Global Information Assurance Certification Paper

Around Network Intrusion Prevention Systems

By Luiz Gustavo Martins Arruda

GSEC version 1.4b, option 1

Abstract:

With the increasing number of cyberthreats around the Internet, such as the fast spreading worms, the need for new technologies that could prevent these attacks from occurring is real. Hardware and software companies are investing too much in security solutions. Most of those solutions have an important task in prevention, like firewalls, intrusion detection systems, honeypots and its honeynets.
Thus, a new technology was born trying to join all resources of those well-designed but not completely successful technologies into one: an Intrusion Prevention System or IPS. Of course, as occurs with every new technology that is raised, it has its problems and solutions.
As it does with the Intrusion Detection System, the IPS has primarily two categories: network-based IPS and host-based IPS. This paper covers the network-IPS technology and its benefits and limitations, showing some of the most used security technologies.

Introduction

During the past few years, security managers were figuring out how to protect their networks with an acceptable effectiveness. They deployed a lot of new technologies including firewalls, intrusion detection systems (IDS), honeypots, and honeynets. But this is not enough when faced to dozens of new security threats that appears every week. The worst part of the story is that you don't need an amazing knowledge anymore to do a huge disaster as we can see in figure 1 [1].

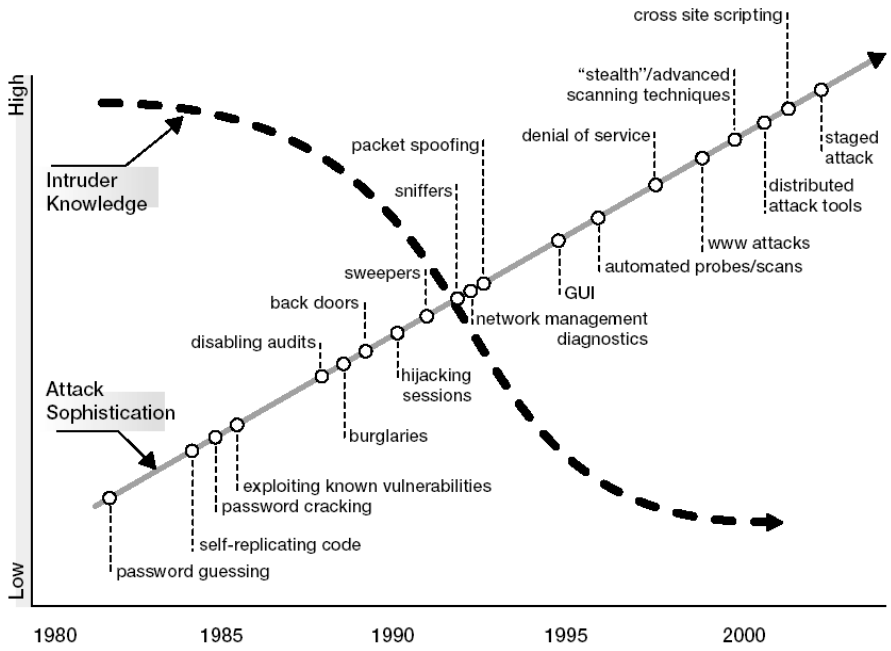

Figure 1 – Attack sophistication vs. Intruder Technology

The well-known "script–kiddies" are powered with a very large numbers of hacking tools provided by the hacker community. So, once we had to worry about hackers, now we have to worry about everybody.
As we saw recently with Code Red, Nimda and the SQL Slammer, new Internet threats can spread very fast through our systems before we can do anything about it. This last one, the Sapphire worm, achieved an impressive spreading mark. When it first appeared, it doubled in size every 8.5 seconds and reached its full rate at about three minutes, doing more than 55 million scans per second [2].
The enemy is faster than we hope they were. And they are very amazing and full of new resources that we never expect. Thus, we have no time to prepare the defense because we are used to defend using the technology of a specific attack. This makes us the turtle against the rabbit.
This scenario makes us think about how to prevent against new security threats because prevention is the key to the security race.

Firewalls

Most of IT personnel think of protection by installing a firewall, as a good option. Indeed, they are not completely wrong except for the fact that you will never know who is knocking at your door.
"A true firewall is the hardware and software that intercepts the data between the Internet and your computer. It is the TCP/IP equivalent of a security gate at the entrance to your company. All traffic (data) must pass through it, and the security guard (firewall) there allows only authorized people (data) to pass into the facility (LAN)." [3]
A firewall can do a very good job allowing and/or denying traffic in and out of your network, but this would be better as a first layer of defense [4].
A good firewall is a well-configured firewall. We know that not every firewall is properly configured. And there is another problem. The number of new vulnerabilities and the speed they spread around the world is terrifying as shown figure 2 [5].

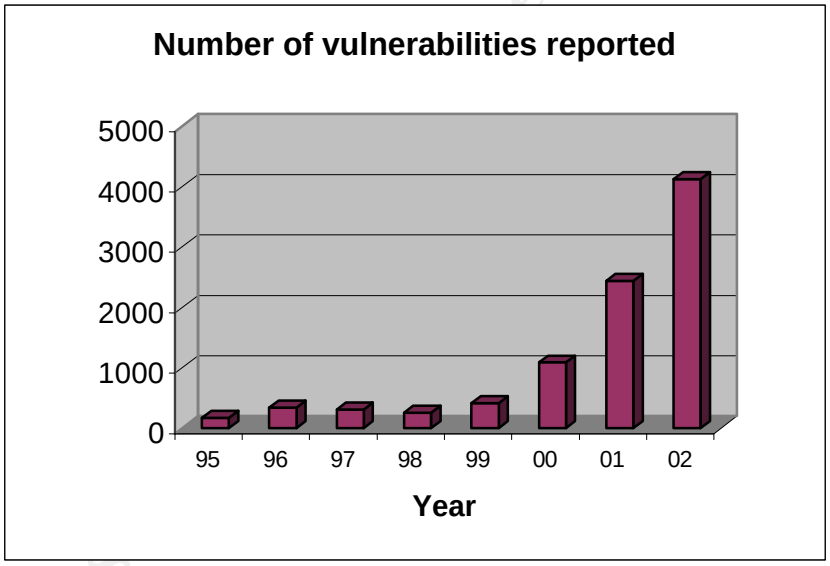

Figure 2 – Increasing Number of Vulnerabilities (CERT/CC)

So, the need for another technology was real. Even if you are an expert firewall manager, you'll probably be a victim of a new attack because you won't have too much time to change the firewall configuration before the attack occurs. The newer the attacks, the faster they spread.
A firewall cannot detect attacks neither can protect your network against them. So, the technology Intrusion Detection System (IDS) was introduced to minimize such problems.

Intrusion Detection Systems

As far as inspecting network traffic, the IDSs are similar to firewalls. The whole difference is that, instead of blocking or denying packets, the IDS look for malicious traffic that could have an attack behavior or other malicious activity, and sends alerts to a management module [6].
By scope, the Intrusion Detection technology can be split into three categories:

- Network Intrusion Detection Systems (NIDS) are software or hardware systems that analyze network packets through a network wire. They use network cards in promiscuous mode to sniff the packets as they pass through the system. They generally have some sensors and a managing console that receives data from the sensors and process them.
- Host-based Intrusion Detection Systems (HIDS) are deployed on a host computer and watch inside processes. They can inspect all types of logs (kernel, system, server, network, firewall, and more) and compare to a signature database for matching. Host IDSs can also verify integrity of data files and applications that you may add to the database. Then, the system stores a checksum (MD5 or SHA1) and a plaintext of each file so it can verify if any file has been corrupted [7].
- Hybrid Intrusion Detection Systems are a combination of the two other systems. They provide attack recognition from an outgoing or incoming traffic of a specific host. Unlike network IDSs, Hybrid IDSs don't inspect every network packet because it could decrease performance due to hard job of traffic analysis. Like host IDSs, hybrid systems also inspect events, directories, registry and data for suspicious activity. These systems are less susceptible to false positives than NIDS [8].

There is another approach referring to Intrusion Detection engines: signature analysis and protocol analysis.

Signature Analysis and Protocol Analysis

Signature analysis was is a technology based on pattern matching. The first methods of signature was simple than today. Every packet was compared byte by byte to a code string that is called attack signatures [6]. These attack signatures were stored in a signature file. So, when a match occurred, an alert was sent to the management console. As every packet was compared to every signature in the file, this engine had to be very well-designed to improve performance. Here is an example of a Snort rule that shows how this engine works:

alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86 a5|"; msg: "mountd access";)

In this signature above [9], the system will search for packets from any IP address (and any port) to any machine in 192.168.1.0/24 network at port 111 (Portmapper). These packets are identified matching the content "|00 01 86 a5|" with the packet payload. Then, an alert with the message "mountd access" is sent to the management console.

Basic protocol analysis uses an interesting engine. It wraps each packet in predefined layers and search for anomalies in these layers. If any field is filled with abnormal data, it is considered malicious and the system is warned.
Recent methods of signature analysis also use basic protocol analysis examining layers 3 and 4 of the OSI model, covering TCP, UDP and IP.
As occurred with signature analysis, the protocol analysis engine has evolved too. New engines now added the layer 7 of the OSI model, which analyzes the application protocols such as HTTP, Telnet and SMTP [6].

The good thing about signature analysis is that the rules are very simple to write and understand [6]. But on the other hand, we can point some threats that can compromise the network. As these types of systems are based on well-known signatures, if a new attack is launched, they won't have enough time to first create the signature and then spread it around the world. And even if your company already has the signature, it does take some time to configure it. Another huge problem is that someone with the source code in hands can modify such string that is part of the signature, deceiving the IDS. And even if your IDS were updated with the new signature, you could have some hosts of your network compromised. Another problem that is really boring about signature-based IDS is the high number false positives. Many of these systems are improperly configured. Then, if your configuration is poor, you will have to spend too much time analyzing fake alerts. As these false positives have to be handled manually by security managers, the performance of the system will decrease.

The problem beyond protocol analysis is a bit different. The rules made for this engine are more complicated, thus consuming more processing time. So, in terms of cost, this is not a recommended option. But, it has an important feature. Unlike signature-based systems, protocol analysis engines can detect some specific types of attack as the zero-day exploit [10]. Since these systems log real activities, they reduce significantly the number of false positives [6].

Honeypots and Honeynets

Honeypots are a security tool with multiple uses such as prevention, detection and information gathering. The mean goal of the honeypot is to be compromised. If no one tries to attack the honeypot, it's worthless. A good example of a honeypot is a system that has well-known network services running. Then, you can log all intrusion attempts and learn how the bad guys are doing to penetrate into your network [11].
Another simple example of a honeypot is a honeytoken. Honeytokens are very useful when used together with an ID system. Suppose you have a database with Social Security Numbers on it. Then, you insert a honeytoken number on it. This bogus SSN has one purpose: being stolen. When the suspicious man tries to grab this number, voilá! This method is very simple and the important thing about it is that the number of false positives can be reduced [12].

One important thing about honeypot is that once they are deployed over a network, they should never affect any network service and/or application. They have a passive purpose.
According to Marty Roesch, there are basically two types of honeypots: production and research. Production honeypots captures only limited information, and are used mostly by companies or corporations. Research honeypots are more complex, capture extensive information and are used by research, military and government organizations [11].
Spitzner, in his paper at Tracking Hackers [13] split the honeypot world in two types: low-interaction and high interaction honeypots. Low-interaction honeypots normally work with limited resources, like emulated service and systems. An example of a low-interaction honeypot is the open source honeyd by Neils Provos. High-interaction honeypots are real systems running real services and applications. Thus, the role hacking activity could be logged and analyzed, making this option better for learning hacking behaviors.
Honeynets are a well-designed honeypot network that serves the same purpose of the honeypot. The big thing about honeynets is that once you are inside the network, all your traffic will be captured and analyzed. They really simulate a real network with fully functional resources. With all the attacker steps recorded, we can easier learn everything about how they break into systems.
Another step into honeypot technology is the virtual honeynet. A virtual honeynet is a honeynet network running on a single computer. The term virtual, means that each operating system appears to be running on an independent machine. The big advantage about virtual honeynet is that you need only one machine, reducing cost and maintenance [14].

Prevention, detection and reaction

Like Bruce Schneier defined on his book, Secrets and Lies, security could break into three areas: prevention, detection, and reaction [11].
Suppose you want to protect your house from the bad guys, the best prevention scenario is to place locks all over your doors and windows, and possibly a barbed wire or an electric fence around the house area. So, you're trying to keep your house from any threat. We all know that it is closer to impossible. Indeed, you're doing prevention. But, we know that someday it will fail. So, if it really does fail, you might want to detect when the bad guy entered your house. A burglar alarm with motion sensors is the best choice for that. Now with this detection system, you will know exactly when he breaks into your house. Ok, but that's the point! What good does it to be warned if there is no reaction?
It all happens with information security systems. Thus, if prevention fails, we might have a detection system. If the detection system fails, we might have a reaction system. So, what we are trying to show here is that everything you do to prevent your system from the blackhats, is worthless because you will always running after them. On the other hand, if you have a system that reacts against an attack, you would be much more protected.

Why prevention not reaction?

Intrusion Prevention Systems, Intrusion Detection and Prevention, the names vary. The name of the technology in this case doesn't matter. Let's define this security system as any device (software or hardware) that has the ability to detect known and unknown attacks, and prevent the attack from being successful [15].
The word "prevent" used in this last definition has the sound of reaction because you are actually reacting at the very moment of the attack. Recently, some security professionals used to work with IDS together with a firewall as a Prevention System. While the IDS detect the attack, the firewall was in charge of blocking it. But, there was still a problem: latency. Some IDS and firewalls can send TCP resets as a prevention method against offending traffic [16].
"Depending on the nature and duration of the attack, it is possible that the IDS response, either in terms of modifying firewall rules or issuing a TCP reset will occur only after the attack has completed."[16]
To reduce this latency, the NIPS use the IDS module to sit in-line, working directly with the packet filtering module.

How the Network Intrusion Prevention System works?

The Network IPS is a mixture of an NIDS technology and a packet filter like a firewall. The modes of operation is similar the old IDS. The difference is that when an attack is detected, the response comes from the firewall, actually blocking that suspicious traffic. This is a very active approach because you have the reaction at the moment of the detection. The Snort-inline configured with Iptables is one good example of a Network IPS. Details of the configuration please look at Tim Slighter`s paper [17].
A basically IPS architecture may look like figure 3 [16], with four functional components: Traffic Normalizer, Service Scanner, Detection Engine and a Traffic Shaper.

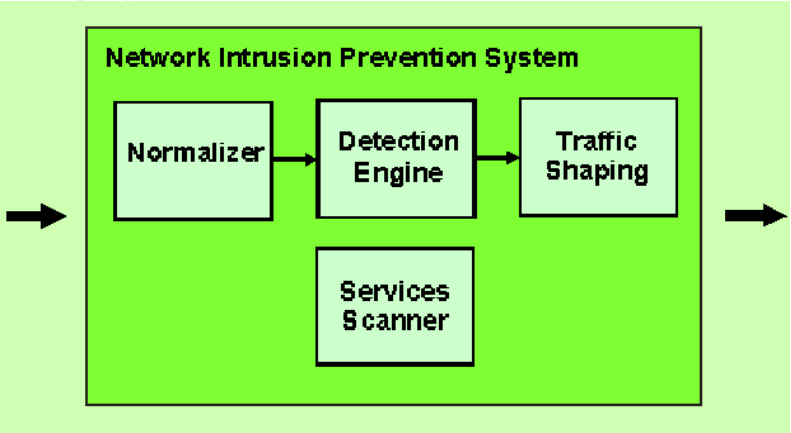

Figure 3 – The Primarily Components of a Network IPS

The Normalizer, has basically three purposes. One is to resolve and interpret abnormal traffic, whether or not having an attack behavior, like bad checksums. Second is to eliminate evasion attacks that are based on packet ambiguities. And third, to perform access control by blocking or permitting IP addresses or ports.
The Service Scanner component gathers information of the potential targets on the local network and builds a table to be used by the Normalizer and the Detection Engine.
The Detection Engine handles the pattern matching. Once the traffic is normalized, the detection can be done faster.
At last, the Traffic Shaper. This component has basically two purposes. One is classification of the traffic by protocol. Two, once the traffic is classified the Traffic Shaper can do the flow management to optimize the processing of traffic.
Despite if its interesting and effective-like design, the network IPS is not 100% attackproof. There will always be new intrusion techniques that really take us to the second place. And not all prevention system is well-deployed.

False positives in IPS: what a problem!

As we saw before, the bad thing about the Intrusion Detection Systems is the large number of false positives. It is no different from the NIPS since we have and IDS module. But, when we have an active response to these false positives, the problem is a little more complicated. Suppose you have a large network behind a NIPS and then you receive a large number of false positives. Now you have a catastrophic scenario because the NIPS denied all that traffic supposing all of those packets having an attack behavior.
Let's illustrate this problem. Suppose a company using Snort-Inline with Iptables and in its DDoS rules contains this line:

alert tcp $EXTERNAL_NET any -> $HOME_NET 20432 (msg:"DDOS shaft client to handler"; flags: A+; reference:arachnids,254; classtype:attempted-dos; sid:230; rev:1;)

Every time these packets are sent through port 20432, an alarm will be sent to the firewall. But, let's be a little more dramatic. Suppose the packet filter module is configured for blocking the source IP address. Then, if someone inside the company tries to communicate with a partner and, by coincidence, the port 20432 is allocated. Now, the firewall has blocked the IP address of the company's partner. What a mess! Think of having a couple of misconfigurations like that.

Conclusion

With the increasing number of incidents, security professionals must be aware of new intrusion techniques that emerge all the time. Firewalls, Honeypots, IDSs, and even the new Intrusion Prevention Systems aren't enough. Like Bruce Schneier said: "Security is a process, not a product". There has to be a lot of intensive work.

The bad guys are out there, thinking, working, researching, trying to find a new way of developing malicious codes that mislead security systems and cause a lot of damage. Prevention is the key!
The Intrusion Prevention System, like any other recent technology, was born to be tested, inspected, improved and used. There is no other way to find bugs and new vulnerabilities. The hackers are certainly doing this job right now.
Concerning Network IPSs, a lot of time have to be spent working on the false positives. It has to be a part of the process, working with host-based IPSs, honeypots, and firewalls. It's a good way of reaching to the main point: effective prevention.

References:

[1] The CERT® Guide to System and Network Security Practices
http://www.awprofessional.com/isapi/product_id~%7BD861C120-31DF-45FC-A8EC-FA99D2605B5B%7D/selectDescTypeId~SAMPLE_CHAPTERS/st~5130B593-BAEC-49C6-B6A8-0035DFD1EA3B/session_id~%7B42B4DB8F-0B8D-426B-9D6B-59AE3A1141A3%7D/content/images/020173723X/samplechapter%5Callench1.pdf

[2] QVISION™. "The SQL Slammer worm incident". A white paper on how QVISION™ effectively alerted and defended against the threat. February 2003.
http://www.q1labs.com/qvision_slammer_white_paper.pdf

[3] Firewall.com. "A Firewall White Paper". August 1999.
http://firewall.com/cgi-bin/jump.cgi?ID=313

[4] Forescout Technologies. "The first 15 minutes: Critical Technical Considerations for Defending Enterprise Networks Against the Next Wave of Internet Threats".
http://i.nl02.net/fscout0000/?d=15&k=webhome&_m=1k%2e000f%2ea%2emfm%2e0 (Free Register Required)

[5] CERT – Vulnerabilities reported. April 2003.
http://www.cert.org/stats/cert_stats.html#vulnerabilities

[6] Tanase, Matt. "The Great IDS Debate: Signature Analysis versus Protocol Analysis". February 2003.
http://www.securityfocus.com/infocus/1663

[7] Red Hat Linux 9: Red Hat Linux Security Guide. "Host-Based IDS".
https://www.redhat.com/docs/manuals/linux/RHL-9-Manual/security-guide/s1-ids-host.html

[8] Recourse Technologies. "Intrusion Detection: Reducing Network Risks". 24 December 2001.
http://www.isp-planet.com/perspectives/ids_p2.html

[9] CERT Coordination Center®. "Writing rules and understanding alerts for Snort, a network intrusion detection system". September 2000.
http://www.cert.org/security-improvement/implementations/i042.14.html

[10] Houston, Brent. "Protect against zero-day exploits". October 2002.
http://www.itworld.com/nl/security_strat/10302002/

[11] Spitzner, Lance. "The value of Honeypots, Part One: Definitions and Values of Honeypots". October 2001.

http://www.securityfocus.com/printable/infocus/1492

[12] Spitzner, Lance. “Honeytokens and detection”. April 2003.
http://www.derkeiler.com/Mailing-Lists/securityfocus/focus-ids/2003-04/0015.html

[13] Spitzner, Lance. “Definitions and Value of Honeypots”. 2003.
http://www.tracking-hackers.com/papers/honeypots.html

[14] Honeynet Project. “Know your enemy: Defining Virtual Honeynets”. 27 January 2003.
http://project.honeynet.org/papers/virtual

[15] Desai, Neil. “Intrusion Prevention Systems: The next step in the evolution of IDS”. 27 February 2003.
http://www.securityfocus.com/printable/infocus/1670

[16] Yee, Andre. Information Security Bulletin, Volume 8, Issue 1 – “Network Intrusions: From Detection to Prevention”. February 2003.
http://www.nfr.com/publications/ISB0801.PDF

[17] Slighter, Tim. “Configuring IPTABLES with Snort-Inline”. 23 January 2003.
http://www.snort.org/docs/Snort-Inline_and_IPTABLES.pdf

Further reading:

Brindley, Adrian. “Denial of Service Attacks and the emergence of Intrusion Prevention Systems”. 1 November 2002.
http://www.sans.org/rr/paper.php?id=818

Sequeira, Dinesh. “Intrusion Prevention Systems: Security’s Silver Bullet?”. 2002.
http://www.sans.org/rr/paper.php?id=366

Cummings, Joanne. “From Intrusion Detection to Intrusion Prevention”. 23 September 2002.
http://www.nwfusion.com/buzz/2002/intruder.html

OKENA, Inc. “A new approach to Intrusion Detection: Intrusion Prevention”.
www.okena.com/pdf/IDS%20White%20Paper.pdf

Intoto Inc. “Inline Intrusion Protection White Paper”. 2002.
http://intotoinc.com/download.php?type=whitepapers&file=Inline-Intrusion-Protection.pdf