CHAPTER 2

Signatures and Actions

Before you decide to buy a product of any kind, you usually want to know exactly what it is you're buying. That way, you don't get something you don't want. This seems like a reasonable goal, but achieving this goal isn't always easy. What's challenging is the vocabulary used to describe what the product does and how it works. Two different products might have a feature with the same name, but the feature in each product might actually be completely unrelated. Without an industry agreed-upon set of definitions, product marketing can use terminology to make each product appealing to customers, even if this usage makes it difficult for customers to compare the functionality between different IPS products.

Take the purchase of a new vehicle, for example. Three different automobiles claim to have drive stabilization systems. That sounds great, but does the system work in the same way for each car? Is one more suitable for your needs than the other? How is the system implemented? Close examination might show how the system in one car reduces vibration when driving over bumpy roads whereas in another car it helps control the vehicle's balance during sharp turns. The name for the feature is exactly the same, but what it actually does is very different.

Seeing through the fog of feature names and marketing buzzwords is especially difficult when the product of interest is in a new technology, such as Intrusion Prevention. Intrusion Prevention System (IPS) product data sheets and websites tend to use vague product descriptors like deep packet inspection, anomaly detection, innate defense models, signatures, and behavior-based and advanced network intelligence. The descriptors might be accurate, but the functionality behind the words is often not consistent from product to product.

The way to see through the words and discern the product functionality is to create clear definitions for commonly used feature names. One feature commonly associated with IPS is signatures. Attack signatures have been around for long enough that the definition should be universally understood, but that's not the case. Simply put, an IPS signature is any distinctive characteristic that identifies something. Using this definition, all IPS products use signatures of some kind, regardless of what the product descriptions claim. To find something and stop it, you must be able to identify it, and for you to identify it, it must display a distinct characteristic. Signatures are distinguished by the following characteristics:

- Signature types
- Signature trigger
- Signature actions

Signature Types

Signatures fall into one of the following two basic categories depending on their functionality:

- Atomic signatures
- Stateful signatures

This section examines these signature types in further detail. Furthermore, the triggering mechanisms explained later in this chapter can be used with both of these base signature types. The major distinction between these two base signature types is whether or not the inspection process requires the IPS device to maintain state about previous actions that have been observed.

Atomic Signatures

Atomic signatures represent the simplest signature type. For an atomic signature, a single packet, activity, or event is examined to determine if the signature should trigger a signature action. Because these signatures trigger on a single event, they do not require your intrusion system to maintain state. The entire inspection can be accomplished in an atomic operation that does not require any knowledge of past or future activities.

> **STATE**
>
> State refers to situations in which you need to analyze multiple pieces of information that are not available at the same time. It also refers to tracking established TCP connections (connections that have gone through the initial three-way handshake). Valid TCP traffic also refers to traffic that has the correct sequence numbers for an established connection. For Network IPSs, state signatures usually refer to signatures that require analyzing traffic from multiple packets.

Atomic Signature Considerations

One drawback with atomic signatures is that you have to know all the atomic events that you want to look for. For each of these events, you then have to create the appropriate signature. As the number of atomic signatures increases, just managing the different signatures can become overwhelming.

Another drawback is that these signatures can be applied only to situations in which the context of the event is not important. For example, assume that you have a simple string match signature that triggers an alert action whenever the traffic that it is analyzing contains **/etc/passwd**. If you apply this simple string signature to monitor TCP traffic, an attacker can generate alerts by sending a flood of TCP packets with the **/etc/passwd** string in payload. The alerts are generated even if the connection is not part of a valid TCP connection (because it is an atomic signature). Furthermore, analyzing the alerts can minimize the time that your security staff spends identifying more serious attacks that represent valid attacks against your network. Generating a large number of bogus alerts can also impact the performance of your monitoring applications and devices.

Nevertheless, atomic signatures have their advantages. First, these signatures consume minimal resources (such as memory) on the IPS/IDS device. These signatures are also easy to understand because they search only for a specific event. Finally, traffic analysis for these atomic signatures can usually be performed very quickly and efficiently.

Host-Based Examples

Host-based IPS examines many operations on the system, including function calls, files accessed, and so on. One common method for detecting anomalous user behavior is to establish a baseline of the operations that a user normally performs on the system. Then by monitoring deviations from the baseline, you can detect potentially malicious activity. For example, if a function call is never invoked normally (except in connection with malicious activity), then triggering a signature action whenever it is called is a simple example of a host-based atomic signature. Another example of this is an application that you consider a problem. For example, you might want to trigger a signature action whenever a command shell is invoked on the local system.

NOTE Command shells are used to access the command-line interface on most operating systems. Accessing the command-line interface is a common mechanism to launch attacks against the system. On any operating system, such as Windows, which relies heavily on a graphical user interface, utilizing the command shell to configure the system can be indicative of uncommon behavior.

Network-Based Examples

A good example of a network-based atomic signature is the *LAND* attack. By inspecting a single packet, your Network-based (or Host-based) IPS can identify this attack. Because everything is contained in a single packet, no state information is needed to identify this attack.

LAND ATTACK

The LAND attack is a denial-of-service (DoS) attack in which an attacker sends a TCP packet (with the SYN bit set) to a system in which the source and destination IP address (along with the source and destination port) are the same. When it was first discovered, many IP stacks crashed the system when they received a LAND attack.

Stateful Signatures

Unlike atomic signatures, stateful signatures trigger on a sequence of specific events that requires the IPS device to maintain state. The length of time that the signatures must maintain state is known as the *event horizon*. Configuring the length of the event horizon is a tradeoff between consuming system resources and being able to detect an attack that occurs over a long period of time.

EVENT HORIZON

Stateful signatures usually require several pieces of data to match an attack signature. The maximum amount of time over which an attack signature can successfully be detected (from the initial data piece to the final data piece needed to complete the attack signature) is known as the *event horizon.* The intrusion system must maintain state information for the duration of the event horizon. The length of event horizon varies from one signature to another. The important point to consider is that an IPS cannot maintain state information indefinitely without eventually running out of resources. Therefore, an IPS uses a configured event horizon to determine how long it looks for a specific attack signature once an initial signature component is detected.

Stateful Signature Considerations

The main limitation to stateful signatures is that maintaining state consumes memory resources on your IPS/IDS device. Usually, however, this is not a significant problem if the IPS product is designed to efficiently use its resources. If your IPS does not efficiently manage resources when maintaining state, then the large consumption of resources (such as memory and CPU) can lead to a slow response time, dropped packets, missed signatures, and so on, which adversely impacts the effectiveness of your IPS.

Requiring a specific event to be detected in a known context increases the likelihood that the activity represents legitimate attack traffic. This minimizes the false positives generated by the stateful signatures.

Host-Based Examples

For a host-based example, we are going to use a commonly used Windows command shell called **cmd.exe**. As opposed to our atomic host-based example earlier in this chapter, in this situation, we do not want to trigger a signature action whenever **cmd.exe** is invoked (because our users use this program frequently). Our examination, however, reveals that many attacks invoke **cmd.exe** remotely. To remotely execute **cmd.exe**, the attacker must make a network connection to the host. This information can be used to refine our atomic signature by adding state. The stateful signature triggers a signature action when **cmd.exe** is invoked, but only if the application invoking **cmd.exe** first accepted a network connection.

The Host-based IPS must remember which applications have accepted network connections. This state information can then be examined whenever **cmd.exe** is invoked.

Network-Based Examples

Often, Network-based IPS signatures are stateful signatures because the information needed can usually be distributed across multiple packets. Even a simple string match signature is usually stateful because the string can occur across multiple packets (because the IPS must examine the data from all the packets until the successful match is made). For example, if you want to search for the string **/etc/password** in an HTTP URL, you might have to check multiple packets because the string can be distributed across more than one packet (although it can occur in a single packet as well).

Other examples of stateful signatures are the signatures used to monitor TCP traffic. To minimize the ability of an attacker to generate a large number of bogus alarms, most TCP attack signatures are valid only if the signature trigger is observed on a valid TCP connection. For example, suppose your signature triggers on the string **/etc/password** in a Telnet connection. Telnet uses TCP port 23, so the first thing that the IPS needs to track is established connections to TCP port 23. Then it also needs to track the sequence numbers for the established Telnet connections. Finally, whenever the string **/etc/password** is observed on an established TCP connection with the correct sequence numbers, then the signature triggers. Without maintaining this state, an attacker can generate a flood of invalid alarms by sending a flood of TCP packets to port 23 containing the string **/etc/password** (without ever actually establishing any valid TCP connections to port 23).

Signature Triggers

The heart of any IPS signature is the mechanism that causes it to trigger. These triggering mechanisms can be simple or complex, and every IPS incorporates signatures that use one or more of these basic *triggering mechanisms* to trigger signature actions. These triggering mechanisms can be applied to both atomic and stateful signatures. Current IPSs incorporate various triggering mechanisms when developing signatures, including the following:

- Pattern detection
- Anomaly-based detection
- Behavior-based detection

TRIGGERING MECHANISM

Triggering mechanisms refer to the conditions that cause an intrusion system to generate a signature action. For example, the triggering mechanism for a burglar alarm might be a motion detector that detects the movement of an individual entering the alarmed room. A Network IPS might trigger a signature action if it detects a packet with a payload containing a specific string going to a specific port. A Host-based IPS might trigger a signature action when a specific function call is invoked. Anything that can reliably signal an intrusion or security policy violation can be used as a triggering mechanism.

PROTOCOL DECODES

Another common triggering mechanism is called *protocol decodes*. Instead of simply looking for a pattern anywhere in a packet, protocol decodes involve breaking down a packet into the fields of a protocol and then searching for specific patterns in a specific protocol field or some other malformed aspect of the protocol fields. The advantage of protocol decodes is that it enables a more granular inspection of traffic and reduces false positives.

Table 2-1 shows the relationship between the various signature types and triggering mechanisms.

Table 2-1 *Signature Type Versus Signature Trigger*

Signature Trigger	**Signature Type**	
	Atomic Signature	**Stateful Signature**
Pattern detection	No state required to examine pattern to determine if signature action should be applied	Must maintain state or examine multiple items to determine if signature action should be applied
Anomaly detection	No state required to identify activity that deviates from normal profile	State required to identify activity that deviates from normal profile
Behavior detection	No state required to identify undesirable behavior	Previous activity (state) required to identify undesirable behavior

The following sections explain the signature triggering mechanisms in detail. Table 2-2 and Table 2-3 provide example signatures that illustrate the various combinations of signature types and triggering mechanisms to help clarify how the different signature types and triggers combine to create useful signatures.

Table 2-2 *Host-Based Signature Examples*

Signature Trigger	**Signature Type**	
	Atomic Signature	**Stateful Signature**
Pattern detection	Searching for the string **confidential** in a data file	Searching for the string **SELECT FROM** in a URI
Anomaly detection	Detecting a function call that is not part of the normal profile	Two function calls that are part of the normal profile, but have never been called within 1 second of each other
Behavior detection	Searching for any invocation of **cmd.exe**	Searching for an e-mail application (program that has previously generated or received e-mail traffic) invoking command.com

Table 2-3 *Network-Based Signature Examples*

Signature Trigger	**Signature Type**	
	Atomic Signature	**Stateful Signature**
Pattern detection	Detecting for an Address Resolution Protocol (ARP) request that has a source Ethernet address of FF:FF:FF:FF:FF:FF	Searching for the string *confidential* across multiple packets in a TCP session
Anomaly detection	Detecting traffic that is going to a destination port that is not in the normal profile	Verifying protocol compliance for HTTP traffic
Behavior detection	Detecting abnormally large fragmented packets by examining only the last fragment	Searching for RPC requests that do not initially utilize the PortMapper

Each of these triggering mechanisms has its benefits and drawbacks. Using the correct triggering mechanism in the appropriate situation greatly improves its efficiency. IPS devices that support multiple triggering mechanisms can more adequately support efficient signatures for a wide variety of activities without significantly impacting the performance of the IPS device.

By understanding the mechanisms that a signature can use to identify an activity, you can more efficiently determine a product’s true capabilities.

Pattern Detection

The simplest triggering mechanism is identifying a specific pattern. This pattern can represent a textual or binary string or it can be other patterns, such as a sequence of function calls. Besides simple string patterns, most systems provide enhanced pattern detection using the following mechanisms:

- Regular expression (regex) patterns
- Deobfuscation techniques

Specifying string patterns using *regex* provides the ability to efficiently search for textual patterns (using a single regular expression) while making it harder to bypass the pattern without detection.

REGULAR EXPRESSION

A *regex* is a pattern-matching language that enables you to define a flexible search pattern. Using *regex*, you can easily define complex search patterns. Many different programs use *regex* to enable you to define custom search strings. In the UNIX world, for example, the **grep** command is a common program that utilizes *regex* to search for text inside of files (or other output). For example, to perform a case-insensitive search for the word **attack** in the file named **output.results**, you can use the following command:

```
grep [Aa][Tt][Tt][Aa][Cc][Kk] output.results
```

This command finds various permutations of the word attack, such as ATTAck, attack, AttaCk, and so on.

Besides searching using flexible string patterns, many protocols (such as HTTP) accept multiple encoding mechanisms for input data. Some of the common encoding mechanisms include the following:

- American Standard Code for Information Interchange (ASCII)
- Hexadecimal
- Unicode

NOTE Hackers often combine these encoding methods in an effort to trick IPS devices and other monitoring applications. In the past, various web servers could be tricked into accessing data and applications from other directories in what is known as a directory traversal attack.

Unless your IPS can decode the input data correctly (before performing a pattern search), it will miss valid strings that have been obfuscated. For example, the following patterns each request the same web page:

- **http://10.10.10.10/index.html/../etc/password**
- **http://10.10.10.10/index.html/%2e%2e/etc/password**
- **http://10.10.10.10/index.html/%c0%ae%c0%ae/etc/password**

In each of these requests, the **..** is being represented differently. If your IPS correctly deobfuscates these requests (before performing a pattern search), then it can correctly detect the attempt to request a web page outside of the root of the web server's directory tree in each of these web requests (which all request the same web page).

Pattern Matching Considerations

Depending on the fidelity of the patterns that your signatures use, one of the problems with pattern matching can be the rate of false positives generated, especially with atomic signatures.

One of the benefits of pattern matching is that the signature clearly correlates to a specific attack. When the signature triggers, you know which attack it is detecting. This is different from anomaly detection in which the signature indicates only that something outside of the configured normal parameters has been detected.

Host-Based Examples

A Host-based IPS product might have the capability to examine the data inside a file on the hard disk. An example of a pattern-based signature trigger is the word **confidential** being detected in any data file. A stateful signature with the same triggering mechanism might look for the word **confidential** in inbound network traffic. The pattern is the same, but can be matched only if multiple packets have been re-assembled.

Network-Based Examples

Most Network-based IPS devices include a robust pattern-matching capability because many of the attack signatures involve searching for patterns in different network protocols. Most of these pattern-matching signatures are also stateful signatures because the information being examined can occur across multiple packets.

NOTE Examining network traffic using pattern matching does not work when the traffic stream is encrypted. However, Host-based IPS might be able to examine the traffic depending on where it is examining the traffic (before or after the decryption).

The following regex string searches for an attempt to change the working directory to the root directory during an FTP session:

```
[ \t]*[Cc][Ww][Dd][ \t]+[~]root
```

Examining traffic with a destination port of 21 (the default FTP server port) detects FTP sessions in which the user attempts to change the working directory to **root**.

NOTE If you have FTP servers listening on ports other than TCP port 21, then you need to configure your IPS to monitor these non-standard ports to have the IPS identify FTP attacks to these other ports.

Anomaly-Based Detection

Anomaly-based (also known as profile-based detection) signatures are not based on a specific event. Instead these signatures trigger when a certain activities deviate from what is considered normal. To utilize an anomaly-based signature, you must first determine what normal activity means for your network or host. This is usually accomplished by monitoring your network (or specific applications on your host) for a specific period of time to observe what is considered normal activity. Once you define normal activity, then you can configure your anomaly-based signature to trigger whenever activity on the network or a specific host deviates from your defined normal profile by a certain amount.

Anomaly-Based Detection Considerations

One of the biggest limitations of anomaly-based signatures is that you need to learn (or define) what is considered normal. As your network evolves, your definition of normal might also have to change. Furthermore, you need to guarantee that during your learning phase, your network is free of the attack traffic that you are going to detect (otherwise, this activity will be considered normal traffic). Furthermore, just defining normal can sometimes be difficult because most networks comprise a heterogeneous mixture of systems, devices, and applications that continually change.

Another potential drawback to anomaly-based systems is that when a signature generates an alert, it might be very difficult to correlate that alert back to a specific attack, because the alert indicates only that non-normal traffic has been detected. More analysis is required to determine whether the traffic represents an actual attack and what the attack actually accomplished. Furthermore, if the attack traffic happens to be similar to normal traffic, the attack might go undetected.

Because anomaly-based signatures do not look for specific attacks, they can be used to detect previously unpublished attacks. This is a major advantage for anomaly-based detection. Instead of having to define a large number of signatures for various attack scenarios, you simply define a profile for normal activity. Any activity that deviates from this profile is then abnormal and triggers a signature action. The drawback is that an alert from an anomaly signature does not necessarily indicate an attack. It indicates only a deviation from the defined normal, which can sometimes occur from valid user traffic.

Host-Based Examples

A good example of anomaly detection is using the Profiler utility from the Cisco Security Agent (CSA) software. This tool allows you to monitor an application over a period of time. During this time, it records all of the functions calls that the application uses. You can then use this information to build a profile for the application that can then be used to identify when the application makes anomalous function calls (any function call that is not in the profile).

NOTE The Profiler utility is useful to analyze custom applications that are unique to your network. Most common applications (for a specific operating system) have already been incorporated into CSA's predefined rules. As of CSA version 4.5, the Profiler utility is now referred to as Application Behavior Investigation.

Sophisticated Host-based IPS tools might be able to apply stateful conditions to function calls recorded during the learning process. If two function calls that are a part of the normal profile occur within 1 second of each other and this has never happened before, this could trigger a signature. This is an example of a stateful signature with an anomaly-based triggering mechanism.

Network-Based Examples

A Network-based IPS can have various anomaly-based signatures. Some simple examples of anomaly signatures with the Cisco IPS solution are its flood signatures. These signatures detect floods of various types of traffic on your network. For example, you might want to monitor the amount of Internet Control Message Protocol (ICMP) traffic on your network. First, you need to measure the amount of ICMP traffic that appears on your network during normal operation. Then the flood signature triggers a signature action whenever the ICMP traffic on the network exceeds the configured maximum threshold for a specific length of time.

Other anomaly-based signatures involve ensuring that traffic to a set of ports matches a defined protocol specification. In this situation, the protocol specification defines what is considered normal. Any traffic that does not conform to this protocol specification is abnormal and triggers a signature action.

NOTE The Cisco application inspection and control (AIC) signatures provide signatures that provide you with anomaly detection signatures for both HTTP and the FTP by verifying protocol compliance.

Behavior-Based Detection

Behavior-based detection is similar to pattern detection, but instead of trying to define specific patterns, you are defining behaviors that are suspicious based on historical analysis. The behaviors define classes of activity that are known to be suspicious. For example, an e-mail client running shell commands using the Windows **cmd.exe** program normally indicates something unusual such as a virus attempting to do something to your system.

Behavior-Based Detection Considerations

Similar to pattern matching, behavior-based signatures must be defined before you can use them. It takes a lot of research to determine behaviors that do not occur normally and can accurately indicate suspicious behavior.

The use of behaviors enables a single signature to cover an entire class of activities without having to specify each individual situation. For example, having a signature that triggers a signature action when an e-mail client invokes **cmd.exe** enables you to apply the signature to any application whose behavior mimics the basic characteristics of an e-mail client without having to apply the signature to each e-mail client application individually. Therefore, if a user installs a new e-mail application, the signature still applies.

Host-Based Examples

Behavior-based signatures are easy to visualize at the host level. A simple example involves the **cmd.exe** program. This program is frequently used by malicious programs to execute commands and scripts on the system. It is also frequently used by users during their normal day-to-day operations. Using a behavior-based signature, you can detect suspicious invocations of **cmd.exe** while ignoring the normal uses of this program. For example, an e-mail client that invokes **cmd.exe** is performing suspicious activity (because the e-mail client does not need to run **cmd.exe** to operate). Incorporating a signature that triggers a signature action whenever **cmd.exe** is invoked by any e-mail client application is an excellent example of a behavior-based signature.

Network-Based Examples

At the network level, you might find it a little more difficult to identify the behavior-based signatures. For example, a signature that identifies attempts to directly access RPC applications is behavior-based, because the normal behavior is to first communicate with the PortMapper.

RPC PROTOCOL

The Sun Remote Procedure Call (RPC) protocol enables one system to run applications on another system across the network. These RPC applications are not bound to well-known ports. Instead, a program called the PortMapper keeps track of which RPC application is operating on which port. When a system wants to communicate with an RPC application on a remote system, it first contacts the PortMapper to find the port that the RPC application is operating. Then the system can communicate directly with the application through that port.

Signature Actions

Whenever a signature observes the activity that it is configured to detect, the signature triggers one or more actions. These actions fall into various categories, such as the following:

- Generating an alert
- Dropping or preventing the activity
- Logging the activity
- Resetting a TCP connection
- Blocking future activity
- Allowing the activity

Alert Signature Action

Monitoring the alerts generated by your Network-based and Host-based IPS systems is vital to understanding the attacks being launched against your network. If an attacker causes a flood of bogus alerts, examining them can overload your security analysts. Therefore, both network and host IPS solutions incorporate the following types of alerts to enable you to efficiently monitor the operation of your network:

- Atomic alerts
- Summary alerts

Understanding each of these types of alerts is vital to providing the most effective protection for your network.

Atomic Alerts

Atomic alerts (like atomic signatures) are generated every time a signature triggers. In some situations, this behavior is useful and indicates all occurrences of a specific attack. Other times, an attacker might be able to flood your monitor console with alerts if he can generate thousands of bogus alerts against your IPS devices or applications.

> **NOTE** As a hybrid between atomic alerts and summary alerts, some IPS solutions also enable you to generate a single atomic alert and then disable alerts (for that signature and source address) for a specific period of time. This prevents you from getting overwhelmed with alerts while still giving you an indication that a specific system is doing something suspicious.

Summary Alerts

Instead of generating alerts for each instance of a signature, some IPS solutions enable you to generate summary alerts. A summary alert is a single alert that indicates multiple occurrences of the same signature from the same source address and or port.

Alarm summary modes limit the number of alerts generated and make it difficult for an attacker to consume resources on your sensor. With the summarization modes, however, you also receive information on the number of times that the activity that matches a signature's characteristics was observed during a specific period of time.

When using alarm summarization, the first instance of intrusive activity usually triggers a normal alert. Then, other instances of the same activity (duplicate alarms) are counted until the end of the signature's summary interval. When the length of time specified by the summary interval has elapsed, a summary alarm is sent, indicating the number of alarms that occurred during the time interval specified by the summary interval parameter.

> **AUTOMATIC SUMMARIZATION**
>
> Besides configuring a signature to generate summary alerts, some IPS solutions also enable you to cause summarization to occur automatically (even though the default behavior is to generate atomic alerts). In this situation, if the number of atomic alerts exceeds a configured threshold in a specified amount of time, the signature automatically switches to generating summary alerts (instead of atomic alerts). Then, after a defined period of time, the signature reverts to its original configuration. Automatic summarization enables you to automatically regulate the amount of alerts being generated.

Drop Signature Action

One of the most powerful actions for an IPS device is the capability to drop packets or prevent an activity from occurring. This action enables the device to stop an attack before it has the chance to perform malicious activity. Unlike a traditional IDS device, the IPS device actively forwards packets across two of its interfaces. Therefore, the analysis engine has the option to decide which packets should be forwarded and which packets should be dropped.

Besides dropping individual packets, the drop action can be expanded to drop all packets for a specific session or even all packets from a specific host for a certain amount of time. By dropping traffic for a connection or host, the IPS conserves resources by efficiently dropping traffic without having to analyze each packet separately.

Log Signature Action

In some situations, you do not necessarily have enough information to stop an activity, but you want to log the actions or packets that are seen so that you can analyze this information in more detail. By performing a detailed analysis, you can identify exactly what is taking place and make a decision as to whether it should be allowed or denied in the future.

Suppose you have a signature that looks for the string **/etc/password** and you configure the string with the logging action (based on the attacker IP address). Whenever the signature triggers, the IPS devices begins logging the traffic from the attacker's IP address for a specified period of time (or specified number of bytes). This log information is usually stored on the IPS device in a specific file. Because the signature also generates an alert, you observe the alert on your management console. Then you can retrieve the log data from the IPS device and analyze the activity that the attacker performed on the network after triggering the initial alarm.

Block Signature Action

Most IPS devices have the capability to block future traffic by having the IPS device update the access control lists (ACLs) on one of your infrastructure devices. This ACL stops traffic from an attacking system without requiring the IDS to consume resources analyzing the traffic. After a configured period of time, the IDS device removes the ACL. Network IPS devices usually provide this blocking functionality along with the other actions such as dropping unwanted packets. One advantage of utilizing the blocking action is that a single IPS device can stop traffic at multiple locations throughout your network, regardless of the location of the IPS device itself. For example, an IPS device located deep within the network can apply ACLs at your perimeter router or firewall.

TCP Reset Signature Action

A basic action that can be used to terminate TCP connections is generating a packet for the connection with the TCP RST flag set. Many IPS devices use the TCP reset action to abruptly end a TCP connection that is performing unwanted operations.

Allow Signature Action

The final action might seem a little confusing, because most IPS devices are designed to stop or prevent unwanted traffic on your network. The allow action is necessary so that you can define exceptions to your configured signatures. When you configure your IPS device to disallow certain activities, you sometimes need to allow a few systems or users to be exceptions to the configured rule. Configuring exceptions enables you to take a more restrictive approach to security because you first deny everything and then allow only the activities that are needed.

For example, suppose that the IT department routinely scans your network using a common vulnerability scanner. This scanning causes your IPS to trigger various alerts. These are the same alerts that the IPS generates if an attacker scans your network. By allowing the alerts from the approved IT scanning host, you can protect your network from intrusive scans while eliminating the false positives generated by the routine IT approved scanning.

NOTE Some IPS devices provide the allow action indirectly through other mechanisms, such as signature filters. If an IPS does not provide the allow action directly (through an action such as permit or allow), you need to search the product's documentation to find the mechanism you can use to enable exceptions to signatures.

Summary

Different products use different terminology to describe their product's functionality. For explanation purposes, our definition of a signature is any distinctive characteristic that identifies something. Based on this definition, all IPS devices use signatures to identify activity in your network traffic and on hosts on your network. Signatures are distinguished by the following characteristics:

- Signature type
- Signature trigger
- Signature actions

Signature types fall into the following two base categories:

- Atomic
- Stateful

The major distinction between these two base signature types is that atomic signatures do not require the IPS device to maintain state information about previous activity.

In conjunction with the base signature types, a signature needs to trigger one or more actions depending on one of the following triggering mechanisms:

- Pattern detection
- Anomaly-based detection
- Behavior-based detection

Table 2-4 outlines the relationship between the base signature types and the triggering mechanisms.

Table 2-4 *Signature Type Versus Signature Trigger*

Signature Trigger	**Signature Type**	
	Atomic Signature	**Stateful Signature**
Pattern detection	No state required to examine pattern to determine if signature action should be applied	Must maintain state or examine multiple items to determine if signature action should be applied
Anomaly detection	No state required to identify activity that deviates from normal profile	State required to identify activity that deviates from normal profile
Behavior detection	No state required to identify undesirable behavior	Previous activity (state) required to identify undesirable behavior

Pattern detection is the simplest triggering because it involves searching for a specific predefined pattern. This pattern might be textual, binary, or even a series of function calls.

Anomaly-based detection involves first defining a profile of what is considered normal. This normal profile can be learned by monitoring activity over a period of time. It can also be based on a defined specification (such as an RFC). Whenever activity is observed that is not included in the normal profile, the signature triggers some action. Correlating the signature to a specific attack, however, can be complicated.

Behavior-based detection is similar to pattern detection, but it detects classes of activities based on known unacceptable behavior. Therefore, instead of many signatures for each unwanted activity, a single signature can watch for a specific behavior. Once the behavior has been detected, the appropriate signature actions are applied.

Detecting unwanted activity is only the initial step in protecting your network. Once a signature triggers, your IDS device must take certain configured actions to mitigate the activity identified. Signature actions fall into the following categories:

- Generating an alert
- Dropping or preventing the activity
- Logging the activity
- Resetting a TCP connection
- Blocking future activity
- Allowing the activity

The alerts (or alarms) generated by your IPS device enable you to monitor the attacks being launched against your network. To efficiently monitor alerts, IPS devices incorporate the following types of alerts:

- Atomic alerts
- Summary alerts