

# Evaluating Network Intrusion Detection Signatures, Part Three

*Karen Kent* 2002-12-18

## Evaluating Network Intrusion Detection Signatures, Part Three

by Karen Kent

last updated December 18, 2002

---

In this three-part series of articles, we are presenting recommendations that will help readers to evaluate the quality of network intrusion detection (NID) signatures, either through hands-on testing or through careful consideration of third-party product reviews and comparisons. The [first installment](#) discussed some of the basics of evaluating NID signature quality, as well as selecting attacks to be used in testing. The [second installment](#) concluded the discussion of criteria for choosing attacks and provided recommendations for generating attacks and creating a good testing environment. This article will wrap up the series by examining other ways of generating attacks with other security-related tools and by manually creating your own attacks.

### Determining Signature Quality

When you want to choose a network intrusion detection solution or you want to see how well your already-implemented NID solution is working, you should pay close attention to the quality of the NID signatures. This is much more difficult than it sounds, as there has been little academic work in this area and commercial efforts to judge signature quality have varied widely. The tools most frequently used to determine signature quality are:

- **Commercial reviews** - Historically, most commercial reviews have focused on the usability of NID products and have paid little attention to accurately assessing signature quality. A new IDS evaluation effort by Neohapsis called [Open Security Evaluation Criteria \(OSEC\)](#) is starting to provide a highly detailed and open process for objectively measuring signature quality. Currently the OSEC is focused on testing more basic components of NID signatures, but additional phases of the OSEC are scheduled to be released in the coming months. What sets the OSEC apart from other efforts to evaluate intrusion detection systems is that they publish detailed information about each test that they perform, report whether each IDS they tested passed or failed each test, and provide additional information as needed on testing (i.e. false positives and incorrect alerts from each IDS).
- **IDS testing tools** - You can purchase specialized tools such as [Blade Software's Informer](#) that are designed solely to test intrusion detection systems. Although such tools

certainly provide benefits to their users, they are too expensive for use in most environments. More importantly, IDS testing tools have acquired a reputation, deserved or not, for having attack suites that are small, outdated, and contain attacks that do not function properly. IDS testing tools can be helpful, but be sure before you purchase such a tool that it really meets your requirements and expectations.

- **Downloaded exploits** - You can download exploits from many different hacker Web sites and mailing lists on the Internet. Unfortunately, there are several risks involved with doing this: most significantly that you have no idea if what you have downloaded does what it is supposed to do. The exploit may be broken, may contain backdoors or other malicious code, or simply may not correctly perform the attack it is supposed to. Only the most knowledgeable, experienced and cautious NID evaluators should consider downloading and running random exploits.

So if you want to determine signature quality, and you cannot find sufficient information on quality from other sources, what's the best option? In many cases, you can make a fairly fast and generally accurate assessment of signature quality by generating attack traffic yourself. For the types of malicious traffic that are most commonly seen, this is generally an easy process, and best of all, requires no investment other than a little time.

## Tools for Evaluating Signature Quality

You may have used vulnerability scanners such as CyberCop, ISS Internet Scanner and Nessus to scan your hosts for vulnerabilities. However, these scanners can also be valuable in evaluating signature quality. As part of the process of finding vulnerabilities, they perform various scans and reconnaissance. Although many vulnerability scanners do not issue actual full-fledged attacks against systems, the types of reconnaissance they perform are consistent with the majority of malicious traffic seen in real-world networks.

[Nessus](#) is a vulnerability scanner that is particularly well suited to evaluating NID signature quality. Nessus is a free scanner that contains a wide variety of scans, probes and attacks – real attacks such as buffer overflows. The checks and tests that Nessus performs are frequently updated to include the latest vulnerabilities and attacks. Also, you can write your own tests using Nessus's scripting language, so if there are particular things that you want to test for that Nessus doesn't already include, you can do so.

Port scanners make up another category of tools that is useful for NID signature evaluations,

with the most popular example being [Nmap](#). Obviously you can use Nmap to test the port scanning detection abilities of an intrusion detection system. But Nmap can also be used to test other IDS capabilities. For example, Nmap issues TCP packets with various illegal flag combinations, and it can also perform RPC scanning and reverse ident scanning. All of these activities will cause various alerts to be triggered on most intrusion detection systems.

As Web-based attacks have become more frequent over the last few years, Web vulnerability scanners have become popular. These scanners attempt to identify Web vulnerabilities by looking for default directories and files, as well as files with known vulnerabilities. The best-known Web vulnerability scanner is [whisker](#). It checks for hundreds of various Web site vulnerabilities, so it's an easy way to see how well your intrusion detection system can identify Web reconnaissance. Another benefit of whisker is that it incorporates several IDS evasion techniques – ways to disguise the true intent of malicious activity from intrusion detection sensors. So you can run whisker with various IDS evasion methods (and combinations of methods) and compare those results to how well your NID detects the non-evasion form of the same activity. This shows you how well your NID handles Web-based IDS evasion techniques.

Of course, there are many types of IDS evasion techniques. There are quite a few ways to attempt to avoid IDS detection by performing games at the packet level, such as unusual fragmentation, duplication, overlap, sequence and timing. [fragroute](#) was written specifically to perform such evasion techniques to test IDS capabilities, as well as to test the actions of operating systems' TCP/IP stacks and networking devices such as firewalls. Historically, many IDSs have not been able to correctly reassemble all types of traffic, particularly certain forms of fragmentation and TCP/IP stream reassembly evasion. Although IDSs have greatly improved in handling these types of conditions, some IDSs almost certainly still do not correctly handle all these evasion forms, particularly when they occur in interesting combinations.

Besides the tools mentioned in this section, there are many other great tools out there that let you test certain facets of intrusion detection sensors' capabilities. However, you may find it difficult or even impossible to identify and gather all the tools you need to perform a thorough evaluation. There is a final way of evaluating NID signature quality that ensures you will have every single test that you want – but it's not for the faint of heart. You could actually create your own suite of scans, probes and attacks.

## Creating Your Own Attacks

It's very unlikely that you will ever want to exhaustively test all the capabilities of intrusion detection sensors. Simply put, there are so many signatures, anti-evasion techniques and other elements in current NID sensors that the task of evaluating them all is unbelievably large. However, it is quite likely that you may be interested in knowing how well the IDS you already have deployed – or the IDS products that you are considering purchasing – handle attacks related to a certain application protocol. Suppose that your company's main interest in deploying intrusion detection is to protect a farm of anonymous FTP servers from external attacks. Then you may be primarily interested in the NID sensors' abilities to detect FTP-based attacks. How can you evaluate this?

With text-based protocols (FTP, HTTP, SMTP, etc.), there are three main options for generating your own attacks: application client programs, telnet, and scripting. Note that as discussed earlier in this series, you should only issue attacks in an isolated test environment unless you are absolutely confident in what you are doing – and even then, you still shouldn't do it.

- **Issue attacks from an application client program**, such as an FTP client or a Web browser. In many cases you can simply type the attack right into the client. If you want your IDS to identify an FTP scan that is looking for default directories, then you can simply try CDing (CD = "change directory") to a default directory from your FTP client. Sounds simple? Well, there's a catch – a big one. Many client programs modify the data that you give them before they send it to the server. If you are attempting a directory traversal attack by putting `../../../../` at the beginning of an FTP pathname, your friendly FTP client may strip this sequence out before sending your command to the server. From your viewpoint, you've issued the attack, but the IDS will never see it because the client altered it. We'll discuss a way to mitigate this risk shortly.
- **Use telnet to send attack text to a server**. You can use telnet to transmit text to any application port. Telnet really can take the place of a Web browser or an FTP client program – if you are very knowledgeable about the application protocol in question. If you know FTP commands, for example, you can telnet to an FTP server's TCP port 21 and send FTP commands to it through telnet. This technique gives you much greater control over the attacks that you send than using an application client does. Also, it's possible to send non-text values through telnet, so if you want to see if your IDS detects binary data in an FTP pathname, you can do that through telnet.
- **Script your attacks and run the scripts**. If you are really ambitious, you may want to script your attacks. [Expect](#) is a great tool for doing this: for several years, it has been used by IDS researchers to create attacks. What's nice about Expect is that you can

simulate both client and server activity. A common set-up is to have two hosts acting as a client and a server. Script the client-side of the attack activity on one and the server-side of the attack on the other. Have [Netcat](#) listen on appropriate ports on the client and the server, and configure the scripts to utilize these ports. When you run Netcat and the scripts, the attack takes place between the scripts – you don't need to install the real application's client or server software on the hosts. Again, this is an advanced technique for those who know the protocols and applications extremely well, but it can ultimately be a big time and resource saver.

Non-text-based protocols, such as DNS and SNMP, provide us with greater challenges in creating our own attacks. Some attacks can be created using client software, while others will have to be manually created. You may need to use packet-crafting tools to create your own packets or alter existing ones. Packet crafters may also be quite valuable in testing other IDS capabilities, because they give you the power and flexibility of making each packet's characteristics exactly as you would like. A detailed discussion of crafting packets and making non-text-based attack traffic is outside the scope of this article.

There's one more catch to consider in creating attacks. In order to generate your own attacks, you must know what attacks you are interested in, and what their characteristics are so that you can duplicate them. This will require some research on your part. Resources like the [Common Vulnerabilities and Exposures \(CVE\)](#) list and the [SecurityFocus Vulnerability Database](#) are extremely helpful in identifying protocol and application-specific vulnerabilities and attacks.

Finally, there is a powerful trio of tools that is invaluable in evaluating NID signature quality, whether you are creating your own attacks or utilizing others' tools:

- [tcpdump](#) can capture your attack traffic. This provides a record of the testing that you have performed.
- [Ethereal](#) can analyze the traffic captured by tcpdump. This allows you to confirm that the intended attacks occurred correctly.
- [tcpreplay](#) can replay the traffic recorded by tcpdump. This allows your testing to be easily reproducible. You can perform testing on an IDS sensor, tune and tweak the signatures, and then replay the attack traffic using tcpreplay to see if the results differ.

## Summary

In this article, we have concluded our examination of methods to evaluate network intrusion detection signature quality. Specifically, we have looked at various tools that can be helpful in the evaluation process. Some of these tools contain suites of scans, probes and attacks, while other tools are useful in testing an IDS sensor's resistance to evasion attempts. Other tools are useful in creating your own attacks. If you are interested in assessing the quality of your NIDS' signature set, or if you are considering purchasing an intrusion detection solution, the tools and techniques presented in this article may give you the information on signature quality that is difficult to find elsewhere.

*Karen Kent is a Senior Intrusion Detection Analyst for the Global Security Operations Center (GSOC) at EDS in Herndon, Virginia. She holds a Bachelor's degree in computer science from the University of Wisconsin-Parkside and a Master's degree in computer science, focusing in network security, from the University of Idaho. 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 "Inside Network Perimeter Security: The Definitive Guide to Firewalls, Virtual Private Networks, Routers, and Intrusion Detection Systems".*

## Relevant Links

[Evaluating Network Intrusion Detection Signatures, Part One](#)  
*Karen Kent, SecurityFocus*

[Evaluating Network Intrusion Detection Signatures, Part Two](#)  
*Karen Kent, SecurityFocus*

[Privacy Statement](#)

Copyright 2006, SecurityFocus