

Network Intrusion Detection Signatures, Part One

Karen Kent Frederick 2001-12-19

Network Intrusion Detection Signatures, Part One

by *Karen Kent Frederick*

last updated December 19, 2001

This is the first in a series of articles on understanding and developing signatures for network intrusion detection systems. In this article we will discuss the basics of network IDS signatures and then take a closer look at signatures that focus on IP, TCP, UDP and ICMP header values. Such signatures ignore packet payloads and instead look for certain header field values or combinations of values. By learning about network IDS signatures, you'll have more knowledge of how intrusion detection systems operate, and you'll have a better foundation to write your own IDS signatures.

Signature Basics

A network IDS signature is a pattern that we want to look for in traffic. In order to give you an idea of the variety of signatures, let's quickly review some examples and some of the methods that can be used to identify each one:

- Connection attempt from a reserved IP address. This is easily identified by checking the source address field in an IP header.
- Packet with an illegal TCP flag combination. This can be found by comparing the flags set in a TCP header against known good or bad flag combinations.
- Email containing a particular virus. The IDS can compare the subject of each email to the subject associated with the virus-laden email, or it can look for an attachment with a particular name.
- DNS buffer overflow attempt contained in the payload of a query. By parsing the DNS fields and checking the length of each of them, the IDS can identify an attempt to perform a buffer overflow using a DNS field. A different method would be to look for exploit shellcode sequences in the payload.
- Denial of service attack on a POP3 server caused by issuing the same command thousands of times. One signature for this attack would be to keep track of how many times the command is issued and to alert when that number exceeds a certain threshold.
- File access attack on an FTP server by issuing file and directory commands to it without

first logging in. A state-tracking signature could be developed which would monitor FTP traffic for a successful login and would alert if certain commands were issued before the user had authenticated properly.

As you can see from this list, signatures range from very simple – checking the value of a header field – to highly complex signatures that may actually track the state of a connection or perform extensive protocol analysis. In this article, we'll be looking at some of the simplest signatures and discussing the complexities involved in developing even the most basic signature. Note that signature capabilities vary greatly among IDS products, so some of the techniques described here may not be possible to implement in the IDS that you use. For example, some network IDS products provide little ability to customize existing signatures or write your own, while other IDS products give you the ability to customize all their signatures and write almost any signature you can think of. Another important factor to consider is that some IDS products can only check certain header or payload values, while other products can give you the data from any portion of any packet.

What Functions Do Signatures Serve?

This may seem like an obvious question, but what is the purpose of an intrusion detection signature? The answer is that different signatures have different goals. The obvious answer is that we want to be alerted when an intrusion attempt occurs. But let's take a moment to think about other reasons why we might want to write or modify a signature. Perhaps you're seeing some odd traffic on your network and you want to be alerted the next time it occurs. You've noticed that it has unusual header characteristics, and you want to write a signature that will match this known pattern. Or perhaps you are interested in configuring your IDS to identify abnormal or suspicious traffic in general, not just attacks or probes. Some signatures may tell you which specific attack is occurring or what vulnerability the attacker is trying to exploit, while other signatures may just indicate that unusual behavior is occurring, without specifying a particular attack. It will often take significantly more time and resources to identify the tool that's causing malicious activity, but it will give you more information as to why you're being attacked and what the intent of the attack is.

Header Values

Now that we've taken a quick look at signature types, let's focus on a simple signature characteristic: header values. Some header values are clearly abnormal, so they make great

candidates for signatures. A classic example of this is a TCP packet with the SYN and FIN flags set. This is a violation of [RFC 793](#) (which defines the TCP standard), and has been used in many tools in an attempt to circumvent firewalls, routers and intrusion detection systems. Many exploits include header values that purposely violate RFCs, because many operating systems and applications have been written on the assumption that the RFCs would not be violated and don't perform proper error handling of such traffic. Also, many tools either contain coding mistakes or are incomplete, so that crafted packets produced by them contain header values that violate RFCs. Both poorly written tools and various intrusion techniques provide distinguishing characteristics that can be used for signature purposes.

This all sounds great – but of course, there's a catch. Not all OSs and applications completely adhere to the RFCs. In fact, many have at least one facet of their behavior that violates an RFC. Also, over time, protocols may implement new features that are not included in an RFC. And new standards emerge over time, which may "legalize" values that were previously illegal; [RFC 3168](#), for Explicit Congestion Notification (ECN), is a good example of this. So an IDS signature based strictly on an RFC may produce many false positives. Still, the RFCs make a great basis for signature development, because so much malicious activity violates RFCs. Because of RFC updates and other factors that we'll discuss later, it's important to review and update existing signatures periodically.

Although illegal header values are certainly a fundamental component of signatures, legal but suspicious header values are at least as important. For example, alerting on connections to suspicious port numbers such as 31337 or 27374 (often associated with Trojans) may provide a quick way of identifying Trojan activity. Unfortunately, some normal, benign traffic may happen to use the same port numbers. Without using a more detailed signature that includes other characteristics of the traffic, you won't be able to determine the true nature of this traffic. Suspicious but legal values such as a port number are best used in combination with other values.

Identifying Possible Signature Components

The best way to understand the issues in developing a signature based on header values is to walk through an example. `synscan` is a tool that is widely used in scanning and probing systems. `synscan` activity was seen in unusually large amounts on the Internet in early 2001 because its code was used to create the first phase of the Ramen worm. This activity makes a great example, because the packets have several distinguishing characteristics. Here are some

of the IP and TCP header values that were present in Ramen worm packets during the first stage of the worm's spread. (Note that my box was configured to silently drop unsolicited traffic, so I only saw the first packet of each attempt.)

- Various source IP addresses
- TCP source port 21, destination port 21
- Type of service 0
- IP identification number 39426
- SYN and FIN flags set
- Various sequence numbers set
- Various acknowledgment numbers set
- TCP window size 1028

Now that we know what the characteristics of the synscan packet headers are, we can start to consider what a good signature would be. We're looking for values that are illegal, unusual or suspicious – in many cases, these characteristics correspond to the vulnerabilities that the attacker is trying to exploit, or a particular technique that the attacker uses. Packet values that are completely normal don't make good signature characteristics by themselves, although they are often included to limit the amount of traffic that we study. For example, we would include the normal IP protocol value of 6 for a protocol, so that we only check TCP packets. But other characteristics that are completely normal, such as the type of service set to 0, are much less likely to be helpful in signature development.

Certain characteristics of the synscan packets are anomalous and could be used in signatures:

- Having only the SYN and FIN flags set is a well-known sign of malicious activity.
- Another sign that these are crafted packets is that the acknowledgment number has various values but the ACK flag is not set. When that flag isn't set, the acknowledgment number should be set to 0.
- Another suspicious characteristic is that the source and destination ports are both set to 21, usually associated with FTP servers. When these ports equal each other, we say that they are reflexive. With a few exceptions (such as certain types of NetBIOS traffic), we should normally not see these two values equal to each other. Reflexive ports don't violate TCP standards, but in most cases they are unexpected and unusual. In normal FTP traffic, we would expect to see a high port number (greater than 1023) as the source and port 21 as the destination.

So far we've identified three likely signature elements: the SYN and FIN flags set, the acknowledgment number set to a non-zero value without the ACK flag set, and reflexive ports set to 21. There are two more items that are noteworthy: the TCP window size, which is always set to 1028, and the IP identification number, which is set in all the packets to 39426. Normally, we would expect the TCP window size to be larger than 1028; although this value is not terribly abnormal, it's more than a coincidence that it's the same in all of these. Likewise, the IP identification number is set in all packets to 39426. The IP RFC specifies that this value should vary among packets, so the constant value is very suspicious.

Choosing a Signature

Because we've identified five potential signature elements, we have many different options for developing a header-based signature, because a signature could include any one or more of these characteristics. A simple signature would be packets with only the SYN and FIN flags set. Although this would certainly be a good indicator of likely malicious activity, it doesn't give us any idea why this activity occurred. Remember that SYN and FIN have traditionally been used together to circumvent firewalls and other devices, so their presence could indicate that scans are being conducted, information gathered or attacks launched. So a signature based on SYN and FIN only may be too simple to meet our goals.

However, a signature based on all five suspicious characteristics may be too specific. Although it would provide much more precise information about the source of the activity, it would also be far less efficient than a signature that only checks one header value. Signature development is always a tradeoff between efficiency and accuracy. In many cases, simpler signatures are more prone to false positives than more complex signatures, because simpler signatures are much more general. But more complex signatures may be more prone to false negatives than simpler signatures, because one of the characteristics of a tool or methodology may change over time.

Let's assume that one of our signature's goals is to determine the tool being used. So besides the SYN and FIN flags, what other attributes would be best to examine? Well, the reflexive port numbers are suspicious, but they don't provide a specific enough signature, as many tools use them, as well as some legitimate traffic. The ACK value set without an ACK flag is clearly invalid and might make a good signature on its own, as well as paired with SYN and FIN. The window size of 1028, although a bit suspicious, could occur naturally. So could the IP identification

number of 39426. We could develop several signatures that use various combinations of these characteristics. It's often unclear as to what the best signature is, especially because the optimal signature may vary among environments and is also likely to change over time.

Conclusion

In the next article in this series, we'll decide which attributes we should use for our synscan signature and then evaluate the effectiveness of that signature against more synscan traffic. We'll further investigate the merits of general signatures as opposed to specific signatures. We'll also continue to look at the role that IP protocol header values play in signature development.

Karen Kent Frederick is a senior security engineer for the Rapid Response Team at NFR Security. She is a graduate of the University of Wisconsin-Parkside and is currently completing her master's in computer science, focusing in network security, through the University of Idaho's Engineering Outreach program. Karen has over 10 years of experience in technical support, system administration and information security. She holds several certifications, including SANS GIAC Certified Intrusion Analyst, GIAC Certified Unix Security Administrator, and GIAC Certified Incident Handler. She is one of the authors of "Intrusion Signatures and Analysis", and she is a contributing author to the "Handbook of Computer Crime Investigation".

Relevant Links

[Network Intrusion Detection Signatures, Part 2](#)

Karen Kent Frederick

[Network Intrusion Detection Signatures, Part 3](#)

Karen Kent Frederick

[Privacy Statement](#)

Copyright 2006, SecurityFocus