

Network Intrusion Detection Signatures, Part Two

Karen Kent Frederick 2002-01-22

Network Intrusion Detection Signatures, Part Two

by *Karen Kent Frederick*

last updated January 22, 2002

This is the second in a series of articles on understanding and developing signatures for network intrusion detection systems. In the [first installment](#) we looked at signature basics, the functions that signatures serve, header values, signature components, and choosing signatures. In this article we will continue our discussion of IP protocol header values in signatures by closely examining some signature examples. Although it may be relatively easy to develop a signature that matches a particular type of traffic, it will likely cause unexpected false positives and false negatives. Signatures must be carefully developed and tested in order to create a signature set that is highly accurate, yet is also as efficient as possible.

Evaluating the Effectiveness of a Signature

In the previous article, we looked at the characteristics of the packets sent by the synscan tool (as implemented in the Ramen worm) and identified traits that were unusual or suspicious, or that violated standards. We then tried to determine which of these traits might make a good signature. Based on those characteristics, let's create a signature that will look for all three of the following attributes in each TCP packet:

- Only the SYN and FIN flags set
- IP identification number 39426
- TCP window size of 1028

The first characteristic, having only the SYN and FIN flags set, is too general to be used alone to identify synscan activity. Even though the second and third items can and do occur in legitimate traffic, the odds of both occurring in the same packet are very low, so it's reasonable to use them along with the SYN and FIN flags to create a detailed signature. Adding the other synscan attributes would not significantly improve the accuracy of this signature, although they would increase the amount of resources needed to identify it. So do these three characteristics make a strong signature for the synscan tool? Well...yes and no. They constitute a great signature for that particular version of the synscan tool; however, there have been many

versions of the synscan tool, and it's certainly likely that some of them have different packet header characteristics.

A much more interesting question is, will the signature that we've developed be able to detect variants on the synscan tool – that is, other tools that have been developed using the synscan tool? In the case of the Ramen worm, the answer is yes; but what about other such traffic? Will our signature be able to find synscan variants? It's often shortsighted to develop just a signature that is specific to a particular implementation of an attack; by using a combination of general and specific signatures, you can often create a much better overall solution. An intrusion detection signature set is much more valuable if it can detect not only known attacks, but also future and unknown attacks. Let's examine this concept in more detail.

Applying the Signature to More Anomalous Traffic

Within weeks of the Ramen worm/synscan activity that our signature is based on, I received some additional scanning attempts that had a similar signature, with a few important differences:

- Instead of only the SYN and FIN flags being set, only the SYN flag was set. This is a normal TCP packet characteristic.
- The TCP window size was always 40 instead of 1028. 40 is an unusually small window size for an initial SYN packet and is certainly much less likely to be seen in a normal packet than 1028.
- The reflexive port number was 53 instead of 21. Old versions of BIND used reflexive ports for certain operations, but newer versions of BIND don't, so we wouldn't expect to see reflexive port TCP 53 very often.

Because there were so many similarities between the first set of packets and the second, we can reasonably conclude that the second set of packets were either generated by a different version of synscan or by a different tool that was based on the synscan code. Obviously, the synscan signature we developed won't catch this variant, because two of the three characteristics we were checking for have changed. So now what do we do? Well, we could create an additional synscan signature that would match the variant. Or we could adjust our goals, and focus on alerting on generally abnormal behavior rather than on particular implementations of tools. Or we could do both, creating some specific signatures that alert on the exploit or scanner implementations and some general signatures that look for basic

anomalies.

General signatures that might be useful for this set of traffic are:

- TCP packet with an acknowledgement value set to a non-zero number, but the acknowledgment flag is not set
- TCP packet with only the SYN and FIN flags set
- TCP packet with the initial TCP window size below a certain value (which would include 40)

Two of these three general signatures would match for both types of packets that we've looked at so far.

Identifying Traffic from Yet Another Variant

Weeks after seeing the apparent synscan variant that we just reviewed, I received some packets with almost the same characteristics. In fact, the only difference was that the IP identification number was no longer static. One likely explanation for this is that this third group of packets was made by a variant of the variant of synscan that made the second set of packets. It's possible that someone took parts of the synscan variant's code and created another tool from it. It's also possible that someone fixed the synscan code so it would no longer have a static IP identification number. This makes synscan traffic less distinctive and more stealthy from an intrusion detection system point of view by eliminating the IP identification number as a potential signature characteristic.

By this point, it should be obvious that the original synscan signature that we developed would not match this traffic. In fact, none of the three components of the original signature are present in this variant. But if you look at the three general signatures we developed, you'll see that two of the three would match this traffic too. Even though the characteristics of the traffic are changing, we can still identify it as anomalous through the use of more general signatures. This reinforces the importance of using general signatures that look for individual anomalies in traffic, rather than relying solely on highly detailed signatures that match particular attacks or exploits.

If you still wanted to specifically identify traffic from these variants, you'd want to create additional detailed signatures that would match them. Again, whether you should do this or not

depends on what your goal is. Many people couldn't care less which tool is being used to attack them; they're only interested in knowing that something bad is happening. But in many environments, there's a strong need to know the likely intent of the attacker. If you simply alert on generally anomalous activity, without understanding the potential significance of that activity, you won't know what the attacker is trying to do. By basing signatures on in-depth research of potential attacks and vulnerabilities, signature sets can be developed that will include general and specific signatures in order to identify the likely source or purpose of a variety of malicious traffic.

Final Notes on Header Values

In the examples in this article, we've seen how several header values can be used to create network IDS signatures. In general, some of the most commonly used header-related signature elements are:

- IP addresses (particularly reserved, non-routable, and broadcast addresses)
- Port numbers that should not be in use (especially well-known ports for particular protocols and Trojans)
- Unusual packet fragmentation
- Particular TCP flag combinations
- ICMP types/codes that should not normally be seen

Of course, any header value can be used in signatures; these items are just more likely to be used than most others.

An issue that we haven't yet addressed is which packets we want to check. If we're using a signature that is based on header values, which packets should we be checking – all of them or some of them? Well, it depends. Because ICMP and UDP packets are connectionless, odds are that under most circumstances, you'll want to check each of them. Since TCP is connection-oriented, sometimes you can just check the first packet in a connection. For example, certain characteristics such as addresses and ports will remain constant in all packets in the connection, so they can just be checked once. Other characteristics such as TCP flags should be different among the packets in the session, so if you're looking for particular flag combinations, you'd want to check for those in every packet. Of course, the more packets you check, the more time and resources it will take.

You may be wondering why we have been focusing on values in IP, TCP, UDP and ICMP headers, and have not mentioned others such as DNS. The reason has to do with the way that packets are structured. Remember that TCP, UDP and ICMP are all IP protocols, so their headers and payloads are located inside the IP packets' payload portion. In order to get TCP header data, for example, we first need to parse the IP header so we can determine that the payload is TCP. Other protocols such as DNS are contained inside UDP and TCP packet payloads, so we have to go through two levels of parsing (IP and UDP or TCP) in order to get to them. Also, it takes far more work programmatically to decode many of these protocols, as compared to the relatively simple structure of TCP, UDP and ICMP headers. This decoding is what really differentiates the protocols, as many network intrusion detection systems lack the ability to do much protocol analysis. We'll look much more closely at protocol analysis in future articles in this series.

Conclusion

During this article and the previous one, we have learned some of the basic concepts of network intrusion detection signatures, particularly the difficulties of signature development. Signatures are moving targets of sorts, because attackers can easily modify tools, which causes the characteristics of their traffic to change. By utilizing two approaches - using specific signatures that look for traffic from specific tools or particular exploits, and general signatures that just look for anomalies in traffic - an intrusion detection system can identify known and unknown attacks. Various intrusion detection systems offer vastly different signature sets, as well as signature writing capabilities and features.

We've also looked at the importance of IP protocol header values, particularly TCP, UDP and ICMP, in network intrusion detection signatures. Many different header values are used in signatures, both to limit the amount of data that will be evaluated and to identify characteristics of anomalous behavior. Although header values are important, there are much more interesting and complicated areas of intrusion detection to learn about. In the next article in this series, we will discuss protocol analysis and will look at signatures that are based on the contents of TCP, UDP and ICMP payloads - protocols such as DNS, FTP, HTTP and SMTP.

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 One](#)

Karen Kent Frederick, SecurityFocus

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

Karen Kent Frederick, SecurityFocus

[Privacy Statement](#)

Copyright 2006, SecurityFocus