

Installing a Snort Server on the IA Network

by

Jason Young

Eastern Michigan University

Abstract

The Snort server that has been set up onto the Information Assurance network is set up for dual purposes. One, it is set up for students to monitor their experiments on the network and two, because it is meant to be a hostile network, the Snort server will act as a sentry that will warn users of possible problems. In addition to this, the server should also be tamper proof in order to properly validate student findings. Tripwire and a specially hardened OS have been implemented in order to combat improper use.

Jason's Senior Project

Every network could use a monitoring server that monitors whatever packets are flowing across the wire just as a precaution. The network that we in the Information Assurance program have set up is in particular need of one as it is most likely going to have malicious things everywhere on it. This is what it is designed for. If you have a network set up to test bad code, you must have a way to collect and analyze this data or you will just have a hostile network with no real use to the academic community. This is where my project comes in. I have setup a Snort server on the network to collect all packets going through it and analyzing them. This allows for a safer network and a more useful network.

History of Snort

The concept of an Intrusion Detection system can be attributed to the United States Air Force. Particularly, one forward thinking individual named James P. Anderson. In 1980, Anderson wrote a paper called “How to use accounting audit files to detect unauthorized access”. As you can tell, this almost perfectly describes the way that a modern day IDS works. Underneath all of the fancy bells and whistles that's all a IDS does; analyzes audit trails of different programs. (Bruneau)

This was all theoretical however. It wasn't until 1984 that Dorothy Denning and Peter Neumann designed an actual system based on Anderson's paper. This system was a rule based system and was called an IDDES or Intrusion Detection Expert System. (Bruneau)

Once the idea had time to incubate in researchers minds, a multitude of different projects were born. Discovery, Haystack, Multics, and the Network Audit Director and Intrusion Reporter were all projects that detected intrusions and were based on James Anderson's

research. (Bruneau)

Description of network and Snort server

The Information Assurance network is specifically designed to be hostile to all the machines on it. We have designed it this way to test the different components of a modern day network against all kinds of malicious things that could happen. It is completely isolated from the outside world through an air gap and nothing can get in from the outside unless someone puts it there for a purpose. Components range from servers and desktops running Linux to windows to Voice Over IP phones and surveillance cameras.

Snort is a Network Intrusion Detection System that passively scans all network traffic that is routed through it to look for attacks and probes. Anything that sets it off is then flagged and an alert is sent to the administrator. If the alert matches a preset criteria that was setup by the administrator, then Snort can automatically block all traffic coming from that machine or just the hostile packets.

Future Of IDS

According to Bruneau, there are two different avenues of research that are showing promise. The first is data mining.

Like it's namesake, data mining is the practice of sifting through years of accumulated junk (information), in order to find that rare gem that you need (piece of information you were looking for). Whatever piece of hardware you have on your networks gateway, be it a firewall, router, or an IDS, it is going to take a beating. With "hacking" tools becoming easier to use and automated worms constantly querying ports, your gateway device can easily be hit millions of times a day. Not even going into the fact that this could potentially overload the device, all of this information has to be logged and categorized in order for it to be useful. Think of the analogy of finding a needle in a haystack. Now imagine trying to find a needle in an ocean full of needles and the one you are looking for is just a little bit different than all the rest. You don't

know how it's different you just know it is different.

Governments in particular have been experimenting with different data mining techniques for years. All those databases with with all your personal information like medical records and spending habits make a juicy target.

The second research avenue is more intelligent software. Whereas the goal of data mining is to produce a significant piece of information from a pile of useless data, the goal of an intelligent system is to know what that piece of information is and what to do with it.

A company called Intellitactics has created a tool that uses six steps to analyze data: “Collection and data consolidation (awareness process), normalize, classifies the assets, prioritize (understanding process) and analyze and response (appropriate response process).” (Bruneau)

Called the Network Security Manager it can be used to extract the correct information. (Bruneau)

Potential Future of IDS

The above section was the future of an IDS as predicted by an expert in the security industry. Here is an alternate look at the future of an IDS based on some of my own research and ideas.

Back in 2002, a well known security researcher named Timothy Mullen wrote a controversial paper called “Defending your right to defend: Considerations of an automated strike-back technology”. The paper can be summed up by saying when something attacks your computer like a virus, your computer would automatically attack back the same computer and use the same hole the virus did except of spreading itself, it would patch the system so it cannot be used to propagate. Surprisingly enough, this paper was only controversial to fellow security people. When described to civilians, they were all for it citing “an eye for an eye” and “what goes around comes around”. It was argued that they didn't understand the repercussions and no matter what it is still illegal to attack a computer. Which side is right? That is still up for debate

and may never really be settled. Barring things like legality and ethics, this is a very interesting concept though. If this idea ever truly took off and was being used by the majority of people, it would create an interesting environment for the Internet. Almost like a war zone. It would be hackers vs. security professionals with regular computer users caught in the middle. If a regular users computer was infected and attacked another computer, that computer would attack it back and attempt to neutralize the threat. You can see why this would be considered “controversial”.

This type of defense was actually implemented in a way. When the MSBlaster worm was first introduced, someone who perhaps read that paper or just had the same idea decided to create a variant of it but instead of creating holes, it tried to fix them. Dubbed “Welchia” this worm would actively scan computer just like MSBlaster only when it found a vulnerable machine instead of infecting it, Welchia would attempt to download a fix and hopefully inoculate the machine before it could be compromised. While this was a good idea in theory, in practice it turned out to be a disaster.

Due to programming errors, the Welchia worm actually was *more* aggressive when scanning computers and overloaded some that way. Another thing that hampered its efforts was the fact that it tried to download a fix from Microsoft's website. It wasn't exactly a small file and with all those download requests from all of the infected machines, something had to give. At first Microsoft's website kept up and it was just the individual networks that got clogged and stopped responding. Finally though, the Welchia worm hit critical mass and the website couldn't keep up with the requests.

The final reason why it was spectacularly unsuccessful is the fact that if it did get the fix downloaded and installed, it was programmed to reboot the host machine. Mission critical machines being shutdown and usually not being booted up correctly. This was the clincher in many peoples arguments as to why an automated system is a bad idea.

While studying this incident and taking into account Timothy Mullen's paper, I came across a book that wasn't really well received. Published by Syngress, “Aggressive Network Self-Defense” by Neil Archibald is pretty much a study of why striking back is generally a bad idea and could potentially put you in the hot seat with law enforcement instead of the original perpetrator. However, in one of the last chapters, there was a short story (fictional, but

everything in it was real so it could happen) about a system administrator at a company which designed efficient fuel cells. A hacker was hired by a rival who wanted to catch up on research. Unbeknownst to the hacker though, the system administrator was given carte blanche to use whatever defenses he wanted, including strike back techniques.

According to the book, there are six steps that attackers use in order to attack a network. Most if not all hackers use this methodology. I won't go into detail for each step, for that you would have to read the book, but here are the steps in order: "Reconnaissance and Footprinting, Network Mapping, Host Mapping, Vulnerability Discovery, Vulnerability Exploitation, and Web Application Hacking." (Archibald)

Based on this methodology, it becomes easier to program an IDS to look for patterns matching this behavior. Once we've identified the threat, we have to identify the *type* of threat so that we can accurately deploy our defenses. One possible scenario has four different threat types that go along with four different response types. The following is from the paper "Digital Response":

"Threat Response Chart

Non-Intrusive/ Friend or Foe	Non-Intrusive/ Foreign or Domestic.	Non-Intrusive/ In State or Not.	Non-Intrusive/ Internal or External.
Somewhat Intrusive/ Friend or Foe.	Somewhat Intrusive/ Foreign or Domestic.	Somewhat Intrusive/ In State or Not	Somewhat Intrusive/ Internal or External.
Full On Intrusive/ Friend or Foe.	Full On Intrusive/ Foreign or Domestic.	Full On Intrusive/ In State or Not.	Full On Intrusive/ Internal or External.
Out Right Hostility/ Friend or Foe.	Out Right Hostility/ Foreign or Domestic.	Out Right Hostility/ In State or Not.	Out Right Hostility/ Internal or External.

In the response category, we have four options. The first is Non-Intrusive. What this means is our response would be limited to our own computer. The different things we could do is only limited by our imagination. From locking the intruder in a virtual jail to creating a maze of folders and files for them to follow, it's all possible and legal because it is our computer. The next option is Somewhat Intrusive. This option is basically limited to setting up a honey-pot on

your machine. A honey-pot essentially records everything a person does on your computer and where they go from there. While it is not as benign as the last option, it is still well within current laws, both legal and ethical. Our third option is Full On Intrusive. Just like it sounds, this option involves digging as deep as possible in order to find out who the attacker is. Examining logs, going to police, social engineering, pre-texting everything short of actually compromising the attackers machine is used. As you may have guessed, the final option is Out Right Hostility. However, this is kind of a misleading name. What this involves is sending something like a Trojan Horse to the attackers computer in order to find out who that person is. This should only be used in certain situations.

Now that all of the responses have been identified, it is time to define the types of threats. Our first threat is either Friend or Foe. What this means is that the attack is either coming from a country that is Friendly to the U.S.A or not. By Friendly, I mean countries that have the same type of computer crime laws as the U.S.A and an extradition treaty with us. The next category is Foreign or Domestic. This is pretty much self-explanatory. you must determine whether or not the attack is coming from abroad or not. This is also important if the attacking computer is just a proxy or not. once we have somewhat narrowed it down, we have to determine if the attack is coming from in state or not. This is most useful if you are considering legal action as this can determine whether or not the F.B.I can be called in. Our last option is whether or not the attack is coming from internal to your network or not. This may seem silly, but not all “attacks” are traditional attacks. Corporate espionage is an attack and that most likely is coming from the internal network.” (Young)

The paper gets deeper into those eight things and their relationships with each other but for our purposes we won't go there. You can see that at the base level there are sixteen different combinations but if we combine different boxes we can get more than sixteen thousand combinations just in case you ever have an IDS that needs that level of detail. With each scenario programmed into the device, it becomes possible to automate a response to a threat. Barring actual artificial intelligence, this is about as close as you can get to an intelligent system.

The way Snort is set up

The way I have set up my Snort server is a unique one. Instead of running my favorite version of Linux (I always use Linux, never Windows, it is more secure) and installing Snort on top I had some things to consider. Number one would be security. Because Linux is now more user friendly, it is now a little less secure than it used to be due to unnecessary options being turned on to placate the average user. It is possible to go through and remove these but that takes a lot of time and there is no guarantee you got them all. Number two was the isolation of the network itself. I could install the operating system easy enough but I would have no way of retrieving and installing Snort as I couldn't access the internet. This severely hampered me. So I had to find something that already had Snort installed on it and was extremely hardened. After many hours of frustrating and fruitless searching, I finally came across The Honey Net Project. The Honey Net Project or THNP had taken a Linux distribution (more specifically, CentOS) and stripped it of all superfluous things I would not need. It had just enough of the core to run Snort and some other vital utilities. The result was a distribution called "roo" (of Winnie the Pooh fame).

Once I had installed "roo" I had some work to do. The first thing I did was setup Snort to work on the network. I won't go into all of the technical details but some of the things I had to set up were IP address information, remote management information, connection rate limiting, etc. Once these were setup Snort was ready to run. However, work still had to be done in order to ensure the integrity of the system. I had to install Tripwire.

Hardening and installing Tripwire

Tripwire is a security and data integrity tool that monitors certain user specified files on

a range of computer systems. Once run for the first time, Tripwire scans all specified files and creates cryptographic hashes of them while storing those hashes in a database. It then sits and scans the file for changes in the hashes. If a change is detected and was not authorized it sends an alert to the administrator to deal with the problem. Since it only stores hashes and not the actual contents of the file however, it cannot reverse the changes. This is why backups are used.

The hardened operating system with Snort and Tripwire is now up and running. The actual operating system is set up with user names and passwords and Tripwire is set up with a user provided password that I have provided. If needed I can supply these credentials to whoever needs to change the system.

Laws And Regulations

Right now, people are still being prosecuted under the Computer Fraud and Abuse act of 1986. This was the year I was born. Close to twenty two years later and we still are going by the same set of rules while the computer industry has changed millions of times. Compared to even a few years ago, the defender has been held back on what he or she is allowed to do because of antiquated computer laws. The government has not been able to keep up with changing technology.

As more and more systems come online and more parts of critical infrastructure are brought online, hackers and crackers are attacking sites based on non-traditional things. It used to be that hacking was a pure and noble thing. It was to advance knowledge and make information free. But now, there is more of a motive. Money and patriotism has replaced the old ways and makes for a more dangerous attacker.

Also, now that whole governments are online, that gives individuals or even small groups of individuals more power than ever before. With the right skills, someone could potentially bring down a whole government. It's already happened for a short while. The government needs to get a move on and update its thinking into the 21st century. There are already cyber vigilantes out there who are taking digital defense into their own hands. The Welchia worm, and overzealous administrators are prime examples of this.

Conclusion

If you talk to a vendor that sells IDS technology or any computer security technology for that matter, you will undoubtedly hear them say that whatever it is they are selling is the end all killer application and the only thing you will need to defend your networks. The reason they can claim this with a straight face is that computer security is a relatively new concept. When the founders of Internet were designing the different protocols they weren't even thinking about security. Wasn't even on their radar. Therefore no matter what new technology we come up

with, it is all dependent on the underlying technology for its base. And this base is thirty years old.

Clearly it is time for something new to be invented. Right now security professionals are playing catch up to hackers who have become more sophisticated and dangerous. Things have to be changed so that the security professionals have an advantage or even equal footing tot he bad guys. However, to go along with these new techniques we need new laws and regulations. Without these all we would have is chaos on the Internet. All it takes is one person to set an example of striking back at an attacker and once people see how well it works, everyone will want to. Then, www will not stand for world wide web, instead, it will be wild wild west.

This project is only the beginning in a series of projects designed to make better use of the IA network. Belinda did the foundation and I have installed the first piece necessary to making it a vital part of the Information Assurance program. I hope to be around to see more projects come online.

References

- Anderson, James P. (1980) Computer Security Threat Monitoring and Surveillance. New York, NY.
- Archibald, Neil. (2005) Aggressive Network Self-Defense. Rockland, MA: Syngress, Inc.
- Bradley, Tony. (2008) Introduction to Intrusion Detection. Chicago, IL.
- Bruneau, Guy. (2001) The History and Evolution of Intrusion Detection. New York, NY.
- Davis, David. (2006) Use Snort to figure out who's trying to break into your network. San Francisco, CA.
- Goodrich, Stephanie. (2008) Enterprise Data Centers. Southfield, MI.
- Honeynet Project, The. (2004) Know Your Enemy: Learning about Security Threats (2nd Edition). New York. Addison-Wesley Professional.
- Jackson, Kathleen. (1991) A Phased Approach to Network Intrusion Detection. Austin, TX.
- Jay, Vie. (2006) Where to place a SNORT Server??? New Jersey. SourceFire.
- Knowles, D, and F Perriot. (2004) Welchia Worm Hacker Definition. Welchia Worm.
- Koziol, Jack. (2007) Real-time Alerting with Snort. New York, NY. Linux.com
- Russell, Ryan. (2003) Stealing the Network—How to Own the Box. Rockland, MA: Syngress, Inc.
- Seifert, C. (2008) Roo CDROM Users Manual. New York. Addison-Wesley Professional.
- Staff, HNS. (2005) The Story of Snort: Past, Present, and Future. New York, NY.
- Wallen, Jack. (2008) Detect Intruders on your network with Snort. San Francisco, CA.
- Young, Jason. (2008) Digital Response. Ypsilanti, MI.