Interested in learning
more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Wanted Dead or Alive: Snort Intrusion Detection System

Distributed IDS can easily be built and supported using currently available documentation. These systems provide a value to the networks on which they exist. Used properly they can provide information to use while adjusting ACLs on network devices such as routers and firewalls. They also provide system administrators an awareness of what their servers are subjected to while on the network.

Copyright SANS Institute
Author Retains Full Rights

AD
CounterTack
CounterTack Active Forensics for In-Progress Attacks
GET THE WHITE PAPER NOW >>>

Wanted Dead or Alive: Snort Intrusion Detection System

Mark Eanes

GSEC Practical Assignment, Version 1.4b, Option 1

October 15, 2003

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Abstract

With the status of intrusion detection system's (IDS) future doubted by some and supported by others, the steps involved in building a distributed IDS are questioned. Issues with deployment and implementation are outlined in summary. Review of currently available documentation and setup of systems, while updating applications required for the builds, are conducted to determine what problems may be encountered and if solutions or workarounds exist. A question of what additional security measures may be performed is addressed, along with a means to duplicate systems inexpensively.

Overview

An emergency response plan for incident handling exists. Screening router and firewall are filtering packets permitted into the network. There is an anti-viral gateway scanning messages destined for the users' mailboxes. But what about the traffic 'permitted' into the network, past the router, through the firewall, which contains malicious code, the latest exploit against the company's web server, or the newest buffer overflow for the organization's nameserver? Is anyone aware this is happening, even as this paper is read?

Some of today's companies are still in the denial phase--'what do we have that they could possibly want?' Still some believe that just by having a firewall up and running they have more than adequate security protection for the network. To some extent, since September 11, 2001 and, more recently, the Blaster and Nachi worms and SoBig virus just this past August, companies have taken the ideas of layered security to heart (Menninger, p.8; Jacques, 3 Sep 2003). Intrusion detection systems are now becoming more prevalent. Robert Jacques reports, "Sales of intrusion detection system devices have grown strongly, despite industry commentators' claims that the technology's popularity is fading" (Jacques, 19 Sep 2003). However, ask Gartner Group and come away with the idea that IDS is a waste of time and will disappear in approximately 2005 (Haines). Needless to say, the security community loudly disagrees (SANS NewsBites). Early warning systems have not been dismantled as the result of the Cold War ending; IDS will be around for some time to come.

Which begs the question--'which IDS is best for the organization?' Snort IDS has gained ground since its introduction into the open source community and not just in the United States. The UK-based magazine LINUX User & Developer just published their review and ratings for Snort v2.0 (Masters, p56-57). It is an excellent system to bring into the network and well documented both in published works and on the web. Actually, there are several guides, posted on multiple sites, running snort on various hardware platforms and different operating systems, with or without database logging, and using a variety of front ends and interfaces for both analysis and management. The problem is what works well on an older operating system may not work the same on the newest version.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

One must know what workarounds are available to get the system running on an up-to-date OS or application.

By taking a look at a few of the guides available on the web and the currently available books, an understanding of the good, the bad, and the ugly in getting that IDS running is revealed. While mainly discussing a LINUX setup, there are areas where Windows-based IDS are also affected. Additional checklists to lock down certain aspects of the applications will be used. Though referencing quite a number of guides and information, no cut-and-paste format will be performed. By reviewing the different sections of each as we build a distributed system, it is hoped you will become better informed of the possibilities and decide for yourself what to use. If a section is accurate and complete, or just needs a little tweak to be updated, or conflicts with other information, it will be pointed out. If a workaround is available, it will be mentioned.

How to Deploy the IDS

There are three basic deployment methods to use in putting the sensors in place on the network: hubs, switches, and taps. Each of these has its own cost, advantages, and disadvantages. All these methods are discussed in great detail in a guide by Internet Security Systems' (ISS) Brian Laing (Laing, pp6-17). Jack Koziol also discusses these same methods in his book, Intrusion Detection with Snort (Koziol, pp94-102). The following table is a summary from these documents:

TECHNOLOGY	PROS	CONS
HUBS Low cost	• No modification to network to install and manage IDS • Reflexive response capability not affected	• Collisions on full-duplex will degrade throughput • Increase in collisions if IDS managed through hub • Single point of failure, when used in-line
SWITCHES Median cost	• No modification to network to install and manage IDS • Reflexive response capability not affected • No single point of failure	• Only one SPAN port per switch • SPAN port could overload switch and will impact performance
TAPS Expensive	• Fault tolerant connection • No impact to traffic flow • Prohibits direct connection	Without extra modification: • Reflexive response not supported • Cannot monitor traffic in both directions

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

The stealth, read-only, or unidirectional cable pin outs are detailed in Jack Koziol's book (Koziol, p98). Another diagram of these pin outs may be obtained from Engage Security's "IDScenter Manual v1.1 RC4" (Kistler).

Remember that when deploying IDS using the stealth cable or tap, or deploying a sensor in stealth mode (no IP assigned to the sniffing interface), you may negate the possibility of using reflexive response unless a lot of extra modifications to networking are made. Deploying the sensor with an IP address on the sniffing interface permits the use of reflexive response, but requires the provisioning of other layers of protection for the sensor itself. This protection can be afforded using a few different types of technology, the most common being a firewall.

You may be asking just what is reflexive response. It is a means of taking a predetermined action in response to a rule being triggered. SnortSam is just one type of application that uses reflexive response. Through use of a output plugin for the snort sensor and an agent loaded on the firewall, router, or another host, SnortSam can interactively adjust the access control list (ACL) to shun or block identified traffic for a preset period of time. This is not attacking back, which is illegal, but a means to possibly stop an attack before it gets started. Use reflexive response with great care, as this can be very risky given the ease in which an IP spoof attack can occur. It would not look good on a resume that you shutdown the organization's Internet access using a reflexive response (Koziol, pp303-311).

How to Implement the IDS

Being familiar with the network's topology and the types of network devices used will also help you in determining which method(s) can be used to deploy the sensor(s). Today, most networks are switch-based. Some web and content filtering software still requires a hub, though some, like SurfControl, are now moving to switches with port mirroring capability, utilizing the switch port analyzer (SPAN) port. The SPAN port is a dedicated or configurable port that receives copies of traffic passing through one or more of the switch's ports. Using the SPAN port capability is almost the same as using a hub except that ports to be monitored can added a little at a time. This is especially useful when fine-tuning a ruleset for a particular zone of the network (Laing, pp7-9; Koziol, pp98-100).

Take a look at the network starting at the ingress point from the Internet. Is there a screening router before the firewall? This small screened subnet is an excellent place to watch packets both incoming to and outgoing from the network. In the areas where a screened subnet exists, a tap will probably be the most appropriate manner in which to deploy the sensor.

Now what about a DMZ? Demilitarized zones (DMZ) can be located off the screening router, referred to as a "dirty" DMZ by Cisco, or off the firewall. This

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

area is the network demarcation point for public and private access. Publicly accessed servers--public DNS, web server, FTP server, external mail server--are located here and are usually built as bastions (systems whose operating system and applications have been hardened). Deploying an IDS here permits the inspection of packets interacting with these servers (Malik, pp26-32; Northcutt et al, pp6-7).

Finally, at the firewall, there is usually some form of address translation--network address translation (NAT) or port address translation (PAT)—being performed for the addresses accessing the Internet. This is an ideal place to set up two sensors--one BEHIND the firewall before the private addresses are translated and the other in FRONT of the firewall after public addresses have been assigned. This provides a means of tracing the path of bad packets. For example, a user has stumbled onto a web site sporting malicious code. As the outside sensor sniffs the session, an alert is generated reflecting the source address and port number along with the destination address of the user's public translated address. Unless each user is statically assigned a public IP, or a report is being continuously run to identify who was dynamically assigned to which address, research must be quickly performed to identify who is involved. Time is ticking. By having the second sensor behind the firewall, a second alert is generated. This second alert has the same source address and port number, but the destination address now shows up as a private address that should be familiar. Possible disaster can be headed off. This type of deployment works well on intranets and extranets also.

A Quick Look at the Distributed IDS

A distributed IDS is made up of three main parts: the sensor, the database, and the analyst's console (Koziol, pp76-78). On small networks, the last two may be combined. Additionally, depending on how the sensors were deployed, up to three network adapters could be used per sensor. A few quick examples should clear things up.

A deployment with one adapter means the sniffing interface and maintenance interface are one and the same. An IP address is assigned to the adapter, which means additional security must be implemented to protect the sensor. This type of deployment usually implies the use of a hub or a switch SPAN port. All alerts travel the production network to get to the database. Anyone can intercept and read these that wants to as they are in cleartext.

A deployment with two adapters usually indicates the sniffing and maintenance interfaces are separate. The sniffing interface will be in stealth mode (no IP assigned) or a stealth/unidirectional cable will be used. The maintenance interface will be on an altogether different network from the production network. Keep in mind, this means skipping across the firewall,

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

routers, and everything else put into place to secure the network. This new, separate network must be secured in some manner to prevent compromise of the production network. Although it will be used primarily to send alerts to the database, secure access to this network must be taken into consideration for those individuals authorized to manage the sensor(s) and view the database.

This last deployment may also have two adapters in-place, a sniffing interface and a maintenance interface. What distinguishes this setup from the above is the number of cables used, the prime indicator that a tap is being used. A tap can only listen in one direction per tap port. So to monitor traffic in both directions from one adapter, one tap and four cables are used to get the sniffing interface in-place. And just like the deployment above, alerts travel on a separate, private network. Management of the sensor(s) and viewing of the database are again restricted until security is worked out.

In keeping it simple and secure, this deployment will be using one adapter and layered host security. In addition to this, the alerts will be securely tunneled from the sensor through the network to the database. The database server will be secured using layered host security, yet still permits secure viewing of the database by system administrators. Management of the sensors and the database will be secured, along with the applications used on each.

A quick glance at the guides and books used is provided in summary below:

"Snort Installation Manual" by Steven Scott
Snort on RH7.3; LAMP (Linux, Apache, MySQL, PHP), ACID; Webmin used for management.

"Snort Enterprise Implementation" by Steven Scott
Snort on RH9.0; LAMP, ACID; SnortCenter used for management.

Intrusion Detection with Snort: Advanced IDS Techniques by Rafeeq Ur Rehman
Snort on RH7.3; LAMP, ACID, SnortSnarf; SnortSam used for reflexive response; IDS Policy Manager used for management; Oinkmaster used for rules update.

Intrusion Detection with Snort by Jack Koziol
Snort on RH7.3 and Windows 2000: LAMP, ACID, SnortSnarf; IDScenter, IDS Policy Manager, SnortCenter used for management; SnortSam used for reflexive response; Barnyard used to offload alerts from snort.

Snort 2.0 Intrusion Detection by Brian Caswell (Ed.)
Snort on RH8.0 and Windows; LAMP, ACID, SnortSnarf; IDScenter, IDS Informer used for management; Oinkmaster used for rules update; Barnyard used to offload alerts from snort.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Also, an article entitled, “Distributed Intrusion Detection with Open Source Tools”, by Jason Chan, is one of the most cited articles on the Internet, yet I couldn’t locate a copy on the web (Chan, pp20+). Being a subscriber to SysAdmin, one which holds onto useful articles, I still have my copy. The article should also be available on SysAdmin’s Archive CD.

While Roman Danyliw’s “ACID: Installation and Configuration” is a bit dated, it provides valuable insights, particularly to the versions of programs used (Danyliw). This proved useful when troubleshooting some the issues discovered.

Notice that among this group of references, the majority use Redhat v7.3. The sensors and database server need to be a bit more up to date. I also wanted as much interoperability between systems as possible. However, in moving to more up-to-date platforms, one must anticipate some problems with dependencies and how certain programs are compiled. Incompatibility between programs, which may occur in one version of the operating system, may or may not occur in another version. To better understand this phenomenon, I built a version 8.0 and a version 9.0 of both the sensor and server while progressing through the guides. This turned out to be one of my better decisions I’ve made.

Software to be used for each system is listed in the appendix, in the order each is to be installed. In most cases, the versions only changed from the original guides, though some additional libraries have been listed. Again, where problems due to dependencies occur, they are pointed out. Whether a workaround or solution exists, it is mentioned.

The Sensor

Start by building the snort sensors with Redhat 8.0 and 9.0 using the guidelines for the Redhat Installation in Steven Scott’s 9.0 guide. Following along is relatively easy down to “14. Select Package Groups” (Scott, 2003, pp9-10). At this point, really go through each group and determine if a particular package is needed on the box. Don’t install MySQL packages as that will be done separately. Do install the openssh, openssl, and perl packages. The option to install apache and mod_ssl now or later, as outlined in the guide, is available.

Once done with the install, go out to Redhat’s errata page (http://www.redhat.com/apps/support/errata/) and install all relevant updates and patches for the packages installed. Perl, openssh, openssl, apache, mod_ssl updates will be among the packages listed. Also, when updating the kernel, be sure to grab the kernel-source package. This will save time in future instances where a need exists to recompile the kernel due to a patch or a newer module being added. Please note Redhat’s policy for distribution support for both versions 8.0 and 9.0.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Amazing as it seems, neither of Steven's guides mention the need for downloading or installing libpcap on the snort sensor; neither does Roman's. The libpcap library is the mechanism used by snort to capture packets on the network. Without it, snort will not operate. The library can be obtained from the tcpdump site (http://www.tcpdump.org) for LINUX. The libpcap library installation must be completed before installing snort on the system. While all the books talk about libpcap, Brian's book provides the best coverage and explanation, to include installation from source or RPM (Caswell, pp65-74). Once the packet capture library is installed successfully, move on to installing snort.

Follow on through Steven's 9.0 guide with the Snort Sensor Installation (Scott, 2003, p20). Then skip to his 7.3 guide after the 'make install' for snort and pick up the instructions for the remainder of the sensor install. Afterwards, still using the 7.3 guide, install webmin (Scott, 2002, pp10-18).

Here, a slight problem with NetSSLeay not compiling properly on a Redhat 9.0 installation is encountered. In the 9.0 guide, it looks to have worked well on paper with SnortCenter as no reference to problems is noted (Scott, 2003, p21). However, on the NetSSLeay site, the following is noted:

> RedHat 9.0 is broken (complain to RedHat about that). Only supported way to build Net::SSLeay (on any platform) is
>
> 1. Make note of which C compiler you are using (possibly set your PATH appropriately)
> 2. Build openssl from sources (do not use those evil binaries or rpms or whatever packages)
> 3. Build perl from sources with same compiler and substantially same compiler flags as you used for openssl (do not use evil binaries)
> 4. Build Net::SSLeay from sources against the perl and openssl you built in previous steps
>
> Do not write me complaining about the brokenness of RedHat 9. If you do not follow above receipe, I will ignore your support request. If I get too many RedHat 9 mails, I'll add them to my spam filter.
>
> It seems the fix is simple in the end: add -DOPENSSL_NO_KRB5 to DEFINE clause in Makefile.PL. Next release will supply this automatically. Thanks for priit.randla@eyp._ee for figuring this one out. (Kellomaki)

On Steven's site, a similar workaround is noted:

> NEWS
> May 29, 2003

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

> Jack Chapin found an easy fix for the Net_SSLeay problem, "I found a simple solution that worked in my case (RH 9.0 installed with Perl & OpenSSL through RPM binaries and updated through the RH up2date utility).
>
> Entering "export LANG=C" on command line before executing the Makefile.PL did the job (I did add -DOPENSSL_NO_KRB5 to the DEFINE variable in the Makefile first though) and everything worked fine afterwards (just make sure you're in same shell through the whole make/install process). At least that's the lazy way to go. Compiling Perl & OpenSSL on same compiler, same machine/OS is the "proper" way, but the above works so far..."
>
> Thanks Jack! (Scott)

I tried using the DEFINE variable statement to no avail. I kept getting other errors. Suffice it to say, the best way is to walk away from 9.0 right now and use 8.0 for the sensors, which works without error, unless there is a compelling need to compile Perl, OpenSSL, and NetSSLeay from source code.

For more on Webmin, there is a book coming out soon by Jamie Cameron. Additionally, SysAdmin's October 2003 issue has a nice article, "Managing Services with Webmin", by Keith Pettit (Pettit, pp23-26). This is an ideal manner in which to break in new system administrators to the LINUX world quickly. It is also a very easy way to manage Snort, among other things.

The Database Server

Installation of this section is accomplished the same as for the above system beginning with the Redhat builds. Then using Steven's 9.0 guide, follow on with the Apache Installation, MySQL Database Installation, ACID Console Installation, and Accessing the ACID Console (Scott, 2003, pp9-18).

The first little glitch was on me for not noticing the boldface NOTE about typing in the passphrase for the apache server key each time the server reboots (Scott, 2003, p11). If the server reboots in the middle of the night, apache needs that key to get started again. Without it, the system will come up, but apache will not be running to provide a page of alerts. Needless to say, until I can figure a better way around this, the passphrase is empty at the moment.

Now comes a dependency battle for libmysqlclient. Steven makes a reference to an issue with MySQL-shared libraries, but does not say exactly what is wrong (Scott, 2003, p13). It turns out to be an issue with php-mysql and the need for the correct level of libmysqlclient.so. php-mysql-4.2.x and php-mysql-4.3.x require this library to be at libmysqlclient.so.10. The newer MySQL 4.x

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

libraries provide it at a higher level and thus the problem. Using the MySQL-shared-3.23.58-1 works, but I recently noticed a MySQL-shared-compat library which provides the compatibility needed by php-mysql but still allows upward mobility with the mysql library version 4.x.

Next, most of the guides and books reference an older version of Redhat and therefore some of the interworkings between applications are not quite right when trying to update these guides and still use the applications required for the builds. A couple of good cases in point are presented here.

PHPlot and ACID are used in the RH7.3 guides and books. Rafeeq's book is correct for the information it contains concerning these two programs (Rehman, pp178-184). Jack's book isn't as specific when it comes to the versions to download, so the problem is hidden until you try following the directions (Koziol, pp129-140). Brian Caswell comes right out in Snort 2.0 Intrusion Detection and makes it very clear that if using ACID version 0.9.6b22 or greater, JPGraph versus PHPlot must be used (Caswell, p316). At the time, I had not received Brian's book and so found this on a snort-user group:

> Acording to this URL
> http://www.andrew.cmu.edu/~rdanyliw/snort/acid_config.html,
> ACID 0.9.6b22
> and greater uses jpgraph instead of phplot. Try this:
>
> Download jpgraph from http://www.aditus.nu/jpgraph/
> Untar to /var/www/html/jpgraph
> Change $ChartLib_path "../jpgraph"
>
> I think that's how I did it. (Morgan)

Had I paid closer attention to Roman's guide, particularly the versions used, it would have been made apparent what was going on and I could have easily fixed the problem (Danyliw).

In a similar case, the GD application must have certain library dependencies satisfied (Caswell, p316; Danyliw; Koziol, pp125-126). Additionally, the only version of GD that I got to work, without error, was 1.8.4, which turns out to be what nearly everyone is using (Caswell, p316; Danyliw; Koziol, pp126-127). There are some conflicts with PHP extensions using the version of GD mentioned in Steven's 9.0 guide (Scott, 2003, p15). A visit to GD's web site identifies the versions involved and provides a patch with instructions, or the older version used by most to get around the problem for now. Ensure the libraries required by GD are satisfied and use non-conflicting versions of GD and PHP, then you should expect no further problems to be encountered here.

Finally, one more kick in the shins with Redhat9.0 and NetSSLeay. While not

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

installing NetSSLeay on the database server, I was hoping for the ability to manage the sensor's webmin configuration from the database server, when necessary. Try as I might, it turned out to be impossible to do. Compiler differences for Openssl and NetSSLeay, as identified earlier, may be preventing the opening of a secure connection. The changelog at webmin (http://www.webmin.com/changes.html) reflects version 1.020 use of SSL by default when openssl and NetSSLeay are installed together; version 1.080 added support for Redhat 9.0. For now, this being the only issue—connectivity between a browser on the database server and webmin on the sensor—I chose to live with this one and leave the database server on 9.0. This is a decision you will need to make for yourself. Again, keep in mind, Redhat's policy is changing how long it will support the various versions it releases.

The Console

With this piece, consider not only yourself but also the system administrators supported. In some organizations, system administrators perform duties as system security administrators and so will need to be able to manage the sensor(s) and database server. Seeing is believing and if system administrators can see the alerts, they will be more apt to address security issues involving their servers. It is their job to ensure the servers are patched and hardened as required; yours is simply to ensure they stayed informed. I see no better way of making that point.

Now How About Some Additional Security

As I mentioned before, security on the sensors and database must be layered to protect them from harm. A good first start is using the UNIX Security Checklist maintained by Australia CERT. It is very complete and questions concerning areas of the checklist can be addressed to and quickly answered by the AusCERT (Australia CERT).

Next, those alerts are still in the clear unless you've done something about it. Introducing Stunnel, a means of secure tunneling a variety of information from Point A to Point B. Combined use of Stunnel with xinetd and TCP_wrappers makes a formidable trio to keep unwanted connections out.

I used Stunnel v3.24. Version 4.x is out, but the configuration file is radically different from the simple setup of v3.x. I noticed in both Jack's and Rafeeq's books that Stunnel v3.2x was being used but command line only (Koziol, pp114-117; Rehman, p174). The question I pose: 'when the server is rebooting at midnight, who types in the command line?' Something else to consider. Stunnel must be running before snort on the sensors, or the output to the mysql database fails, then snort fails.

I searched for a startup script for Stunnel that could be used, both on client

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

and server, and found one written by Mike Miller with Jim C. Nasby on the FreeBSD Diary site (Miller). Cut and paste their script with a text editor and replace the three lines under 'start' with the following:

```
#server
#$(STUNNEL) –d mysqls –r 127.0.0.1:3306 –p /usr/local/etc/stunnel/stunnel.pem
–N mysqls –s stunnel –g stunnel
#client
#$(STUNNEL) –c –d mysqls –r databaseserverIP:3307  –N mysqls –s stunnel \
-g stunnel
```

Now, uncomment either the client or server line for the system you need to run. Following Jason Chan's setup for Stunnel, very few problems will be encountered and a better understanding of what is happening can be gained. I borrowed a little information from his article and used the '–N mysqls' for the benefit of TCP_wrappers checking (Chan, pp22, 24). There is a noticeable difference in /var/log/messages entries when using it. The startup script needs to be copied to /etc/init.d/mysqls and links made to the runlevel directories. **This is very important**—ensure the startup for mysqls occurs before the snortd startup in each of the runlevels 2-5. A kill link is not really required as the system will terminate all programs running before it reboots, but if used, it must be set after snortd is killed.

I entered the following file in xinetd to permit TCP_wrappers functionality for Stunnel, careful to modify it for client and server:

```
# default: on
# description: stunnel for mysql
#Server server_args are listed first
#Client server_args are listed second
#Comment out the one not needed
service mysqls
{
        port            = 3307
        protocol        = tcp
        socket_type     = stream
        wait            = no
        user            = stunnel
        group           = stunnel
        server          = /usr/sbin/tcpd
        server_args     = /usr/local/etc/stunnel/stunnel -d mysqls \
        -r 127.0.0.1:3306 -p usr/local/etc/stunnel/stunnel.pem -N mysqls
        server_args     = /usr/local/etc/stunnel/stunnel -c -d mysqls \
                        -r databaseserverIP:3307  -N mysqls
        log_on_success  += DURATION USERID
        log_on_failure  += USERID
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
}
```

I also did a similar file for ssh which is shown below:

```
# default: on
# description: openssshd server
service sshd
{
        protocol        = tcp
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/tcpd
        server_args     = /usr/sbin/sshd
        log_on_success  += DURATION USERID
        log_on_failure  += USERID
        nice            = 10
}
```

Webmin can be wrapped in a similar manner. The instructions for running webmin with xinetd may be found the site's FAQ page, question 21 (http://www.webmin.com/faq.html). Don't forget, webmin has IP Access Controls which I recommend the use of. A layered defense will slow the opponent down, if not stop an opponent, from taking over the sensors or database server.

As mentioned earlier, TCP_wrappers are also used; they're a part of any good checklist. Wrap ssh, mysqls, and webmin. Use 'ALL: ALL' in the hosts.deny file. Remember where the sensors are located on the network when using wrappers—is the address making the connection request using the private address or the NAT'd public address?

Place the names of all your sensors and their IP addresses in the database server's host file. This will aid in sensor names resolution in the MySQL database and ACID console screens.

Finally, I've been using this program in conjunction with TCP_wrappers for some time. Portsentry v1.1 can still be obtained, though not at the Psionic site. Follow the README and INSTALL instructions through the 5 easy steps, and then sit back and watch the logs on the Internet sensor. I ran a standalone sensor for a year on the Internet running Portsentry. When I finally replaced the box, I had 20 screens of unique IP addresses in my host.deny file. I never deleted addresses; it never repeated addresses; the system was never compromised. Psionic has been picked up by Cisco, who is now releasing the program in their Threat Response product.

There are a few other ways to secure the boxes. A copy of SANS's Securing

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Linux-A Survival Guide for Linux Security can be obtained through their web site.

Using Iptables, Bastille Linux, or some other firewall product to allow only pre-authorized connections to the sensors and database server are all viable options, some of which are addressed in the books. There are several great firewall products to chose from depending on the OS platform installed.

An area that needs a little more security applied to it is the database server. Again, after searching the web, I found a series of checklists for the Apache, PHP, and MySQL applications from Security Focus web site. Written by Artur Maj, these will ensure the information is accessible, yet the system is secure. Please read these in the order listed, The Apache checklist cannot be used to the full extent because PHP and MySQL are used. However, it does provide a means to chroot the Apache service so it runs as other than root. The other two checklists provide a means to tighten PHP and MySQL (Maj). The PHP checklist and Jack differ on which file to modify—whether php.ini-recommended or php.ini-dist. Jack goes into specific detail on which file to use and why. As the PHP checklist is written primarily with FreeBSD in mind, I defer to Jack (Koziol, pp127-128).

Finally Done, But You Need More Sensors…

How many sensors are needed? What about back-up copies for quick recovery? Here's a quick method to reproduce the systems, bit for bit the same. First, ensure the hardware is identical. I had several systems donated to the cause, identical all the way down to the chipsets. Symantec's Ghost may be used, budget and licensing permitting. Otherwise, try this.

Place a second IDE hard drive of identical size in the IDE chain as a slave to the system to be used as a master. Once this is done, boot the system, get to the command line and type the following:

```
dd  if=/dev/hda of=/dev/hdb conv=noerrors
```

A bit for bit copy is made of the disk, something I picked up through forensics. By running through the following checklist, subsequent systems with their own unique keys and settings may be setup.

Webmin – Adjust IP access control list and recreate the SSL key.

Portsentry – Adjust portsentry.ignore file, then type 'make linux', 'make install'.

Network config – Adjust the hosts file, the DNS, the hostname, IP address, subnet, gateway.

ssh – Delete all known_hosts entries in .ssh at /root and in user home directories.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Type ‘ssh-keygen’ and values for the flags (-b, -t, and –N) to generate a new key.

Stunnel – Type ‘make cert’ while in the stunnel directory. Type in information similar to this:
‘openssl req –new –x509 –days 364 –nodes –config stunnel.cnf –out stunnel.pem –keyout stunnel.pem’
Reply to the prompts for information.

TCP_wrappers – Adjust IP addresses as required.

NetSSLeay – From the NetSSLeay directory, type ‘Makefile.PL’, then ‘make install’. Reply to the prompts for information.

snort.conf – Adjust the network variables as required. In database output plugin, type ‘sensor_name=’ followed by the hostname.

mysqls startup script– Both the script and the file in xinetd should all read the same as far as addresses—no change required.

Conclusion

Distributed IDS can easily be built and supported using currently available documentation. These systems provide a value to the networks on which they exist. Used properly they can provide information to use while adjusting ACLs on network devices such as routers and firewalls. They also provide system administrators an awareness of what their servers are subjected to while on the network.

A review of IDS deployment strategies using hubs, switches, or taps and a brief discussion on IDS implementation on the network was presented. Various guides, books, and articles currently available were summarized and used, in reference, to build updated sensors, database server, and console. During this, some problems inherent with bringing systems up to the bleeding edge were encountered, such as compiler and dependency issues. Workarounds and solutions were provided, where feasible. Additional layered security measures were put into practice using various checklists and technologies to secure and harden both sensors and database server. Finally, an inexpensive means of duplicating sensors and a checklist for changing settings to provide uniqueness were provided.

It is my sincere hope that companies continue to take to heart the seriousness of providing detection for the things occurring within their networks. Sarbanes-Oxley may now be poised to press the CIO into action where he once sat idly by and asked, ‘how’d that happen?’

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Appendix
Software (listed by system, in order of install)

Sensor:

libpcap	http://www.tcpdump.org
mysql-client	http://www.mysql.com/downloads/mysql-4.0.html
mysql-devel	http://www.mysql.com/downloads/mysql-4.0.html
snort 2.x	http://www.snort.org/dl/snort-2.0.x.tar.gz
snortrules	http://www.snort.org/dl/snortrules-current.tar.gz
snortd	http://www.superhac.com/docs/snortd.txt
NetSSLeay	http://www.symlabs.com/Net_SSLeay/Net_SSLeay.pm-1.21.tar.gz
webmin	http://www.webmin.com/download.html
snort webmin_plugin	http://www.snort.org/dl/contribs/front_ends/webmin_plugin

Database Server:

Apache	ftp://ftp.redhat.com/pub/redhat/linux/9/en/os/i386/RedHat/RPMS
mod_ssl	ftp://ftp.redhat.com/pub/redhat/linux/9/en/os/i386/RedHat/RPMS
mysql-server	http://www.mysql.com/downloads/mysql-4.0.html
mysql-client	http://www.mysql.com/downloads/mysql-4.0.html
mysql-shared	http://www.mysql.com/downloads/mysql-3.23.html
(mysql-shared-compat)	http://www.mysql.com/downloads/mysql-4.0.html
create_mysql	(file is bundled within the snort archive)
php	ftp://ftp.redhat.com/pub/redhat/linux/9/en/os/i386/RedHat/RPMS
php-mysql	ftp://ftp.redhat.com/pub/redhat/linux/9/en/os/i386/RedHat/RPMS
ACID	http://www.snort.org/dl/contrib/data_analysis/acid/acid-0.9.6b23.tar.gz
ADOdb	http://php.weblogs.com/Adodb#downloads
GD	http://www.boutell.com/gd/http/gd-1.8.4.tar.gz
JPGraph	http://www.aditus.nu/jpgraph/downloads

Console:

PuTTY	http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
Openssh	ftp://ftp.redhat.com/pub/redhat/linux/9/en/os/i386/RedHat/RPMS

web browser (Mozilla, Netscape, Internet Explorer)

Additional Security:

Stunnel	http://www.stunnel.org/download/source.html
Portsentry	http://www.smittyware.com/contrib/psionic.php

NOTE: For MySQL files download identification purposes, client=client programs, server=server, devel=libraries and header files, shared=dynamic client libraries, and shared-compat=dynamic client libraries (including 3.23.x libraries).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

References

Articles from the Web

Australia CERT. UNIX Security Checklist v2.0.
URL: http://www.cert.org.au/render.html?it=1935
(28 Aug. 2003).

Danyliw, Roman. 9 Oct. 2002. ACID: Installation and Configuration.
URL: http://www.andrew.cmu.edu/~rdanyliw/snort/acid_config.html
(20 Jun. 2003).

Haines, Allison. 11 Jun. 2003. Gartner Information Security Hype Cycle declares intrusion detection systems a market failure: Money slated for intrusion detection should be invested in firewalls.
URL: http://www4.gartner.com/5_about/press_releases/pr11june2003c.jsp
(20 Jun. 2003).

Jacques, Robert. 3 Sep 2003. Blaster and SoBig change the landscape.
URL: http://www.vnunet.com/News/1143355 (6 Sep. 2003).

Jacques, Robert. 19 Sep 2003. IDS is dead, long live IDS.
URL: http://www.vnunet.com/News/1143747 (20 Sep. 2003).

Kellomaki, Sampo. 15 May 2003. Net::SSLeay Home page.
URL: http://www.symlabs.com/Net_SSLeay/ (25 Jun. 2003).

Kistler Ueli. 2003, IDScenter Manual v1.1 RC4.
URL: http://www.engagesecurity.com/docs/idscenter (5 Sep. 2003).

Laing, Brian. 2000. How to Guide-Implementing a Network Based Intrusion Detection System. URL: http://www.snort.org/docs/iss-placement.pdf
(21 Jun. 2003).

Maj, Artur. 14 May 2003. Securing Apache: Step-by-Step.
URL: http://www.securityfocus.com/infocus/1694
(3 Sep. 2003).

Maj, Artur. 28 Aug. 2003. Securing MySQL: step-by-step.
URL: http://www.securityfocus.com/infocus/1726
(3 Sep. 2003).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Maj, Artur. 23 Jun. 2003. Securing PHP: Step-by-step.
URL: http://www.securityfocus.com/infocus/1706
(3 Sep. 2003).

Miller, Mike. Stunnel – another way to avoid plain text passwords.
URL: http://www.freebsddiary.org/stunnel.php.html
(15 Jul. 2003).

Morgan, Joel. 19-Nov 2002. Problems with graphs in ACID. snort-users group.
URL: http://marc.theaimsgroup.com/?l=snort-users&m=103771488419710&w=2
(15 Jul. 2003).

Scott, Steven J. 2003. SuperHac.com:Hacking on a Whole New Level Home page. URL: http://www.superhac.com (27 Jun. 2003).

Scott, Steven J. Aug. 2002. Snort Installation Manual: Snort, MySQL, and ACID on Redhat 7.3. URL: http://www.snort.org/docs/snort-rh7-mysql-acid-1.5.pdf
(20 Jun. 2003).

Scott, Steven J. Apr. 2003. Snort Enterprise Implementation: Snort, MySQL, SnortCenter and ACID on Redhat 9.0.
URL: http://www.superhac.com/docs/snort_enterprise.pdf (27 Jun. 2003).

SANS NewsBites. SPECIAL SECTION: "IS IDS DEAD?"—Gartner IDS report evokes strong response. Volume 5 19 Jun. 2003.

Books

Caswell, Brian. (Ed.). Snort 2.0 Intrusion Detection. Rockland:Syngress Publishing, Inc., 2003.

Koziol, Jack. Intrusion Detection with Snort. Indianapolis:Sams Publishing, 2003.

Malik, Saadat. Network Security Principles and Practices. Indianapolis:Cisco Press, 2003.

Northcutt, Stephen, et al. Inside Network Perimeter Security: The Definitive Guide to Firewalls, VPNs, Routers, and Intrusion Detection Systems. Indianapolis:New Riders Publishing, 2003.

Rehman, Rafeeq U. Intrusion Detection with Snort: Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID. Upper Saddle River:Pearson Education, Inc., 2003.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Periodicals

Chan, Jason. “Distributed Intrusion Detection with Open Source Tools.” SysAdmin: The Journal for UNIX System Administrators. August 2001 Volume 10 (2001): 20+.

Masters, Jon. Intrusion detection system snort 2.0. [Review of the program Snort version 2.0]. LINUX User & Developer. August 2003 Issue 31 (2003): 56-57.

Menninger, Marc R. “September 11th Two Years Later: Is Our Information More Secure?” The ISSA Journal. September 2003 (2003): 6-8.

Pettit, Keith. “Managing Services with Webmin.” SysAdmin: The Journal for UNIX System Administrators. October 2003 Volume 12 (2003): 23-26.

© SANS Institute 2003, Author retains full rights

Last Updated: April 21st, 2015

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS SEC401 London	London, GB	Apr 27, 2015 - May 02, 2015	Live Event
SANS ICS London 2015	London, GB	Apr 27, 2015 - May 02, 2015	Live Event
SANS Bahrain 2015	Manama, BH	May 02, 2015 - May 07, 2015	Live Event
SANS Security West 2015	San Diego, CAUS	May 03, 2015 - May 12, 2015	Live Event
SANS Secure India 2015	Bangalore, IN	May 04, 2015 - May 23, 2015	Live Event
SANS Secure Europe 2015	Amsterdam, NL	May 05, 2015 - May 23, 2015	Live Event
SANS/NH-ISAC Healthcare Cybersecurity Summit	Atlanta, GAUS	May 12, 2015 - May 19, 2015	Live Event
SANS Melbourne 2015	Melbourne, AU	May 18, 2015 - May 23, 2015	Live Event
SANS Pen Test Austin 2015	Austin, TXUS	May 18, 2015 - May 23, 2015	Live Event
SANS Secure Thailand 2015	Bangkok, TH	May 25, 2015 - May 30, 2015	Live Event
SANS ICS Security Training - Houston	Houston, TXUS	Jun 01, 2015 - Jun 05, 2015	Live Event
SANS ICS410 Vienna in Association with IAEA	Vienna, AT	Jun 06, 2015 - Jun 10, 2015	Live Event
SANS Dublin 2015	Dublin, IE	Jun 08, 2015 - Jun 13, 2015	Live Event
SANSFIRE 2015	Baltimore, MDUS	Jun 13, 2015 - Jun 20, 2015	Live Event
SANS Rocky Mountain 2015	Denver, COUS	Jun 22, 2015 - Jun 27, 2015	Live Event
SANS Pen Test Berlin 2015	Berlin, DE	Jun 22, 2015 - Jun 27, 2015	Live Event
Cyber Defence Canberra 2015	Canberra, AU	Jun 29, 2015 - Jul 11, 2015	Live Event
SANS Capital City 2015	Washington, DCUS	Jul 06, 2015 - Jul 11, 2015	Live Event
Digital Forensics & Incident Response Summit	Austin, TXUS	Jul 07, 2015 - Jul 14, 2015	Live Event
European Security Awareness Summit	London, GB	Jul 08, 2015 - Jul 10, 2015	Live Event
SANS London in the Summer	London, GB	Jul 13, 2015 - Jul 18, 2015	Live Event
SANS Minneapolis 2015	Minneapolis, MNUS	Jul 20, 2015 - Jul 25, 2015	Live Event
SANS San Jose 2015	San Jose, CAUS	Jul 20, 2015 - Jul 25, 2015	Live Event
Security Operations Center Summit & Training	OnlineDCUS	Apr 24, 2015 - May 01, 2015	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced