Global Information Assurance Certification Paper

An Introduction to Snort and the Principles of Intrusion Detection for Mac OSX

Ian Martin: GSEC Essentials Course Version 1.3 (December 12, 2001)

This paper assumes the Mac user/administrator to have very little Unix knowledge, consequently there is a very significant quantity of Unix basics involved herein. I have struggled with this dilemma (I am sure it will prove awkward to assess as a SANS exam piece) but feel that if I produced a paper that accepted an 'a priory' level of Unix knowledge (tar, /compile, make, etc.) I would be alienating my target audience and therefore any hopes for the success of the paper's objectives would be unrealistic. Remember that OSX has been available for less than a year...

I therefore present an Information Security 'How-To' paper, rather more bloated than I originally envisaged, but hopefully accessible to the widest possible Macintosh User Audience.

1 Summary and Objectives

The paper begins with an introduction of the principles of intrusion detection and its applications to the Mac OSX community, there will be initial sections covering, OSX, 'Treat Vectors' and 'Defence in Depth'. It will then explain the issues surrounding NIDS and its placement on the network, including specific recommendations by the author. We will use as our example, one of the most widely respected and supported Open Source Packages [1] available, namely SNORT.
Next, the initial set-up of the Mac and a number of basic Unix commands. Those commands necessary for the first time user to create directories, expand a '.tar.gz' package, and run './config', and 'make.' The installation of 'libpcap' and 'Snort' will then be explained in detail.
The final section is 'hands-on' with Snort. The three modes of operation, the modification of its '.config' file and an explanation of Rules and Signatures. ARP and auditing with 'syslog' are used to illustrate specific facets of Snort functionality. The section ends with an OSX generated portscan to test Snorts response, culminating with a brief discussion on false positives and the importance of organisations like Incident.Org in the users defence strategy.

1.1 Mac Security, Issues? What issues?

There were some virus issues with the old Mac OS's, nVir and the likes, (remember 'Disinfectant'), AutoPlay, and of course the ubiquitous Microsoft Macro viruses, although one could argue that if you didn't run Microsoft software, you didn't get the macro viruses! so technically not a Mac virus. However the generally non scriptable OS and, lets be honest, small user base, kept the Mac happily beneath the Bad Boy Radar, and that was that. Now with OSX and the entire planet on the Internet, the Mac has become an exceedingly visible and attractive target.

"BSD is now three times as popular on the desktop as Linux", Apple's Ernest Prabhakar told attendees at the annual USENIX BSD Conference [2].

An obviously bullish Public Relations quote, referring to the massive increase in the BSD user base since MAC OSX, which uses 'FreeBSD' as a foundation for its core. However with Unix being the computer criminals operating system of choice and with the majority 'newbie' OSX user having very little, if any, Unix knowledge, things are going to get very interesting. It must be acknowledged that Apple have gone a long way to producing an 'out of the box' UNIX that is very secure (almost everything Networkable in Unix is switched off by default even the 'root' account, its a wonder it goes at all!). This is much more secure than your average Linux installation but there is still room for improvement, and details of additional security measures that can be taken with your Mac are to be found here.[3]

In case you were wondering OSX is based on a combination of the Mach kernel (3.0) developed at Carnegie Mellon University (Steve Jobs used it in his NeXT box), and operating system services based on 4.4BSD Lite 2 and FreeBSD. Certain bits for this collective mish-mash of code Apple have made into an open-source project called Darwin [4]. So now you can even contribute to the Apple legend by writing bits of the OS yourself.

2 Introduction to IDS - *Just because you're not paranoid doesn't mean they're not out to get you*

2.1 Threat Vectors

In the security world they use the term 'Threat Vector'. This is the method the 'Threat' (computer criminal) uses to get to the Target (your machine or your data). Under law a 'computer criminal' is an individual who trespasses on your network or any system owned by you, or your Organization, without prior consent. This is ANYONE who for ANY reason targets your machine's/network's and tries to gain access to it for ANY reason. From Script kiddy or Seasoned Hacker to major Business Competitor or Local/Foreign Government. A recent report by the CSI (Computer Security Institute) shows computer crime and subsequent financial losses to organizations is growing at an alarming rate. [5]

Well that's the Who covered and now the How. A good analogy would be to imagine a mosquito, this is the treat vector for Malaria, a countermeasure against the Malaria threat would be to spray the breeding ponds so you got rid of the Mosquito's entirely, (this is known as detection and response) or you decided that there are simply too many ponds out there and so went out and bought lots of mosquito netting (this is known as prevention).[6]

Threat Vectors can be outlined in the following ways and it useful to note that there are specific defense strategies dealing with each of these treat vectors, in fact, we will be looking at these next. Note: This list is only illustrative of the 'Defense in Depth' strategy and should not be taken as in any way complete. [6]

Threat Vector	Defense Strategy
Outsider attack from the Network	1 Firewall 2NIDS HIDS 4Passwords5Cryptography
Outsider attack from the phone	1 Disable Dial-In 2HIDS 3Passwords 4Cryptography
Insider attack from local network	1 NIDS 2HIDS 3Passwords 4Cryptography
Insider attack from local system	1 Passwords 2Cryptography
Attack from malicious code.	1 AntiVirus Software on Host and Mail Server
Social engineering Attack	1 Security Policy Documentation 2 User Education

A social engineering attack is like phoning a user and pretending you are the help-desk thereby obtaining a valid username and password. Or being Mr. Big at a business meeting and demanding the office modem number because you need to retrieve a document. Here's another attack.[7]

With every threat vector, the first measures you should be taking is preparing a security policy document which outlines your organizations position on everything from Password usage to who is in charge of the backups when the backup person is off. You should then make sure everyone has read and understands it. The following URL's provide guidelines on the production and delivery of security policies. [8]

2.2 Defense In Depth

Ok so imagine an ONION, with your data (or whatever it is that you are trying to protect) at the center and wrapped around and around are layers of security defense mechanisms.

The Layers

1 The first sort of defense you could have would be to have the very data encrypted so someone would need to get or crack your password or passphrase. 'Pretty Good Privacy' would be an Ideal program to investigate for protection at this level. [9]

2 Next up would be Application security this could include username and password login, as some database programs utilize.

3 User and Group Accounts, Resource Access and Logging would be critical at the OS security level.

All the above are as applicable to the potential criminal coming in over the "wire" as they are to the potential criminal 'insider' within your organization who has the freedom to walk right up and sit in your chair.

4 You may even lockup the box itself and not allow floppy, Zip, Jaz, CD, access (bootable media). You could also add Boot Passwords, and Screen Saver passwords. This security layer deals specifically with your Insider Threat.

Note: A lot of media noise is made about the biggest danger to a company's systems comes from the inside, some say as much as 90% of the entire threat and from a financial perspective this is quite true. An 'Insider' will have a greater knowledge of what and where most of your sensitive data is, be it to sell, destroy or corrupt. They are also most likely to get caught, as you generally know where they live.

BUT THE HIGHEST FREQUENCY OF ATTACKS WILL COME FROM OUTSIDE YOUR ORGINISATION.

5 Still on the machine itself we could have any or all of the following, a Personal Firewall product, or 'finger printing'/ logging' system, these we can loosely describe as Host Based Intrusion Detection, [10]. There are a variety of systems available at this level.[11]You may also have one or more AntiVirus packages on the machine.

6 Now we are loose on your Local Area network and here is where we find one of the places where we can install our Network Intrusion Detection System SNORT. The basic premise here is that the NIDS software sits on a machine on your network and, in a non-intrusive way, (without adding any additional load to the network itself) can observe ALL the traffic that passes on the entire network. It can then, using a collection of Signatures (SNORT Rules) analyze said traffic and log/report any suspicious activity, all in the blink of an eye. There are a variety of other systems available at this level. [11]

7 Finally your network Firewall, the proverbial Cerberus that connects your internal LAN to the Big Bad Internet.

Note: It is recommended practice to make sure that when configuring your firewall's access control lists (ACL's), set them to 'deny all' by default. This should be the 'fail safe' state, then just add the services you really need, this will make life really difficult for your Organizations Limewire enthusiasts.

8 What do you mean 8 I thought that was it. Well it so happens that some Organizations possibly yours included, will have more than two connections on their firewall, they may have their Mail and or Web servers on a separate network. Still connected to the Firewall, but with different Access Control Lists (Rules) to the ACL's employed on your internal LAN (generally less strict) this area is commonly known as the De-Militarized Zone or DMZ. This is also an Ideal place for a NIDS (why? see 9)

9 Yes you could also put the NIDS in the "wild" i.e. on the DMZ or outside the Firewall completely, absolutely exposed to the entire Internet. What am I nuts... well this is an option, and a lot of security professionals do this. It enables them to see a complete picture of everything that is coming at your Firewall and if you had a second NIDS sensor inside your firewall, you would be able to compare the logs and see just how well your Firewall and its ACL's were holding up.

Placement of the NID here is however outside the scope of this document. To deploy it effectively would necessitate explaining the configuration of a very secure Unix installation and a specialized 'Dumb" NIDS Sensor (i.e. Shadow, made by the US Military and available at all good ftp sites)[12] to collect the data. Then the use of a secure file transport mechanism (SSH) to send the data to Snort, (which would be sitting inside the Firewall) for analysis. [13]

Note: I am making the general assumption that most Mac based offices are significantly smaller than your average multi-national and so do not have an enormous budget to throw around. So if I were to ask one of you for a 'spare G4' to sit on the outside of your firewall (If you have a firewall... and of course you do!), doing nothing but logging Internet activity, I think your answer would be along the lines of "What! Are you joking! Do you know how much these things cost etc....

The conclusion to this Onion story is that there is no 'Magic Bullet', no 'one size fits all' monster application that will protect your organizations data from every eventuality (don't let that sales guy tell you otherwise). Teamwork has to be the order of the day, multiple security products working together at different levels within your organization to protect your data. NIDS is just one tool in your security toolbox (like a good pair of mole grips).

3 Placement strategy's for the NIDS

NIDS comes in a variety of shapes and can be installed in a variety of ways, we will be dealing with the most basic, that is, having a sensor and an analyzer on one machine.

The Common Intrusion Detection Framework has been written to help identify a set of components that together define an intrusion detection system. [14] This defines the E box, A box classifications, there is also a D box which is basically the hard disk where we write the data and a C box (counter measure) which is not a component of the SNORT system but can be found in some major commercial products. This would be, in effect like the NIDS analyzing an attack then actively modifying your firewall's ACL's to shut it out. A big problem here is that a Distributed Denial of Service Attack on the NIDS could force the C box to shut down your net entirely.

What is a sensor? What is an Analyzer?

1. Sensor (E box)

A module that does the information capture from the network. Using a particular network card on the host machine and switching it into what is called promiscuous mode initiates the capture. Promiscuous mode? Instead of the normal mode where the card is only listening for either its own name (address) on the network or a specific group name (multicast or broadcast address) that it belongs to, it listens to everything, now that's a lot of information!

2. Analyzer (A box)

A module that takes the data from the sensor and analyses it using a predetermined set of filters called rules which include the signatures of known intrusion/suspicious activity, it can then take appropriate action, i.e. messaging/paging/SMS you that a suspicious activity is taking place. Or if it has a C box then it could initiate its own countermeasures.

Note: The methodology and design of Signatures (rules) is one of the biggest issues of not only NIDS but also HIDS and IDS in general. A very interesting 4 part article on the subject is to be found here.[15]

Note: You can also have multiple sensors at various points on your network (in a large switched environment), all sending data to the one analysis machine. However, as we are Mac users, I don't think we warrant that kind of deployment, and anyway, once you know how to do one it's not twice as hard to do multiples.(got any spare G4's?)

3.1 Network Cable Systems

Another assumption, is that your Network Cabling Infrastructure will utilize UTP (good old Category3 or CAT5) Unshielded Twisted Pair. This is because the Mac (since the PPC 7200/7600) comes with one of these ports built in. Before then you had to buy an extra bit called a AAUI transceiver (Apple Auxiliary Unit Interface) box to plug the UTP cable into, anyway I digress; this issue is about the two kinds of 'distribution boxes' that UTP facilitates and their underlying difference.

Hubs Vs Switches In the red corner 'the cheep and cheerful' in the blue corner 'the fast and expensive' (but getting cheaper by the day)

Hubs

Slow network speeds generally 10 Mbs and every one shares the available bandwidth. With hub based technology all machines connected to the hub can potentially hear all the other machines conversations, like living in an apartment building with very thin walls, so its easy to snoop on other conversations, if you are in a 'promiscuous' mood.

If you are still running with Hubs then you are set. Just plug your NIDS machine in anywhere, as all points will be equally effective.

Switches

Fast network speed generally from 100Mbs to 1Gbs (will, in the majority of cases, take 10Mbs too, for printers and that cute Mac Classic you can't seem to get rid of) and all that bandwidth is yours alone. Switches give you the penthouse suit with your own private lift to the ground. No-one can here you and you can't here anyone else.

This is the Achilles heel of the NID, with more and more organizations using Switched

technology, NID sensor placement is becoming a bit of a problem. It needs to snoop on ALL the other conversations, i.e. to see all the network traffic and if it is on a switch then it can only see itself, now this is somewhat of a setback.

Note: Certain switch manufacturers are now responding to this issue by including sensor technology in their switches and analysis software for your machine (Windows/Unix) but this is far beyond the budget of your average organization. There are also other solutions to the Switch issue using 'taps' etc. For an insight into current directions and an independent assessment of existing products including Snort, check out the NSS Group.[16]

So if you are running with switches then what do you do? Well, I am going to suggest that you find/borrow/buy a small (5or8) port hub, you probably have kept your old ones after you upgraded to switches. If you are on a brand new switched installation that never had any hubs, then I'm afraid that all I can suggest is look in the local computer mart, computer exchange magazines or Internet for a used one.

3.2 Instructions for a Switched Network

TO BE DONE OUTSIDE NORMAL WORKING HOURS

1 connect the UPLINK port of the HUB to any spare port on your SWITCH (not the up-link)
2 disconnect your firewall from your switch and plug it into any spare port on the hub.
3 connect your NID machine to any spare port on the hub.

"Why do it this way?" Well we did outline earlier that the overwhelming majority of attacks came from the Internet and they do. Therefore in this position your NID will not only be in prime place to listen to all the data coming in to your network from your Firewall, but will also hear what is going out from your Network before it gets to the Firewall! You may be surprised to find that you have disabled all appropriate ports on your Firewall, including those used by programs like IRC or Limewire only to find a whole lot of new ports opened up. Firewall circumvention is now a feature of these products. Unless that is, your firewall is configured to deny all by default.
Remember the onion, - Defense in Depth.-

To recap we have from outside to inside:

1 Internet
2 Firewall
3 NID
4 Firewall and NID connected to HUB
5 Hub connected to your Switched LAN

Note: Snort is capable of full IDS packet analysis on a 'maxed-out' 100Mbs LAN so you can be assured that it won't miss anything when sitting on a 10Mbs Hub. Add to that, the fact that unless your organization is very rich, your Internet feed will be no where near 10Mbs.

4. Installation - *What's with all this Unix command business?*

Ok there is no way to get around this issue but if you want to be able to use these security tools you are going to have to learn a bit of Unix. Now you don't have to be some kind of 'geek' that knows every extension in the system folder type, it is quite straight forward, and I will try to give understandable explanations of what we are doing and why we are doing it as we go along.

Note: I don't think the Mac community has quite realized that OSX has brought the Pandora's box of UNIX software to the Mac. Thousands of powerful, highly specialized sophisticated, diverse and refined Free programs. Way beyond anything you will see on a Microsoft Windows box. Did I say Free well yes I most certainly did. [1] Just from a Security perspective you now have the same tools available to you as your Attacker. There is a whole wonderful subterranean world lurking inside the Terminal Application on your Dock.

4.1 Let's get Interactive.

You are going to need to have installed on your Mac both the OSX (Duh!) and also the OSX Developer Package which comes very thoughtfully on an entirely different CD. Mine is Grey and cream and says Mac OSX Developers Tools, it is not installed by default, but is shipped with every Mac so you have no excuse's. Both the OSX and the Developers CD need to be the same version number or it won't work! (For the purposed of this 'how-to I'm using 10.1)

The installation package is called 'Developer.mpkg' and the icon looks like a box, just double click and follow the instructions.

Once you have installed and rebooted you will need to go get the following pieces of Unix software.

snort-1.8.4.tar.gz	http://www.snort.org/dl/
libpcap-current.tar.gz	http://www.tcpdump.org

4.2 Getting 'root' - NetInfo Manager

To do some of the stuff we are about to do you will need 'root' privileges. This is the 'don't mess with me because I'm important!' account on a Unix box. A similar account on a Windows Machine would be the 'Administrator', on a Mac, well the closest thing would be what ever the username and password that you give to administer your Apple Share File Server, in other words, a very omnipotent account. Now your probably thinking 'oh yea that's me because my OSX setup says I can administer this machine'. Well sorry but that's not strictly true, Apple have disabled the 'root' account by default, probably a wise move as the first time user can behead themselves with 'root' privileges. So the first thing we are going to do is enable it!

Find the OSX partition/disk on which you installed OSX, then go: Application Folder -> Utilities Folder-> and open the program NetInfo Manager.

Under the menu bar titled 'Domain' you will see a menu for 'Security' and an arrow to 'Authenticate' this is your only option at the moment. Selecting this will offer a dialog for you to type that all important username and password that OSX told you would enable you to Administrate the machine. Now go back in again and you will see 'Enable root user', select it and you will get a subsequent dialogue box telling you that you will have to "re-authenticate to make additional changes". You need to do that right now, so as to give the root account a password. So for the last time, go back in and use the "change root password" menu.

To check our modifications, lets open a Terminal window "command shell" or just "shell" in Unix speak and type some stuff (your machine name may be different than what you see below). The Terminal application button can be found on the Dock.

```
[localhost:~]me% su -
Password:
[localhost:~] root#
```

So first we type the 'su' command this is for 'set user' just like logging in under a different person, as long as you have the password which you do, Using "su" on its own assumes that you want the root account and just prompts for the password. I could have easily have said "su fred" then it would have asked me for freds password.

Note: "su -" tells Unix that I would like all of the root environment too, this would be any special startup scripts 'root' uses and access to root's applications. The password field is left blank, no bullets.

It works? Then you now have 'root' and are sitting in the root account home folder. How do I know this, well type 'pwd' this stands for print working directory and 'Hey Presto' Unix tells you where you are.

```
[localhost:~] root# pwd
/private/var/root
```

To exit from the root user account type

```
[localhost:~] root# exit
[localhost:~] me%
```

4.3 Creating your installation environment

We are about to create a number of folders or 'directories' to prepare for the installation of the software you previously downloaded, Many thanks to Joe McAlerney of 'www.SiliconDefense.com' for the install details.

I suggest that what you will need is a folder to keep the numerous bits of stuff that these applications need during their day to day operation. I assume its only you who are going to use Snort (as it requires root privileges) so why not create a 'security' folder within your User folder on the machine in question. To do this, get back to your Terminal window and type the following;

[localhost:~] me%mkdir /User/yournamehere/security

Note: You should be back in your own account here and not still in 'root' as this will have a direct effect on the folder permissions.

mkdir make directory the /a/b/c is the 'path' to the folder where you want the new folder to be made. I will from now on start calling folders directories, as this is getting confusing (they are the same thing after all) and the last bit is, yes, the name of the directory we are creating.

Note: You probably already know this but Unix is a case sensitive OS, however Apple have for some strange reason decided to bend the rules somewhat. I can type 'Users' and 'users' and get to the same directory now this would Never happen in any other Unix as they would be seen as two completely different directories. To be safe and compliant with existing practices keep all your files and directories in lowercase.

Next, go back into the 'root' account and create the following directories in the same way. These directories are standard Unix conventions, my default OSX install did not have a '/usr/local/' directory this is the area where most Unix's put your 3rd party Applications like say Snort or Libpcap.

/usr/local/src
/usr/local/include
/usr/local/include/net
/usr/local/lib
/var/log/snort

Now to see if they are there and to get some practice moving around the directories in the command line.

Note: With OSX, all the standard Unix directories, the ones containing the really juicy stuff, are not visible on the Mac desktop (another safety feature?) so if you want so see inside the "/var/log/snort" directory or look at the settings in '/etc/syslog.conf' then the only way is to use the Terminal window and type some commands.

5 Basic navigation commands in Unix

Most Unix commands come with their own form of radio buttons or check boxes if you will. They are called "Flags" or "Switches". A command expects a Flag when it sees this '-' and there is always a space between the command and the '-' but no space after. You will also notice that Unix is not the most verbose OS in the world so when you create a new directory or delete all your work it doesn't comment it just presents a new input line. Anything it doesn't like it gives you the old "command not found" routine.

5.1 Some commands

man	my personal favourite, the on-line Unix manual, need to see what a command does or what Flags you can use with it? this is the boy, you can even type 'man man' and get to the manual page about the manual, thorough or what ?
man cd	gives the manual pages on the cd command
cd	change directory
cd /users/x/security	should take you to the directory you made earlier (note the space after the cd)
cd ..	move up one directory
ls	list the contents of the directory
ls -al	list all the files in the directory including 'a' the invisible ones and 'l' make it a long list with lots of info. An invisible directory/file always starts with a '.' dot.

Note: the asterisk '*' this is what's known as a wild card so if I execute the following command,

ls	sysl*	I should get every entry in the directory that starts with sysl
ls	*og	I should get every entry in the directory ending with og

Recap: all directories have been created in the right places, if not, then the below command will delete them and you can try again.

"rmdir directorytoremovename"

We now need to create a user account called 'bin' as this will take ownership of some of the programs we are installing this does not come as standard on the Max OSX install. To do this we will utilizes the command line interface of the NetInfo Manager application that we used previously (niutil).

On Unix a running application (like a Word processor) would be called a "process" now someone or something owns all processes. For example: If I log in then run a word processor, then I own the program and its running with my authority, so it can only do things that I can do like create or delete files in my directory and no one else's. Now there

are certain processes that need to be owned independently of 'real' Users. So there are 'Users' on the system that don't have any login accounts and you won't see a User folder for them in the NetInfo Manager. Because this 'User' is actually part of the system. The user 'bin' is one such user that we are about to create. Another Unix standard that we shall adhere to.

```
[localhost:~] root#  niutil -create / /users/bin
```

To check it's been created go into the NietInfo Manager and it should be there near the top under Users.

NetInfo Manager is Apples own Unix thing and is part of the Darwin Open-Source Project, it has also been shown to be one of the primary security concerns of OSX [17]

5.2 Extraction - *Are you sure this won't hurt?*

Exiting from the 'root' account and moving to your '/users/yourname/security' directory where you have placed the two downloaded files (if you haven't then best to do it now), we shall proceed with the extraction of the data.

The Unix files you downloaded will have the extension '.tar.gz' or sometimes '.tgz', this indicates to the initiate that the files have been archived by the 'tar' program , standing for 'Tape Archive' (your Unix version of Retrospect Backup) and also compressed using the standard Unix compression package 'gzip' (Unix version of StuffIt). We shall follow the standard practice here and check the archive files first. We are checking which files will be extracted and the Paths that will be used in the extraction process.

Note: Due to UNIX's legendary " You must know what You are doing attitude" it won't ask you if you are sure whether you want to delete your tax return folder that took the whole holiday weekend to do and you haven't done a backup yet... it just deletes it.

So how is this an issue with a backup? Well Absolute Path Names, you may find that an archive wants to create a directory called "applications" now this is ok if for example it starts with say "temp/applications" as this will create a directory called "temp" in your current directory and then a directory called "applications" within that. However if it just starts with "/applications" then this is an Absolute Path name. "/" at the start means an Absolute Path from your root directory. Note "./" at the start is a Relative Path (ie starts in which ever directory you happen to be in).

The root directory is analogous to, but not exactly like, the Icon of your disk/partition on the desktop (HD), in respect that all other folders are contained within it. During extraction, with an Absolute Path specified in the archive file, 'tar' will tell Unix to create a directory called "applications" at the "/" directory, the outcome of this is that your existing applications directory (and everything in it) is deleted Ouch!

Here is the command to first CHECK that the zipped archive has no Absolute Paths

[localhost:/users/me/snort.p] me% zcat snort-1.8.4.tar.gz | tar -tf -

Note: its important that you are not in the root account here as 'tar' will apply the permissions of the 'operating account' and if its 'root' then every file you extract will be 'owned' by the root account. Why does this matter? Well, when you come to edit some of these files you will not be able to use the Mac TextEditor as it will be operating with your own account (the one you logged in with) and you will find you don't have permissions to modify the files!

Zcat is a compression program and will un-compress the snort file. Here we see in action, one of Unix's most powerful features, it's called output redirection, and this is done with the "pipe" "|" command. (Imagine pluming) Here we are taking the OUTPUT data from the compression program as it de-compresses it and 'piping' it into the INPUT of the Tape Archive program. This in turn has the '-t' Flag for table (list of contents) set and the '-f' Flag for file set, the "-" after the 'f' Flag is known as standard output (the screen) so we are now piping the Tape Archive output to the screen.

Note: Using MacClassic OS this would entail
1.opening a file with one program then
2.exporting that file with said program so as it was of a compatible format to be read with a second program, then
3.opening the exported file with the second program and viewing it on the screen. During this process we have created an export file on the machine, i.e. we have un-stuffed the file in order to read it. Unix lets us do this with no additional file creation and much quicker, it's the way it was designed.

Below is the start of the output from the above command showing that the 'tar' program will be creating a directory called "snort-1.8.4" (note there are no Absolute Path names), in the current directory and we can now proceed to extract the archive.

snort-1.8.4/MIBS/SnortCommonMIB.txt
snort-1.8.4/MIBS/SnortIDAlertMIB.txt
snort-1.8.4/contrib/ACID-0.9.6b11.tar.gz
snort-1.8.4/contrib/create_oracle.sql
snort-1.8.4/contrib/Spade-092200.1.tar.gz
snort-1.8.4/contrib/Guardian.tar.gz
snort-1.8.4/contrib/Net-SnortLog-0.1.tar.gz
Snip

Here is the command to extract the zipped file, notice the only difference is the 'x' instead of the "t", x is for Xtract. I know this isn't as easy as 'drag and drop' with StuffIt expander, but its got to be easier than you first thought?

[localhost:/users/me/security] me# zcat snort-1.8.4.tar.gz | tar -xf -
[localhost:/users/me/security] me#

Like I said, absolutely NO comment from Unix just another command line prompt. If you now do a 'ls' at the prompt, you should find the 'snort-1.8.4" directory in your current directory.

Following the exact same procedure for the 'libpcap-current.tar.gz' file. You will have now successfully extracted the required software onto your disk. You have just learned the very basics of how to extract any (well almost any) .tar .gz or .tgz file that you will come across on the net.

5.3 Compiling the code, the open source way

Yes, well, there is a very good reason for this, very briefly its portability.

The idea is this: Because Unix runs on so many different types of machines and there are so many different flavors of Unix, that its much easier for the people who write the program to give you the raw code, (GNU License) and then let your own particular Unix compile it for you. This is because your Unix will know exactly what kind of hardware/peripherals and software you have on your machine and what it's capable of. Much like getting a tailor made suit. If they didn't do this then they would have to compile or make a 'binary' file (application) for every possible eventuality. Like one for a G4 with 0S9.1, one for a G3 with OS9.2, etc very time consuming. Of course the commercial application developers like Microsoft will provide you with the binary (application) and a very clever Installer that does all this figuring out for you. After all that's one of the benefits of paying for their software right?, But they would still have to give you a different Installer for each platform, Intel, Mac , Sun (I think they have a version that runs under Solaris) and they even need a different installer for their own OS, either 9x or 2000 or XP.

We shall start with compiling and installing Libpcap, as it needs to be installed first or Snort won't compile but first a few short words about Libpcap.

5.4 Libpcap

In technical terms, Libpcap is a system-independent interface for user-level packet capture.

Its the program that the 'Sensor" E box, part of Snort asks for the data (packets), from the network card in order to pass it to the analysis part, A box. Since almost every system vendor provides a different interface for packet capture (re hardware/firmware), the libpcap authors created this system-independent API (Application Program Interface) to ease in porting and to alleviate the need for several system-dependent packet capture modules in each application.

So it's basically acts as a one size fits all listening device (promiscuous bug) for any type of machine. There is even libpcap for Windows, Ugh! Programmers like the Snort Programmers, and many others, then use this guy instead of having to write their own

bug for every different type of machine out there.

One more thing, always read the README files, even if you don't understand a word now it will become clearer with time. How?

```
[localhost:users/me/security/libpcap-2001.11.18] root#less README.txt
```

The 'less' command is a screen reader program, you can scroll down and up and do other stuff too, pretty luxurious for Unix, really! To escape from the 'less' command type 'q' at any time.
'man less' for more info.

Now for the configuration and installation, get back to the 'root' account and type the following whilst in the libpcap directory.

```
[localhost:users/me/security/libpcap-2001.11.18] root# ./configure
```

The './' signifies that its the 'configure' file in the current directory you want and not any other 'configure' that might be floating about. You should get something like the following output on your screen if all goes well,

```
loading cache ./config.cache
checking host system type... powerpc-apple-darwin1.4
checking target system type... powerpc-apple-darwin1.4
Snip
creating Makefile
creating config.h
```

That's the first part over. Unix, or to be specific, the 'configure' program, (shell script) that came with this installation has assessed your systems attributes and created a 'Makefile' and a 'Makefile.in' file, suitable to your machine. The 'configure' program will also write this lot to a log file called 'config.log'.

Next run 'make', this will do the actual compiling of your program by including all the system wide attributes of your machine with the source code of the program you downloaded.

```
[localhost:users/me/security/libpcap-2001.11.18] root#make
```

Gives you some thing like this,

```
cc -O -I.  -DHAVE_CONFIG_H -c ./pcap-bpf.c
cc -O -I.  -DHAVE_CONFIG_H -c ./pcap.c
Snip
ar rc libpcap.a pcap-bpf.o pcap.o inet.o gencode.o optimize.o nametoaddr.o etherent.o savefile.o bpf_filter.o
bpf_image.o bpf_dump.o scanner.o grammar.o version.o
ranlib libpcap.a
```

Lastly we run 'make install' as this will now put everything that's been compiled into the

right directories, you know, the ones we created earlier. If you are curious about the numbers below, 755 and 644 etc. that's Unix shorthand for 'user' and 'group' and 'everyone else' access privileges, (i.e. read, write, execute). The same stuff you can see under Finder>File>Show Info>Privileges (from the pull down menu) for more info on the Unix way see 'man chmod' for a thorough explanation.

```
[localhost:users/me/security/libpcap-2001.11.18]root#make install
```

again, here is what Unix has to say on the matter

```
[ -d /usr/local/lib ] || \\
   (mkdir -p /usr/local/lib; chmod 755 /usr/local/lib)
/usr/bin/install -c -m 644 libpcap.a /usr/local/lib/libpcap.a
ranlib /usr/local/lib/libpcap.a
[ -d /usr/local/include ] || \\
   (mkdir -p /usr/local/include; chmod 755 /usr/local/include)
/usr/bin/install -c -m 644 ./pcap.h /usr/local/include/pcap.h
/usr/bin/install -c -m 644 ./pcap-namedb.h \\
   /usr/local/include/pcap-namedb.h
[ -d /usr/local/include/net ] || \\
   (mkdir -p /usr/local/include/net; chmod 755 /usr/local/include/net)
/usr/bin/install -c -m 644 ./bpf/net/bpf.h \\
   /usr/local/include/net/bpf.h
[ -d /usr/local/man/man3 ] || \\
      (mkdir -p /usr/local/man/man3; chmod 755 /usr/local/man/man3)
/usr/bin/install -c -m 644 ./pcap.3 \\
   /usr/local/man/man3/pcap.3
```

Right, that's it. You have successfully installed libpcap.

Now follow exactly the same process with Snort.

1 get to the Snort directory
2 ./configure
3 make
4 make install

6. Running Snort

Snort has three 'modes of operation. 'packet sniffer' mode, 'logging' mode and 'full-on' intrusion detection mode. We will demonstrate all three modes here, I recommend that at some stage, you read the 'SnortUserManual.pdf', its in your 'snort-1.8.4' directory, [18] The format of this section loosely follows the manual. The commands explained below have been specifically selected to highlight certain behaviors of network traffic applicable to a 'how-to' tutorial. In Unix there is always more than one way to do things.

6.1 Packet Sniffer Mode

First lets see if it's working, so enter the following command at the prompt;

```
[localhost:users/me/security/snort-1.8.4] root# snort -i en0 -v
```

You should see something like this (if not then see the section below on 'netstat' and interface numbers).

```
Log directory = /var/log/snort

Initializing Network Interface en0

      --== Initializing Snort ==--
Decoding Ethernet on interface en0

      --== Initialization Complete ==--

-*> Snort! <*-
Version 1.8.4 (Build 99)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
```

Above is the all systems go message, hopefully you are seeing this on you screen. To exit use CONTROL 'C' (that's generally Unix speak for I've had enough of this and want to do something else for a change) you will be given a fresh command prompt.

Below is a sample of the kind of data that will fly bye you as you watch.

```
04/02-19:17:13.471985 192.168.1.100 -> 239.255.255.253
IGMP TTL:1 TOS:0x0 ID:13923 IpLen:24 DgmLen:32
IP Options (1) => Opt 148: 0000 1600
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Now to explain the flags, the '-i' Flag is for Interface. Snort needs to know which one of the network interfaces (cards, real or virtual) to use. On my Mac and on yours (unless? see below) that's en0 for 'Ethernet 0'.

To list all your available interfaces use the network status 'netstat -i' command (the -i Flag is for, yes, interface). It should be the same unless you are NOT running with built-in Ethernet, you may, for instance, have a 1GiG PCI Ethernet card, that would give you the interface number 'en1'. Some of this information can be found in the Mac application, Applications ->Utilities -> 'Network Utility'

```
[localhost:users/me/security/snort-1.8.4] root#netstat -i

Name  Mtu   Network      Address        Ipkts   Ierrs    Opkts Oerrs  Coll
lo0   16384 <Link>                     6126     0     6126     0     0
lo0   16384 127        localhost                6126     0     6126     0     0
en0   1500  <Link>     00.30.65.8d.f9.02      11595     0      534     0     0
en0   1500  192.168.1  192.168.1.100          11595     0      534     0     0
```

'man netstat' for loads more info, this is a very interesting program, as it can display lots of data about who your Mac is talking to and what its listening for, 'netstat -a' will show you metaphorically speaking, the 'doors' you have open which, if any are vulnerable, an

Attacker will be on the lookout for. 'netstat -a' is great for insomniacs, but for the real adventurous amongst you, try 'lsof' thats 'list open files' this program also provides the path to the applications that are opening the 'doors' very useful for tracking down rogue applications or Trojans, 'man lsof' for more info.

Right back to work, under the Name bit you see en0 that's my card, the 'lo0' is for the 'loopback' interface', all machines have this as it provides a mechanism for the network portions of the OS to execute certain functionality, like talking to itself for example.

The -v flag stands for verbose (got to be a Unix joke).This is the simplest way to run Snort. All its doing is reading what's on the network, and, with the help of libpcap throwing the TCP/IP packet headers at the screen (if it was a letter this would be the address on the front only not the contents). Not very radical but we only want to see if it is working. This Mode of Snort is called 'Packet Sniffer Mode'. For more detail on this mode and additional Flags refer to the Snort Manual.

6.2 Packet Logging Mode

Now to expand our test, we are going to log something to file, (everything actually). We will keep the' -v' flag switched on just for the time being so as to monitor that we are in fact getting some input. So the new command looks like this:

```
users/me/security/snort-1.8.4] root#snort -i en0 -l /var/log/snort -v
```

You should get the same startup splash screen and rolling data as before. Now the flag '-l' is for 'log', when you use this, you are going to need to tell it where to log the data to. If you don't then... well try it and see. So we will use the directory we prepared earlier. '/var/log/snort'. Using the plain '-l' flag, Snort will create sub-directories based on the IP addresses of the trace.
The '-v' at the end is the verbose. I hope by now you are getting the hang of Unix command structure syntax, y[n]?

Lets look at the log then so 'cd' to '/var/log/snort' and lets see what we have.

```
[localhost:/var/log/snort] root# ls -l
total 144
drwx------  3 root  wheel    58 Apr  6 10:14 10.224.0.1
drwx------  4 root  wheel    92 Apr  6 10:22 192.168.1.100
drwx------  3 root  wheel    58 Apr  6 10:15 255.255.255.255
-rw-------  1 root  wheel  72358 Apr  6 10:24 ARP
```

What you get will definitely vary from this, and will depend on your network. The above, is an 'ASCII text based' (readable by humans as opposed to 'binary'' readable by the machines) capture of traffic on my network, which consists of just my Mac and a Cable Modem which is connected to the ISP. Internet access from my Mac is currently off. We have a file called ARP (Address Resolution Protocol), and three directories each with an IP address for a name, these are 'source IP addresses i.e. the originators of the traffic. Note: the size of the file, 72k, I only left Snort running for 3 mins (put the kettle on) that's

a lot of chat don't you think? (Hold that thought because we will come back to it). This is known as broadcast traffic, a necessary background noise.

6.3 Background Noise

Very briefly; each Mac will have been given its own IP address by a Human, you may have been involved in doing this yourself within your organization. This address can move around though (i.e. its not fixed to the machine). But all machines DO have a fixed address, and it's called a MAC address(Machine Address Code). This is UNIQUE in the world, a big long number that includes the manufacturers ID and other stuff. It is this address that machines use to chat to each other. NOT the IP address. So an ARP request, see below, is simply one machine shouting "Attention everyone, will the machine who has been given the IP address of 10.224.2.209 by the Humans, tell me what its MAC address is so we can chat"

```
[localhost:/var/log/snort] root# less ARP

04/06-10:16:21.528371 ARP who-has 10.224.2.209 tell 10.224.0.1
04/06-10:16:21.528371 ARP who-has 10.224.2.209 tell 10.224.0.1
```

Here we see my cable modem, 10.224.0.1, asking for the MAC address of 10.224.2.209, ARP broadcasts this request, in other words sends it to an IP address that every machine is configured to listen to by default and this is why you see a 255.255.255.255 directory in my list. This is the default broadcast address, only the machine who has been given the specified IP address will respond. For your information that address will be a machine on the ISP's site. Feel free to 'cd' to the other directories and 'less' the files there.

For a comprehensive introduction to the TCP/IP protocol suite you may wish to look at the following sites [19] There is no way I'm going into that, or I will be here forever.

You may think all this detail on ARP is a bit unnecessary, but this illustrates that there is traffic on our network even when we are not using it for anything. Imagine a production network of 100 Macs, a couple of File Servers, a few Printers , a Mail Server and Internet connection and then (depending on its positioning), guess how much traffic Snort would pickup over 3 minutes!
For more details on the Packet Logging mode, especially the 'tcpdump' binary file format, check out the manual.

So out of this mess of noise, how on earth is Snort able to sift through it all, analyze it, find any suspicious activity then alert you?

6.4 Intrusion Detection Mode

Lets type the following and see what pops up:

```
/snort-1.8.4] root# snort -i en0 -dev -l /var/log/snort -c snort.conf
```

You should get some thing like this:

```
Log directory = /var/log/snort

Initializing Network Interface en0

      --== Initializing Snort ==--
Decoding Ethernet on interface en0
Initializing Preprocessors!
Initializing Plug-ins!
Initializating Output Plugins!
Parsing Rules file snort.conf

+++++++++++++++++++++++++++++++++++++++++++++++++++++
Initializing rule chains...
No arguments to frag2 directive, setting defaults to:
   Fragment timeout: 60 seconds
   Fragment memory cap: 4194304 bytes
Stream4 config:
   Stateful inspection: ACTIVE
   Session statistics: INACTIVE
Snip
886 Snort rules read...
886 Option Chains linked into 99 Chain Headers
0 Dynamic rules
+++++++++++++++++++++++++++++++++++++++++++++++++++++

Rule application order: ->activation->dynamic->alert->pass->log
```

Ok this is the big one so lets look at that command again and break it down into sizable chunks.

1 We have added some more instruction Flags to the straight -v". We also want to see '-de' this asks Snort to log the data, i.e. the contents of the letter, '-d'. Also the 'datalink headers', the MAC addresses, '-e'. This is because Attacks can take any form and we could easily miss one if we didn't initially set out to look at everything. However for high speed collection we would leave out the -vde as it wastes time and use the '-b', Binary file type instead as it collects the entire message by default. To read this file, refer to the manual.

```
/snort-1.8.4] root# snort -i en0 -l /var/log/snort -b -c snort.conf
```

2 The next addition is the use of the '-c', the configuration file, and it is in this, that the Analytical Aspects of Snort's operations are to be found. Do we keep the data? Throw it away Raise an alarm or fight back? This is where the decision-making process of NIDS happens.

6.5 The snort.conf file

We won't be modifying this file, as it is a sample template so we will first need to copy it to create one that we don't mind messing up.

[localhost:ian/snort.p/snort-1.8.4] root# cp -p snort.conf snort.conf.dist

cp	copy
-p	preserve as many attributes as possible eg date and time. permissions, etc.

What this does is to create a new file called snort.config.dist (dist for distribution) and it's this one that we will keep as our original. I know APPLE 'D' at the desktop will do this too but hey, "when in Rome..."

Now to edit it, if we messed up and extracted the files whilst still in the 'root' account then we would have to use a nifty Unix text editor like 'vi' or 'emacs' for this, but as I know you were paying attention, let's use TextEdit instead.

Open TextEdit and go get your snort.conf file, it's in the snort-1.8.4 directory.

Snip
This file contains a sample snort configuration.
You can take the following steps to create your
own custom configuration:
#
1) Set the network variables for your network
2) Configure preprocessors
3) Configure output plugins
4) Customize your rule set
Snip

Well that's pretty straight forward so lets look at step 1, you may have noticed that the start of each line in the config file starts with a '#' this is the 'comment' character, in other words , when Snort is reading this file and sees a '#' it ignores the line and moves on. This is a tradition from programming, the idea is that Humans can make notes for themselves in the body of the raw code which won't compile into the finished program. A very good idea, even better when people use it in practice.

6.6 Setting Network Variables

1. You should have an idea of your organizations IP address block, lets say its the reserved private Class 'C' block (giving 254 possible machines on the net) which would be 192.168.1.0/24. The '/24' is shorthand for the 'subnet mask', in octets that would be 255.255.255.0, which is the way the Mac displays it in System Preferences -> Internet & Network -> Network

so modify the line var HOME_NET any

to read var HOME_NET [192.168.1.0/24] or what ever your address block is. This will enable

Snort to log packets relative to your home network.

Note: As you may have noticed, my configuration is just one machine connected to the Internet, I don't have a fixed IP address and the 'pool' of addresses from which I am assigned are all potential Attackers, so I use the default: 'var HOME_NET any'

For more data on IP subnets this multi media tutorial is all you should require, you will need the Windows Mediaplayer Plug in though! [20]

Best keep the external network to 'any'

Next, I will assume that you DONT have a DMZ on your network and that you DONT keep any WEB servers or Mail servers or DNS servers there. If you do then you will need to modify these entries otherwise you can leave them as is.

If you do have a DMZ then you may add something like the following, your address will be different:

```
var HTTP_SERVERS       200.1.34.0/28
```

Lastly the rule path, you only need to modify this if you think that you will be keeping the rules files in a separate directory than this config file. As we wont then we shall leave it as it is. For your information, the './' is Unix shorthand for the current directory, you have seen this before.

Right lets save that and try it out. OK?

6.7 Configuring Pre-Proccessors

We don't! Think of these little cuties as your PhotoShop plugin's, the have been designed to de-code certain types of network traffic and Attacker tricks. Its just way beyond this 'How-To' to get involved with them and they are all pretty much set by default.

6.8 Configuring Output Pluggins

This is where we can tell Snort what to do with all the data that it has accumulated and what it thinks of it: there are two basic types:

1 Writing: Either to a single massive 'binary' file, using the 'log_tcpdump' plugin or writing to a number of 3rd party database applications, check out 'snort-1.8.4/README.database' on instructions to creating the database structures. Unfortunately there is no "Filemaker" 'how-to' there, (a possible future project for someone?)
There are also plugin's for 'XML', 'unified' and others. Apart from 'log_tcpdump' all are outside the scope of this 'how-to'.

2 Notification: If there is something urgent for us to know about, sending messages via 'syslog' and 'console' or SNMP (Simple Network Management Protocol)

These modules are NOT mutually exclusive, so you could run a database output module and the 'syslog' module simultaneously. To explore this part of the config file, we shall be looking at the syslog plugin, this presents an opportunity to say a little bit about Auditing.

6.9 A bit about Auditing

Auditing in Computer Security terms has nothing to do with bookkeeping,
Auditing is not a replacement for other security measures
Auditing will not prevent an Attack.

It is he process of verifying that a system is not being tampered with, and if it is altered in any way, it should tell us which portions of the system are suspect. Subsequently this should be enough information for us to assess what remedial action is required. A tutorial (with case studies) on Auditing by the legendary Wietse Venema of TCP Wrappers fame, is available here. [21] So its some sort of real-time monitoring system that has built in messaging and logging, i.e. SNORT

6.10 An Overview of Syslog

Unix has a program called 'syslog'. This program runs from startup and its basic function is to monitor, report (to the console) and record (to a log file) any events that the administrator of the system deems of interest, 'man syslog' for detail.

In brief, a log message is 'tagged' with a 'Priority' which in turn is made up of a 'Facility' and a 'Level' the Facility describes the Generic Application Type that is generating the log (we shall shortly see Snort generated syslog messages). The Level is the seriousness of the issue, from the basic "thought you might like to know' (info) to the 'WARNING IM GOING TO BLOW UP" (Emergency) type.

Syslog, not surprisingly, has a 'syslog.conf' file, which resides in the '/etc' directory. It is here that we instruct syslog what and where to record, for example:

```
*.info;mail.none;authpriv.none          /var/log/messages
authpriv.*;remoteauth.crit          /var/log/secure.log
authpriv.*;remoteauth.crit          logserver.mynet
```

The above is an excerpt from a sample syslog.conf file, note the format:

Facility.Level Output

the '*' wild card is used to denote 'all'. The '.' separates the Facility from the Level and the ';' is used to include additional programs in the argument. Note that the Level type displayed actually means those Level's messages and every Level message above it too. None means none. So the above lines tell syslog to take the following action:-

Take 'all' programs 'info' Level and above messages '*.info'; except for mail because we don't want any of its messages, just too much clutter, 'mail.none'; Now because this file is freely available to all users, we don't want any secure messages (ie user login names) so 'authpriv.none'. Then put them all in a log file called messages. '/var/log/messages'
now take 'all' authpriv level messages 'authpriv.*'; and remote login messages of a critical level and above 'remoteauth.crit' and write these to a secure log file. One to which only administrators have accesses permissions.

Note: The last instruction is repeated but instead of a log file it has the address of another machine. This is because we cannot assume that we are smarter than the Attackers. They may have been able to compromise our machine and cover their tracks, by re-writing the log files then using a Trojan version of 'syslog' which does not record certain actions. So as a matter of good security practice, we make sure that the log files are duplicated. Sending them to another 'syslog' application running on another machine does this. Our Security Policy will then have specified how often we compare the logs from the different machines. Any difference and we are under Attack!

One further line from the syslog.conf file, this is the bit that concerns the messaging system and is defined at the start of the file. The method Unix uses to notify humans in 'real-time' is via a special type of 'Terminal' called a 'console'. Reading the file it says:

.err;kern.;auth.notice;authpriv,remoteauth.none;mail.crit /dev/console

Which means. Anything with an error message and above; all kernal messages (that's the core of the OS); authentication messages from notice level and up; no authentication or remote authentication (console is readable by all) and only critical and above mail messages. So, just the important stuff that could have an impact of the stability on the machine. Notice that the 'output direction' part of the command which was to a text file (/var/log) is now to a device, '/dev/console'. /dev is the directory where Unix keeps its, for want of a better word, 'device drivers' (technically Unix thinks of everything in terms of files, even hard disks, but we wont go into that here). Think of '/dev' in terms of 'Apple System Folder -> Extensions Folder'.

You can see from the snort.conf file that by default Snort will write to 'auth.alert' so that syslog will see Snort as an AUTH 'Facility' who's 'level' is ALERT. Then post its messages to the console. Syslog doesn't write any 'auth' log files by default, which is why Snort uses this 'Facility', why log here when Snort writes its own logs!

Now to enable this just remove the # 'comment' character
output alert_syslog: LOG_AUTH LOG_ALERT

like so
output alert_syslog: LOG_AUTH LOG_ALERT

I'm sure you thought that was just too much info, when I could have quite easily just said 'take out the hash' but this is a security paper and you are running an OSX Mac, so you do need to know about syslog and be able to use it.

6.11 Customize your rule set

Last but not Least the Rules. These are the Templates or Signatures that Snort uses when analyzing the your network traffic. You will notice some are commented out by default, as they need specific 'tweaking'. The Snort Web page carries a list of all the latest rules, there are new rules being written every day, as Attackers are coming up with more and more clever ways to attack a system. This is exactly like an Anti-Virus Manufacturer's 'virus definition files'. As you can see, all the rule files are in your snort-1.8.4 directory.

Note: over 80% of the SnortUserManual.pdf is devoted to this subject, so if you really feel the need to write one then read it.

The first thing you should know is that Rule Writing is probably not the easiest part of Snort to come to terms with (hence the massive amount of attention it gets in the manual). But due to its very nature (open source), Snort has thousands of very keen and clever users out there who just love writing complex rules. So my advice would be to use them. However, I can't write a Snort 'how-to' with out at least explaining the structure of rules, Here's an example rule from the 'exploit.rules' file.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"EXPLOIT ssh CRC32 overflow /bin/sh";
flags: A+; content:"/bin/sh"; reference:bugtraq,2347; reference:cve,CVE-2001-0144; classtype:shellcode-
detect; sid:1324; rev:1;)
```

Note that this is all one line, so these rules can get very very long.

This first bit is the Rule Action, and the who what and where.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 22
```

alert	this is the rule action to be taken by snort, 'alert, pass, log, activate, dynamic'
'tcp'	the type of protocol, could be 'icmp' or 'udp' too
$EXTERNAL_NET	this is what ever you have defined as EXTERNAL_NET in the '1 Set Network Variables.
any	this is the 'port' that the attack is coming from.
->	direction operator, <> would be by bi-directional
$HOME_NET	this is what ever you have defined as HOME_NET in the '1 Set Network Variables.
22	the target port number, in this instance the port that the secure shell program listens on.

The rest, the bit in () are the rule options, notice the separator ';'

```
(msg:"EXPLOITsshCRC32overflow/bin/sh";flags:A+;content:"/bin/sh";reference:bugtraq,2347;reference:cv
```

e,CVE-2001-0144; classtype:shellcode-detect; sid:1324; rev:1;)

msg	prints the " " message to the alert and packet logs
flags	tests the tcp packet for certain flags, settings
content	look for the " " content in the packet payload
reference	external attack reference id's
classtype	rule classification id
sid	snort rule id
rev	rule revision number

Well that's quite a mouthful and as you can see very comprehensive. If you do have the desire to write your own rules, and why not? then I refer you to the manual. Well that's about it for the snort.conf file, lets just save all those modifications we made and move on to seeing it all in action.

6.13 Using Snort in NIDS Mode!

Time to put all this new knowledge to work, we shall be running a simulation of the reconnaissance part of an Attack, a portscan, one of the methods that an Attacker will use to look at your network and machines in order to see what 'doors'(ports) are open. A quality portscanner, like 'nMap' [23] is one of the best recon tools available to Attackers, with this program an Attacker will even find out what type of operating system you are using, i.e. Mac OSX, Linux, Windows. So can then craft their Attacks to suit. Scary eh! Well nMap is Unix software but hey, why don't you down load it and try it out, its Free and after all, you shouldn't have any problems compiling or installing it.

We wont be using nMap for this test today, what we will be using is a tool that comes shipped with every Mac OSX. Its called Network Utility and you can find it in 'Applications -> Utilities' folder. I'm sure someone will know why Apple decided to include a portscaner in this program. The consequences of randomly pointing it out over the Internet are patently obvious but I'm sure thousand of new OSX Users are doing just that with no idea of what it is that they are doing (raising portscan alarm bells on Secure Systems all over the place). We shall be using this tool to direct a portscan at your Snort machine and ONLY that machine.

You will first need to get SNORT up and running, get the IP address, then hop on to an other machine to do the portscan, we will then return to the SNORT machine to see what appears in the 'console'.

First the SNORT command (we can use less flags now thanks to the '.conf file)

```
[localhost:users/me/security/snort-1.8.4] root#snort -i en0  -c snort.conf
```

Now get your IP address, you get this from 'netstat -i' or from the info tab in the Network Utility program. Next, move onto the scanning machine and open its Network Utility

program, the last tab at the end is called 'portscan' and that's exactly what we are going to do. Type in the IP address of your machine, make sure the 'only test ports between' is un-checked, then hit 'scan'.

Back at the 'console' window on you SNORT machine you should have something like the following, notice the first two lines, which are NOT generated by Snort but are generated from syslog, notifying that the user 'me' was trying to get root and couldn't do it. What happened here was that I actually typed the password wrong the first time but got it right the second. A lot of failed attempts would indicate some one trying a 'bruteforce' Attack on the root password, there is a program called 'crack' out there for Unix which does just that, so the 'console' is a very handy item to keep on your desktop. The next three lines are Snort generated and show that it has successfully picked up the portscan and has logged the offending machine's IP address.

```
Apr  9 11:03:31 localhost su: BAD SU me to root on /dev/ttyp1
Apr  9 11:03:38 localhost su: me to root on /dev/ttyp1
Apr  9 11:05:14 localhost snort: spp_portscan: PORTSCAN DETECTED from 192.168.1.10
(THRESHOLD 4 connections exceeded in 0 seconds)
Apr  9 11:05:16 localhost snort: [1:615:2] SCAN Proxy attempt [Classification: Attempted Information
Leak] [Priority: 2]: \{TCP\} 192.168.1.10:1173 -> 192.168.1.100:1080
Apr  9 11:05:16 localhost snort: [1:618:1] INFO - Possible Squid Scan [Classification: Attempted
Information Leak] [Priority: 2]: \{TCP\}
```

So there you have it your own NIDS running under OSX. With around 600 Rules of known Attack Signatures it wont be long till you are alerted to some Attack or another from the Net, so we shall now look at the implications of what is known as false positive and false negatives.

6.14 False positives and false negatives

A false positive is were your IDS program notifies you that there is something suspicious going on and it thinks its an Attack of some description where in fact it isn't, and its only legitimate network traffic. These are the bain of every security administrator, you will get false positives and you will need to understand why they are being generated. This is where you will develop your experience with the basic network protocol, TCP/IP and what is, and is not, legitimate traffic on your network. Ultimately you may find yourself writing or modifying Rules for SNORT to accommodate your specific network topology.

A false negative is where your IDS doesn't report anything and you are under attack. A classic false negative scenario would be the 'ILOVEYOU' virus. This spread at an alarming rate before the Anti-Virus companies had a chance to update their virus definition files, (Rules in Snort IDS language). So the lesson here is not any different, make sure you have the latest SNORT rules on your system or you could be getting a false sense of security.

Note: NIDS should be just one of your defense tools. As it has what could be termed 'a fatal flaw' due to the very nature of 'passive' network monitoring. It will 'Fail Open', this means that if your SNORT machine goes down for any reason, your

network will still be open but unmonitored, so an Attacker could carry out an exploit without your knowledge. The reverse of this situation would be a Firewall, which is designed to 'Fail Closed', i.e. if it breaks for whatever reason then no traffic would be allowed either in or out of your network.
Remember - Defence in Depth

6.15 Incident Response Organizations

What do you do when Snort reports suspicious activity and you think you think you have a real live intrusion going down?

Do you:

A: Panic and immediately shut down your Firewall.
Drastic, but certainly effective, but how long do you keep it down?

B: Discover and block the inbound IP attack addresses at the Firewall.
This can work, but attackers generally use dial-up accounts, which in turn use address pooling. So you would have to start blocking large chunks of addresses and this will lead to a Denial of Service within your organization. Possibly the Attackers primary intention in the first place!

C: Visit http://www.incidents.org/ ,[22]
Look up the Snort info (i.e. port number and IP address) to see if this is a known current exploit and if or how it affects your system and what action to take.

To make the most of these services here, you should obtain the 'client software' from the Internet Storm Centre, which will then enable you to up-load your Snort logs, make sure you also sign up to the Public Intrusion Mailing List.

7 Conclusions and Observations

So that's about it, thanks for staying with me, its been one hell of a learning curve all round. If you have enjoyed the ride and still feel adventurous then I recommend that you head over to insecure.org and test drive 'nMap' [23].

On this 'How-To":
This 'how-to' was produced as an exam piece for the 'practical assignment' of the SANS GIAC Essentials course.

On Information Security:
Keep informed, the world of computer crime is, with out a shadow of a doubt, the fastest developing area in Information Technology, subscribe to as many relevant mailing lists you can and do make sure that you investigate all the 'console alerts' not just the Snort ones.

On Unix:
Unix has been around for 20 years. It is the core of the entire Internet and isn't likely to disappear during in our lifetime,(no matter what Mr.Gates would fervently wish for). More importantly, it is now the core of Apples OS strategy. This may be your first foray into the world of the command line but it won't be your last.

On Snort:
There is so much more in this program that I haven't covered, its all in the manual, please read it!

And Finally:
Updates, Updates, Updates. Apple have already released numerous security fixes for OSX make sure that you have them all installed as your vulnerability level with OSX is far greater than OS9.

References:

1 Stallman, Richard. "The GNU Project" (1988).
URL: http://www.gnu.org/gnu/thegnuproject.html (1 April 2002)

2 Preston, Norvell. "Improving the Security of a Default Install of Mac OS X (v10.1)" (March 5, 2002).
URL: http://rr.sans.org/mac/default_install.php .
Source: Orlowski, Andrew. "BSD '3 times as popular as desktop Linux' – Apple." 14 Feb 2002. URL: http://www.theregister.co.uk/content/4/24060.html (1 Mar 2002).

3 Multiple Authors, "Apple Issues" Featuring 6 articles. SANS Reading Room
URL: http://rr.sans.org/mac/mac_list.php (18 Mar 2002)

4 Apple Press Release, "Apple Releases Darwin 1.0 Open Source" (5 April 2000)
URL: http://www.apple.com/pr/library/2000/apr/05darwin.html (Archive Material)

5 CSI, "Computer Crime and Security Survey." (5 April 2002)
URL: http://www.gocsi.com/press/20020407.html (5 April 2002)

6 Northcutt, Stephen. "Intrusion Detection The Big Picture" Part 1 (page 17). PDF flie Part of the SANS GIAC Course material. (v1.0 July 2000). (rev1.4a 6 June 2001)

7 CERT® Incident Note IN-2002-03. "Social Engineering Attacks via IRC and Instant Messaging"
URL: http://www.cert.org/incident_notes/IN-2002-03.html (19 Mar 2002)

8 The Computer Security Policies Group, "Computer Security Policies - Contents and Delivery!"
URL: http://www.computer-security-policies.com/ (2001)
Alternative Source: The SANS Institute "The SANS Security Policy Project"
URL: http://www.sans.org/newlook/resources/policies/policies.htm (2002)

9 MIT, "Welcome to the MIT Distribution Center for PGP (Pretty Good Privacy)"
URL: http://web.mit.edu/network/pgp.html (2002)

10 Zirkle, Laurie. "What is host-based intrusion detection?" Intrusion Detection FAQ. The SANS Institute Resources.
URL: http://www.sans.org/newlook/resources/IDFAQ/host_based.htm (2002)

11 Sobirey', Michael ."A list of 92 IDS Systems". "Michael Sobirey's Intrusion Detection Systems page" (15 June 2000)
URL: http://www-rnks.informatik.tu-cottbus.de/~sobirey/ids.html

12 Shadow Intrusion Detection Project, Information Assurance Office, Warfare Center - Dahlgren Lab, Current softwate version 1.7 (Sept 2001)
URL: http://www.nswc.navy.mil/ISSEC/CID/

13 Barrett, Daniel J., Silverman, Richard E., "SSH: The Secure Shell The Definitive Guide" (1 Feb 2001)
URL: http://www.snailbook.com/

14 Ptacek, Thomas H., Newsham, Timothy N. "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection"
Paragraph: 1.1 The CIDF Model of Intrusion Detection Systems (Jan, 1998)
URL: http://www.snort.org/docs/idspaper/

15 Frederick ,Karen Kent. "Network Intrusion Detection Signatures, Part One" SecurityFocus Online, InFocus Magazine,(19 Dec 2001)
URL: http://online.securityfocus.com/infocus/1524

16 Paragraph beginning: "Network Node IDS (NNIDS)" The NSS Group (2001) URL: http://www.nss.co.uk/ids/introduction.htm

17 Miller, III ,Roland E. Paragraph: 2.4 Account Security: NetInfo. "Mac OS X 10.0 Security Essentials" SANS Reading Room (21 Aug, 2001) URL: http://rr.sans.org/mac/OSX_sec.php

18 Roesch, Martin. "Snort Users Manual" Snort Release: 1.8.4 SnortUserManual.pdf (13 Mar 2002)

19 Hedrick, Charles L. "Introduction to the Internet Protocols" Computer Science Facilities Group RUTGERS The State University of New Jersey (3 July 1987) URL: http://oac3.hsc.uth.tmc.edu/staff/snewton/tcp-tutorial/ (20 Jan 1994)

20 Botsford, Charles C., "A Free Lecture-based Educational Course on IP Addressing and Subnetting" URL: http://www.learntosubnet.com/ (2001)

21 Farmer, Dan. and Venema, Wietse. "Internet Security Auditing " (30 April 1996), URL: http://www.fish.com/security/auditing_course/

21 Web site for the SANS Incident Response Centre. URL: http://www.incidents.org/

23 Download area for nMap, 'THE' Unix portscanner, URL: http://www.insecure.org