

The Future of IDS

Matthew Tanase 2001-12-04

The Future of IDS

by *Matthew Tanase*

last updated December 4, 2001

Introduction

Writing about the future is a risky venture. More than likely, one will end up wrong, or worse yet - so far off that memories of the forecast bring waves of embarrassment. There is, however, the slightest chance of success; lucky for me then that the task at hand, a discussion of the future of Intrusion Detection Systems (IDS), is a bit easier to model.

IDS, much like the security industry itself, has grown rapidly over the past few years. These tools have become essential security components - as valuable to many organizations as a firewall. However, as in any environment, things change. Networks and crackers are evolving fast, demanding that security tools keep up. Intrusion Detection Systems face several daunting, but exciting challenges in the future and are sure to remain one of our best weapons in the arena of network security.

Past and Present

An Intrusion Detection System monitors machines or networks for anomalies, attempted breaches, compromises and general misuse. For many, myself included, the term IDS immediately conjures up images of network-based tools like Snort, Shadow or any one of a myriad of commercial packages. It's interesting to note then, that the earliest forms of IDS can be traced back over 20 years. These initial programs focused on discovering security incidents via recurring reviews of system audit logs and accounting files - much like modern log analyzers such as Swatch. These programs evolved over time, incorporating real-time log monitoring and more elaborate system checks, and are now referred to as Host-Based Intrusion Detection Systems (HIDS). More recently, we have witnessed the rise of the Network-Based Intrusion Detection System (NIDS). These applications began as packet sniffers that ran captured data against a set of rules or filters, flagging those that appeared to be malicious in nature. Today, NIDS signatures are created and released quickly in response to new exploits, serving as one of our best defenses. Many of us juggle elaborate rule-sets to catch the latest, greatest attacks as

well as the classics. Which brings us to present day.

Problems for the Modern IDS

Most network managers can attest to the rapid rise of the switched environment. Once an expensive alternative to the ubiquitous hub, switches now dominate both the commercial and home networking markets. And who's complaining? Switches offer better performance in most situations and protect against packet sniffers by sending traffic only to intended ports. Hubs, on the other hand, simply copy all information to every available port. Since NIDS are packet sniffers at their core, the switched network creates a definite problem. There are countermeasures, but they bring additional complexity and costs. High-end switches incorporate management and configuration consoles that include port mirroring and spanning options, copying all data sent to a range of ports to one designated as the mirror. The IDS would then be connected to the special port. In theory, this works great, but the reality is a bit more harsh. A network with heavy traffic will quickly bog down the switch itself. Even worse, the IDS will miss packets if the mirrored traffic is higher than the amount allowed by the designated port. Companies such as Shomiti Systems provide excellent hardware solutions, network taps, which solve the full-duplex problems but raise the setup cost.

A big reason many networks operate in a switched environment can be attributed to the performance benefits. This speed increase is necessary due to the overall increase in traffic volume that most organizations are experiencing. Obviously, there are limitations to how much traffic can be handled by an IDS. These performance claims are often some of the strongest selling points for commercial products, some of which can handle gigabit levels. However, with more traffic comes the bane of IDS - false positives. Anyone with Intrusion Detection experience has waded through alerts and logs, and analyzed the flagged traffic, only to discover that a sizeable percentage was harmless. The corollary risk of this is that the admin will skim log files so quickly, thanks to numerous false alarms, that they miss the real threats. And how do we reduce these nuisances? Some narrow their filters via precise tuning, while others expand the rule-set to ignore the offenders. What a cycle - more traffic brings an increase in false alarms, but more rules hurt performance. Moreover, specific filters miss subtle variations of old attacks, while generic rules increase mistakes. It's enough to make a security guru scream!

IDS performance issues are common complaints. We all have false positive horror stories. Thankfully, speed is continually enhanced by vendors and open source developers, as are false

positives. ISS RealSecure 7.0 is hoping to eliminate false positives altogether, saving organizations time and money. However, a more serious problem threatens the very existence of NIDS - encryption.

How many of you couldn't live without SSH? Me too. The dream of universally encrypted traffic is rapidly approaching. IPv6 and large wireless networks spell trouble for the payload analysis used by most NIDS. The signatures many of us depend on won't help much if they can't be matched against any data. So what does this mean? Well for one, no more false positives! But should we just forget about the concept of Intrusion Detection? Of course not, lets look at how future products will tackle all of these challenges.

Immediate Future

Let's address the two "easy" problems first, switched networks and traffic increases. I have no doubt that IDS vendors (commercial and open source) and hardware will keep pace with network traffic increases. The top performing packages will come at a steep price, as will the hardware they require, but the organizations that demand such results can usually afford it. As we discussed earlier, there are already some workarounds for NIDS in a switched environment. I also look forward to more applications such as [Hogwash](#) (based on [Snort](#)), an in-line "packet scrubber". A device like this could sit invisibly between two networks and monitor all traffic exchanged, regardless of switches or hubs, while remaining immune to attack attempts. The future is off to a bright start.

A more challenging problem being addressed right now is analysis and correlation. No matter how good the IDS analyst, we are only human. Abuse such as slow port scans are difficult to detect, especially on large networks. Projects such as Spice and Spade are working to make it easier. Acting as anomaly detectors, they examine strange packets and look to group them using sophisticated statistical analysis. Tools like these aid the ID analyst by pulling out "needles in the haystack" and bringing them to our attention. It's much easier to discover and categorize patterns when you have all the relevant data in front of you, without the noise that generally follows. Furthermore, applications similar to these will be used to fine tune filters and rules in order to reduce false positives, over time providing a kind of IDS feedback system, based on administrator input and response.

Long Term

While attending an IDS wish list discussion at the Baltimore SANS conference, I noticed that one topic dominated the conversation: correlation. It's no coincidence then that it's the ultimate theme of this article: the future of IDS lies in data correlation. The IDS of tomorrow will produce results by examining input from several different sources. I believe the notion of NIDS and HIDS will disappear, leaving us with a group of distributed components performing specific tasks. I'll explain just what each of these components needs to do in order for this model to succeed.

The concept of HIDS plays an important role in this scenario. Encrypted traffic demands that we shift packet analysis, an important part of ID, to the host. It is difficult to find another solution. There is however, a distinct advantage gained by using this method of analysis. The signatures can be tailored to one host, as opposed to the heterogeneous mix of Microsoft, Unix and application specific rules in place on most NIDS. Moreover, a recurring scan could quickly monitor which services and programs run on a machine, allowing for an even more precise rule-set. So, instead of a sensor capturing all traffic on a network, the client machines will monitor their own traffic. Many of us are already doing simple network analysis on personal machines. I run [Tiny Personal Firewall](#) on Windows machines, and [Snort](#) or [Shadow](#) on Unix boxes. But the client machines need to do more - tasks similar to those that current HIDS perform. Applications such as the [LIDS project](#), provide kernel and OS specific modifications which can monitor logs, administrative actions, system accounting and data integrity in real-time. Unfortunately, the results stored by such programs are usually only reviewed by an administrator when a user's machine has been compromised. In the future, this won't have to be the case, as clients will automatically report data to centralized monitoring stations. Let's look at how this will make security administrators' jobs easier.

At the center of this new IDS model lies a familiar concept - the management station. This box, similar to current NIDS consoles in a multi-sensor environment, will analyze flagged traffic reported by the distributed client machines. But now, this IDS "server" is dedicated to analysis and correlation - no longer is it required to capture and filter data for diverse environments. This machine's sole purpose is to present a security manager with the most pertinent information culled from the individual hosts. Additionally, each client's OS monitoring results will be reported and analyzed. In this fashion, we could have an incident or intrusion "snapshot".

Imagine the scenario - an administrator is paged from the console. He or she quickly gets access to the machine only to learn that Joe's computer in the accounting office has sent an

alert - several of his system files have been modified. The console also presents the administrator with summaries and graphs of recent traffic reported by Joe's HIDS, as well as those on the accounting network. An attacker has scanned the entire network looking for susceptible machines, found one with Joe and exploited the hole in an outdated HTTP server that was left running. Even worse, Joe's machine has reported a lot of outgoing traffic directed at one particular external machine. The console's analysis has labeled this anomalous behavior as a potential DoS attack. Now our administrator has a concise report of what happened - exploit scan and successful attack, why it happened - outdated HTTP server, and the results - corrupted system files and malicious out-bound traffic. I don't know about you, but that could sure save me some time. Our administrator can quickly review the client configurations which are readily available at the console, thanks to the recurring scans taking place, learn that no other machines have the vulnerable server in place and proceed to cleaning up Joe's machine.

The future of IDS looks promising, if the above model can emerge. But how it can be done? Host machines need to constantly look for behavior (network or system) that is designated as malicious or strange. Malicious events may be discovered using pattern matching on system logs and network traffic. Since the patterns are recognizable actions or triggers, albeit evil ones, they can be found by reviewing the data and comparing it to known signatures. This type of setup exists today in virtually every firewall and IDS - traffic analysis based on distinct patterns. We just need to expand the process to analyze system events as well; several individual programs that do this are available. Abnormal events are easy to detect, but difficult to understand.

The way to solve this challenge lies in statistical analysis and predictive artificial intelligence performed on strange data sets. The management console, having received these abnormal event notifications from many clients, needs to concentrate on possible relationships, relevance and correlation. It needs to determine the likely triggers for such infrequent events. Moreover, the console will need to communicate with multiple pieces of the network architecture: firewalls, routers, switches and even different ID systems. In the future, an IDS protocol or reporting format will be a requirement: routers could relay SNMP traps and network statistics, firewalls could transfer failed packets for analysis and different ID systems could exchange findings. The possibilities are endless, leaving us with the definitive network monitor, manager and security package, thanks to the pooled efforts of each individual component.

Conclusion

I have no doubt that the IDS is here to stay, although future systems will undoubtedly take a different form than our modern day versions. The ideas presented here, while optimistic, are attainable. The mathematical and AI (artificial intelligence) concepts required for success are already being developed, tested and improved upon. SRI has a great start with the NIDES and EMERALD projects, using a distributed model similar to what is outlined above. [ISS](#) is developing products that will scan networks for vulnerabilities and modify the IDS filters based on the results. [Lancope's StealthWatch](#) uses a "flow-based architecture," which can recognize abnormal behavior. Several other features outlined above are being incorporated into upcoming products, all of which will improve with time and research.

Ultimately, I think that future IDS will merge all of the independent network components and tools which exist today, into a complete and cooperative system, dedicated to keeping networks stable. There will be many distributed elements performing specific jobs, each passing the results onto a higher level for correlation and analysis. As always, the ultimate authority will be our own judgment.

[Matthew Tanase](#) is President of [Qaddisin](#) a network security company based in St. Louis. He has studied computer security for 10 years and holds a dual degree in Electrical Engineering and Computer Science. Currently, he provides network and security consulting services for universities, start-ups, small businesses and large corporations.

[Privacy Statement](#)

Copyright 2006, SecurityFocus