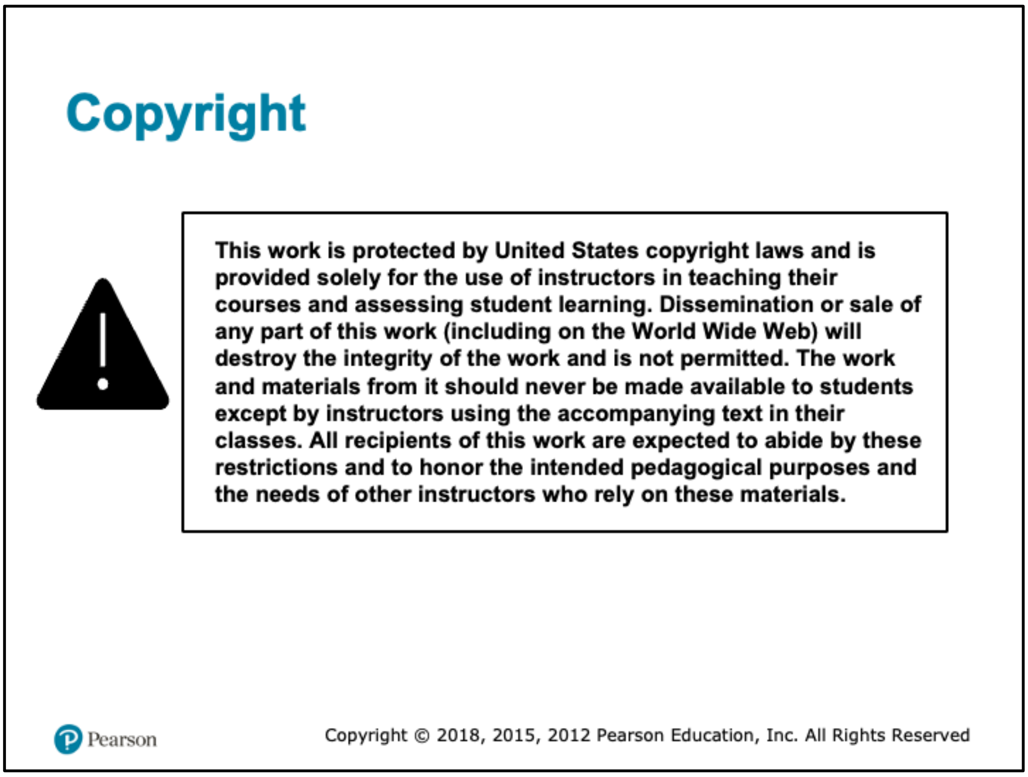
A honeypot outside the external firewall (location 1) is useful for tracking attempts to connect to unused IP addresses within the scope of the network. A honeypot at this location does not increase the risk for the internal network. The danger of having a compromised system behind the firewall is avoided. Further, because the honeypot attracts many potential attacks, it reduces the alerts issued by the firewall and by internal IDS sensors, easing the management burden. The disadvantage of an external honeypot is that it has little or no ability to trap internal attackers, especially if the external firewall filters traffic in both directions.

The network of externally available services, such as Web and mail, often called the DMZ (demilitarized zone), is another candidate for locating a honeypot (location 2). The security administrator must assure that the other systems in the DMZ are secure against any activity generated by the honeypot. A disadvantage of this location is that a typical DMZ is not fully accessible, and the firewall typically blocks traffic to the DMZ the attempts to access unneeded services. Thus, the firewall either has to open up the traffic beyond what is permissible, which is risky, or limit the effectiveness of the honeypot.

A fully internal honeypot (location 3) has several advantages. Its most important advantage is that it can catch internal attacks. A honeypot at this location can also detect a misconfigured firewall that forwards impermissible traffic from the Internet to the internal network. There are several disadvantages. The most serious of these is if the honeypot is compromised so that it can attack other internal systems. Any further traffic from the Internet to the attacker is not blocked by the firewall because it is regarded as traffic to the honeypot only. Another difficulty for this honeypot location is that, as with location 2, the firewall must adjust its filtering to allow traffic to the honeypot, thus complicating firewall configuration and potentially compromising the internal network.

An emerging related technology is the use of honeyfiles, that emulate legitimate documents with realistic, enticing names and possibly content. These documents should not be accessed by legitimate users of a system, but rather act as bait for intruders exploring a system. Any access of them is assumed to be suspicious [WHIT13]. Appropriate generation, placement, and monitoring of honeyfiles is an area of current research.

The internet data from the honeypot at location 1, outside the firewall passes through the external firewall and two L A N switches or routers and then to the honeypots at location 2 and location 3 and passes to the internal networks and service networks for example web, mail, D N S etcetera.



Copyright

Pearson