



endace  
accelerated

WHITEPAPER



# TO BENCHMARK OR BENCH PRESS?

## SNORT® PERFORMANCE TESTING

→ [WWW.ENDACE.COM](http://WWW.ENDACE.COM)  
[enquiries@endace.com](mailto:enquiries@endace.com)

The performance and capacity of Snort® IDS implementations has proven to be an extremely difficult thing to qualify and quantify. Just measuring the performance of a basic Snort device has, in itself, shown to be a near impossible task. Add to this the range of concepts being put forward to accelerate this powerful IDS solution and its clear to see why security professionals are left dazed and perhaps just a little confused by the countless claims and promises to make their favorite open source security application run just a little more efficiently.

### > THE WARM-UP

Let's just consider how we would benchmark a Snort installation. Some vendors and consumers, alike, consider a good benchmark as simply seeing how much traffic they can push through a Snort sensor without dropping packets (as reported by the IDS itself). While, at first-pass, this may seem reasonable, this direct approach is not a valid testing methodology.

Snort performance and capacity is a combination of a multitude of factors. The most obvious is the ruleset that is applied to the traffic being processed. A ruleset with one rule that says "pass all regardless" is going to produce nearly no load and thus allow more packets through the sensor faster. Naturally, while this is a meaningless statistic, there are actually still some that publish such performance numbers with a single rule applied. If you see a performance report that touts 10 gigabits of traffic without a testing background describing which ruleset was in place, there probably wasn't one.

Assuming we do want to test with a realistic ruleset(!) what rules should we use? Naturally, there are very few sensors deployed around the world which run the exact same ruleset. Nor should they. Indeed, this per-segment / per-threat customization is a fundamental concept in IDS. Rulesets must be tuned and modified for the environment they're protecting and the traffic they anticipate encountering. Additionally, the rulesets must reflect the tolerance of false positives, and the stance on policy and user behavior present within an individual organization or department. In general, the more policy issues you're interested in enforcing or identifying, the more rules you will, naturally, end up running.

### > WILL A MUSCLE STRAIN KEEP YOU OUT OF THE GAME?

So, already, we reach an impasse. Outwitted by one un-definable variable: The ruleset. Lets suggest for now, however, for the sake of this discussion, that there was some way to overcome this issue. We must next consider what type of traffic we should drive into the Snort IDS server in order to provide a comprehensive test of its throughput capabilities. Did we mention the fact that all sensors see different types of traffic?! Checkmate. Game, set and match goes to the IDS environment.

Furthermore, an often forgotten (or overlooked) fact is that the type of traffic, along with the concentration of certain protocols, will also have a significant effect on load. An environment that is heavy on netbios or HTTP will produce more load than an environment full of telnet traffic. Even within clusters of identical protocols, connection length, data flow, and transactional complexity also have major consequences. A traffic flow of all HTTP at around 500Mb/sec could be either a very high load or very small load, depending on whether it consists of hundreds of thousands of very small transactions, or a small number of very large file retrievals.

While load, in terms of traffic type, has a lot to do with the number of rules applied, it is actually more to do with the deployed Snort preprocessors plug-ins relevant to those protocols and their configuration. Since a high percentage of malware and attacks are netbios and HTTP born, a large number of rules are present to analyze these protocols. In the case of netbios, the protocol is extremely complex, contains many variables and has massive data transfers interspersed with header and administrative overhead traffic. Any traffic used to test that is heavy on HTTP or netbios will incur a greater load. But these protocols are also the vast majority of traffic a sensor will process in the real world, so they must be a significant part of a benchmark. Quite a predicament. Even when simply taking into account just these two measly protocols.

At this point, your average IT professional is likely thinking that it must be easier to bench press 300 pounds (136 kilograms) than to benchmark Snort. Fortunately, either way, Endace is here as your spotting partner.

CONTINUED →



corporate headquarters  
☎ +64 9 262 7260

usa  
☎ +1 703 964 3740

asia pacific  
☎ +65 6744 1832

emea  
☎ +44 1223 370 176

## > BENCHING LIKE A PRO

A possible solution, even when just considering just these two protocol variables, is to generate 'real' traffic on all ports and all protocols. The roadblock to this approach, however, is the tools which are typically employed to create such data flows.

The majority of network traffic generation devices are designed to test network hardware (such as layer 3 / layer 4 routers, switches and firewalls) and are therefore built to generate the maximum possible number of packets at different sizes and on many ports. What's inside the packet's payload simply isn't considered at all important - just that there's something in the packet. This is great for testing these 'low-touch' devices, concerned with only network routing and switching, but is meaningless in terms of a high-touch deep packet inspection (DPI) device, such as an IDS.

One of Snort's great challenges is preprocessing - the act of reassembling streams, detecting and kicking out bad packets. If a packet doesn't belong to an established stream, conform to the fragmentation norm for that stream or have a valid checksum, Snort should just drop the packet, assuming it's either garbage or an attack. While this is technically the case, in reality such traffic is, more often than not, simply trash. As any packets outside of established streams will mean nothing to the intended recipient, they are also considered junk.

In order to produce as much traffic on as many ports as fast as possible, typical traffic generators, pump-out what are, in IDS parlance, trash packets. We can push all sorts of packet debris at Snort out of such traffic generators and Snort will immediately discard it - all while reporting to us a spectacular traffic throughput rate. An utterly meaningless result as none of this traffic has actually gone through the processor-intensive tasks of reassembly or signature matching.

The fact of the matter is, to truly test a Snort configuration or hardware setup we must have a variety of real, fully-populated, packets, with genuine attacks. The traffic should also represent, as much as possible, the environment on which the sensor will be deployed. The guy shopping for a sensor to protect a high capacity web farm isn't going to care much about its performance when encountering netbios traffic. The results should also be reproducible, ruling out the chance that the IDS caught a lucky (or unlucky) break by detecting more (or less) anomalies than it would on average. And don't think even the smallest change in any of these areas will not make a difference. When benchmarking, it's important to generate a baseline then re-test with every desired variable.

## > ENDACE: YOUR SPOTTING PARTNER

In independent testing, the NinjaBox-Z was repeatedly shown to accelerate Snort by a magnitude of 16x over typical server deployments. Using a VRT ruleset from Sourcefire® typically seen on larger network sensor deployments, along with genuine traffic and attacks, the platform consistently demonstrated ability to fully process all data originating on a highly-populated 10Gb Ethernet segment.

Indeed, it is test and measurement technology from Endace which enables the ability to effectively benchmark IDS deployments. By capturing and accurately time-stamping (in hardware) all live data from a network, Endace can then use time-release techniques to replay those packets out to the segment not only in the sequence they arrived, but also at the exact same time interval. This unique capture/replay capability provides the most authentic view possible of historical network events.

Like the bench press, benchmarking is a weighty exercise and when it comes to Snort acceleration, the NinjaBox-Z is a proven world class performer.