# Global Information Assurance Certification Paper

**Intrusion Detection using ACID on Linux**
**GSEC Practical Assignment**
**Version 1.2d**
Rusty Scott
7 September 2001

## Introduction/Background

At a recent meeting with our IT staff, we were discussing the sometimes overwhelming amount of security required in managing a network and we quickly came to the conclusion that the entire group could devote nearly all of their time to network and system security and never have a shortage of projects. One member of our team made the comment that in that case, we would "…have a soccer team full of goalies!" So, the questions of where we draw the line in the sand with respect to security goes on. Appropriate amounts of effort (time) and implementation (hardware/software) are necessary to protect our systems and users and it needs to be maintained within the constraints of our staffing and fiscal resources.

Because our environment has not been one that is conducive to firewalls and the policies that are associated with them, we have developed a set of security practices that includes a number of key features mentioned in the SANS defense in depth model including:

- Maintaining current operating systems, patch levels and service packs
- Unix host-based integrity checking with rdist
- Anti-Virus software on PC's
- Installing tcp wrappers on unix services
- Centralized syslog (by both unix and Windows systems) coupled with swatch
- Installing a hard-to-guess password algorithm
- Periodically running password cracking routines

While this list includes some good perform related to system security, they are either host-based procedures or logging facilities. A network-based layer of security would add an important layer to the existing security model. In a Linux Journal article on the securityfocus.com's web site, Nalneesh Gaur states "Security has now evolved to a point where there is a widespread awareness in the IT community regarding the need for IDS." Until the policies are developed at our site to support the use of a firewall where hosts and services are specifically being blocked or allowed, a network intrusion detection system (NIDS) will provide the ability to analyze network traffic for suspicious activity without denying services to our users. Dave Wreski and Christopher Pallack further explain in an article posted on linuxsecurity.com's web site that "An NIDS captures and inspects all traffic, regardless of whether it's permitted or not. Based on the contents, at either the IP or application level, an alert (can be) generated".

A popular tool used for network intrusion detection on many networks today is snort, which is available from http://snort.sourcefire.com . Wreski and Pallack continue on by saying, "Snort is a "lightweight" NIDS in that it is non-intrusive, easily configured, utilizes familiar methods for rule development, and takes only a few minutes to install. Snort currently includes the ability to detect more than 1100 potential vulnerabilities". The cost of snort ($0) and the

availability of analysis front ends for snort-sniffed networks make it an attractive NIDS solution. One such front end is the Analysis Control for Intrusion Detection (ACID), part of the AirCERT project from Carnegie Mellon. ACID is a PHP-based analysis engine designed to search through and process a database of incidents generated by security-related software such as IDSes and firewalls. Sensor systems are used to collect alert data that is sent to the ACID system for further analysis. For the purpose of this paper, the sensor and ACID system are one in the same.

## The pieces

The operating system of choice for this project was linux. Our IT group has deployed linux systems as Samba servers for providing file and print services to PC laboratories and Netsaint monitor systems and found it to be stable and reliable while performing admirably even on modestly configured Intel hardware. This project was built on a 400Mhz pentium II system with 160Mb RAM.

### Linux:

The folks at Tummy (http://www.tummy.com) produce a linux distribution called Kevin's Redhat Uber Distribution (Krud). Mercifully for systems administrators, more linux installation processes (like Krud's) are being designed with fewer services running by default. System owners can then turn on only the services they wish to run. Tummy's subscription service provides a straight-forward mechanism for staying current with respect to the operating system and security patches. For $65/year, you can receive monthly updates to the OS, security patches and the plethora of additional packages supplied by Tummy.

### Apache:

The Apache web server (version 1.3.14) was installed during the initial operating system and very few changes were necessary to complete a vanilla, although functional installation. It would be worthwhile to spend some time going through your web server configuration file (httpd.conf) to verify that at least the basic settings (server port #, server process user ID and group ID, location of html documents) are appropriate for your environment. The httpd.conf file itself is well documented and more information related to configuring the apache server can be found at http://httpd.apache.org/docs/misc/FAQ.html .

### MySQL & PHP:

First, I downloaded the most recent stable version of both of these packages; MySQL version 4.23.40, from http://www.mysql.com and PHP version 4.0.6 from http://www.php.net . The MySQL web site has an excellent series of short papers that will guide you through a cookbook installation of MySQL and PHP. These papers include the 'configure' script options, compiling instructions, post-installation tasks and testing procedures to verify a successful installation. From the main MySQL page, go to "Articles", then "Building a Database-Driven Web Site Using PHP and MySQL".

*Important notes:* When running 'configure' in the PHP source directory, be sure to use the '--with-gd' option if you are considering producing graphs with ACID. Also, after installing PHP

(using *make install*), be sure to make the appropriate edits to your httpd.conf file so your web sever can find PHP. See http://www.mysql.com/articles/ddws/6.htm for details.

**Snort:**
As stated earlier, snort is one of the most popular network intrusion detection packages available today. Snort compares the attributes of IP packets to sets of rules that are built around signatures of known probes, scans and outright attacks.

There are plenty of options for building snort, but a simple download, tar extraction, *make*, *make install* will suffice. Getting snort to act according to your needs depends much upon the snort.conf file which allows you to set network variables that define your network and the scope of the external network, configure preprocessors, configure output plugins and customize your rule sets.

Database support is necessary to log alerts to something other than an ascii file. The database support needed for this project has been included since snort version 1.6.3. For more detailed and the most recent information related specifically to the database component of snort, see http://www.incident.org/snortdb . In order for snort to work with MySQL, be sure to configure the 'output database' setting in the snort.conf file to include 'mysql' with the appropriate information about the database name and access info. The database structure must then be built so that snort can log its data into the appropriate fields. Specific instructions for this can be found in the README.database file in the snort source code directory.

**Want graphs?**
If you are interested in producing graphical output from your ACID installation (it's optional), you'll need PHPlot and the GD libraries. According to the ACID documentation, you will need GD 1.8.x and PHPlot version 4.4.6 or newer. My Krud distribution contained GD 1.8.3 so I was able to use the Red Hat Package Manger (rpm) to extract it from the disks. It's also available from http://www.boutell.com/gd . Installing PHPlot 4.4.6 is simply a matter of retrieving the tar file from http://www.phplot.com , extracting the scripts and locating them in a in a php-accessible directory.

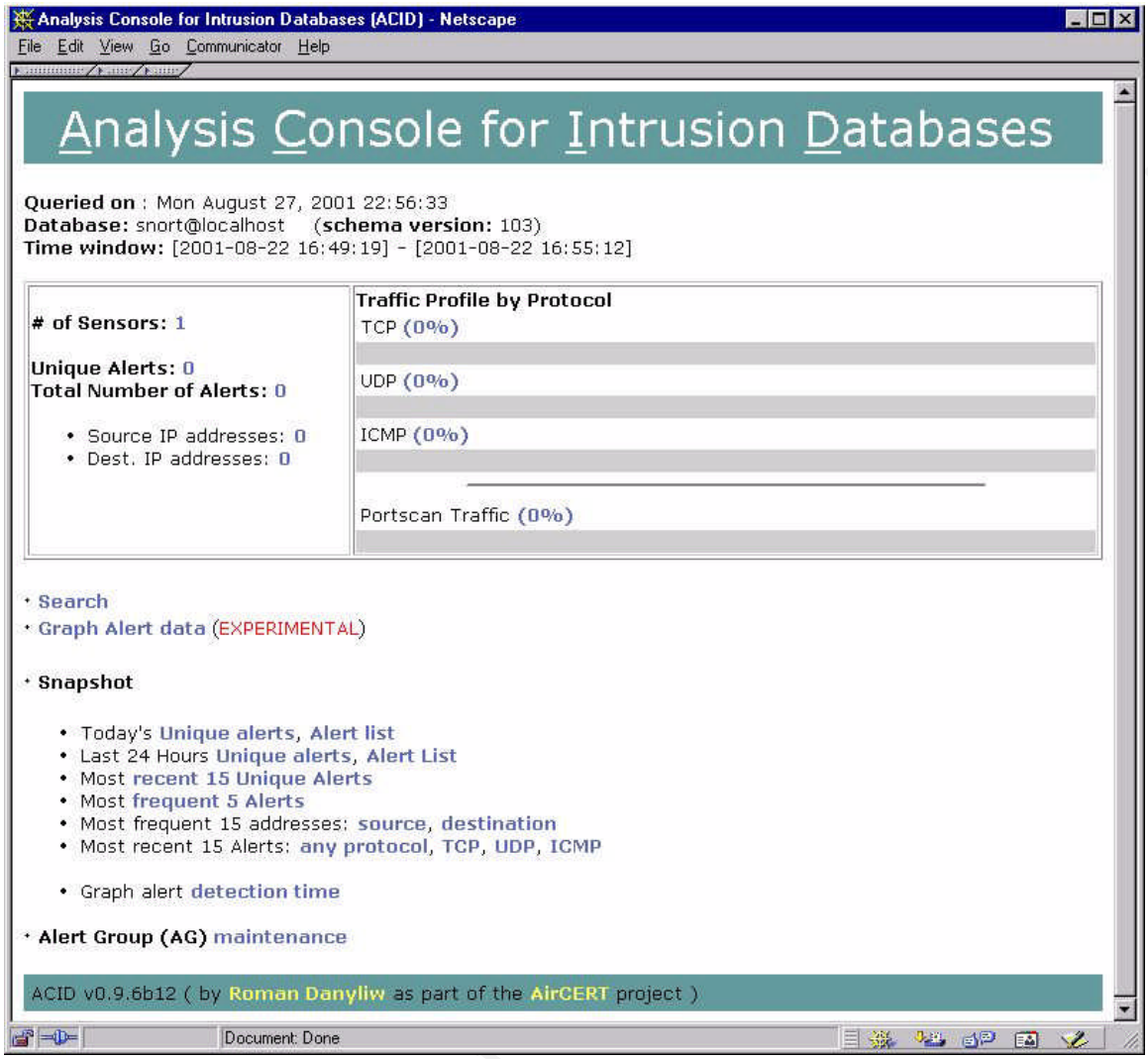
**Adodb:**
One last installation before beginning the work with ACID: since PHP's database access functions are not standardized, a database abstraction library is required. ADODB (Active Data Objects Data Base) is used for this and is available from http://php.weblogs.com/adodb . The installation process consists of extracting the files into a directory that is accessible by your web server. You will define this path in the ACID configuration file.

**ACID:**
The installation and configuration of ACID is a fairly easy process consisting of extracting the gzipped file obtained from http://www.cert.org/kb/acid (version 0.9.6b12) into a web-accessible directory and modifying a few variables in the configuration file 'acid_conf.php'. These

include:

| | |
|---|---|
| *$DBlib_path :* | *full path to the ADODB install* |
| *$DBtype :* | *type of the database used ("mysql")* |
| *$alert_dbname :* | *MySQL database name where the alerts are stored* |
| *$alert_host :* | *host where the database is stored* |
| *$alert_port :* | *port where the database is stored* |
| *$alert_user :* | *username into the database* |
| *$alert_password :* | *password for the username* |
| *$ChartLib_path :* | *full path to the PHPlot install* |

Make the necessary modifications to these variables and point your browser to the directory that contains these files. If no errors pop up, your next step will be to work your way through two screens that you will only see once--the first time you fire up the ACID system. One will report that the table acid_ag is not present. Follow the instructions (and link) to the 'Setup Page' to configure the database. The setup page will present a 'Create ACID AG'button that, if everything is in its proper place, will finalize the process. With everything in its place, the main ACID page will look something like:



**Deployment steps**

The basic philosophy for deployment was:

1. Make sure the system logs correctly to the database.
2. Aim high with respect to alert numbers.
3. Adjust settings (scope of network, snort rules, etc.) to weed out false positives.

I began by using the default rulesets from the snort 1.8 installation and defined the $MY_NET and $EXTERNAL_NET variables to 'any'. Since this system was attached to the network in a switched environment, it only 'saw' packets that were addressed specifically to it along with broadcast packets. A quick nmap scan of the ACID from another computer on the network generated alerts to verify that snort, MySQL, PHP and ACID were all getting along.

Once it was determined that the system was logging correctly, the ACID system's switch port was defined as a monitor port on the switch ('spanning' in Cisco terms) so that it was able to inspect all incoming and outgoing traffic on four class B subnets. This produced a large amount of alerts (over 23,000 in 8 hours), so it was clearly time to adjust the network variables in the snort configuration file (snort.conf) to more specifically define what it was we wanted to study. The following modifications were made to define HOME_NET as our 4 subnets and the EXTERNAL_NET as everything outside of those subnets:

*var HOME_NET [x.x.1.0/24,x.x.2.0/24,x.x.3.0/24,x.x.4.0/24]*
*var EXTERNAL_NET !$HOME_NET*

**Turning down the knob**

At this point, it's necessary to start weeding out what you can safely conclude are false positives or extraneous alerts for your particular network. For example, snort's stream4 preprocessor was built to perform stateful inspection of TCP sessions and TCP stream reassembly. It was generating a LOT of alerts related to the NT/W2K systems on our network that were considered legitimate. In the interest of shutting down the fire hose, I added the command-line argument 'noalerts' to the stream4 preprocessor in the snort.conf file to turn them off. Also, I missed configuring the DNS servers in the snort configuration file which triggered a large amount of false DNS-related scans. Defining *DNS_SERVERS* to include our servers eliminated those particular alerts.

It is also worth mentioning that in a week, this system has generated over 13,000 alerts. Well over half of those are some type of IIS probe. If you're not concerned with these probes, it would be prudent to comment 'include web-iis.rules' out of your snort.conf file. If you are interested in these warnings, expect your database to grow quickly.

**Viewing the data**

The ACID front end provides a lot of flexibility for viewing data collected by, in this case, snort. The queries created by the links under the 'Snapshot' label on the front page are very useful—for example, a single click to a list of unique alerts over the last 24 hours or to the most frequent 5 alerts nicely summarizes the types of probes or attack attempts your network has seen. For example:

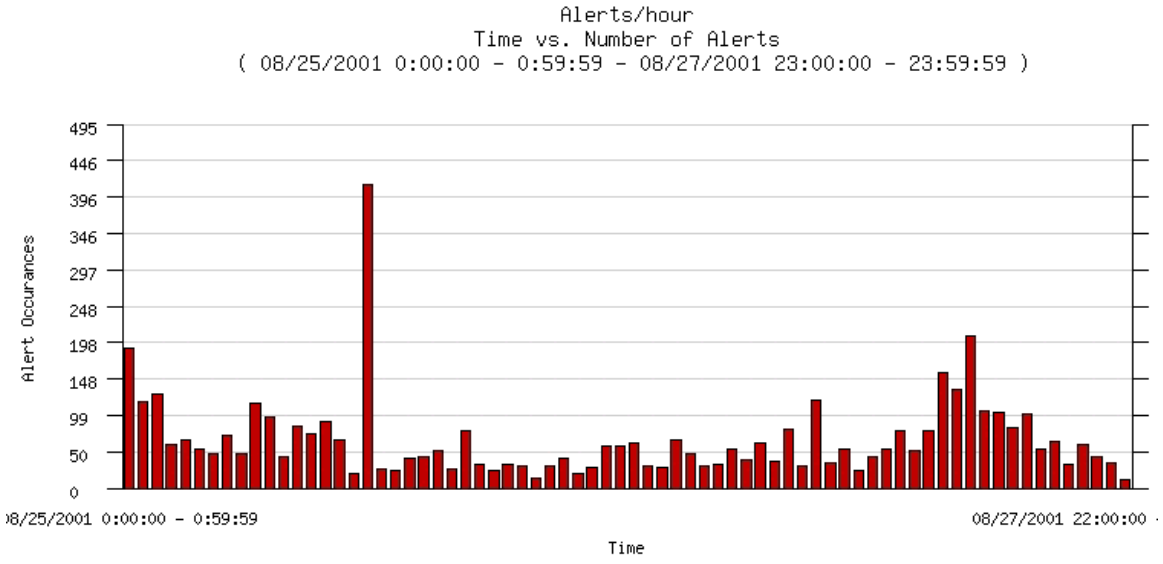
Displaying 5 Most Frequent Alerts

**Queried DB on** : Thu August 30, 2001 07:48:07

| | < Signature > | < Total # > | # Sensors | Src. Addr | Dst. Addr | < First > | < Last > |
|---|---|---|---|---|---|---|---|
| ☐ | ICMP Destination Unreachable (Communication Administratively Prohibited) | 2947 (22%) | 1 | 146 | 35 | 2001-08-22 16:49:19 | 2001-08-29 15:18:03 |
| ☐ | [arachNIDS] WEB-IIS ISAPI .ida attempt | 2915 (22%) | 1 | 1232 | 60 | 2001-08-22 16:51:48 | 2001-08-27 21:34:16 |
| ☐ | WEB-IIS cmd.exe access | 2258 (17%) | 1 | 694 | 61 | 2001-08-22 16:51:48 | 2001-08-29 15:09:39 |
| ☐ | [arachNIDS] [CVE] codered II attempt | 1038 (8%) | 1 | 480 | 59 | 2001-08-27 21:40:44 | 2001-08-29 15:04:16 |
| ☐ | [arachNIDS] MISC Large ICMP Packet | 974 (7%) | 1 | 80 | 127 | 2001-08-22 16:53:43 | 2001-08-29 14:45:26 |

Action { action } Selected ALL on Screen

The 'graph alert detection' button allows you to view the number of the alerts that have been triggered over a specified period of time, shown in hours, days or months. Also, there is a facility labeled as EXPERIMENTAL (Graph Alert data) that allows you to view the number of alerts aggregated by time (hour, day, month), by IP address and by port numbers. Using this facility, a graph of the number of alerts per hour over a three-day period looks like:




## Tuning, an on-going process

This is clearly as starting point in our attempt to monitor network-based intrusion attempts. Tuning will be a continuous process of, not only weeding out false positives, but also adding new rules as new variants of intrusions (like the Code Red worm and its mutations) are developed.

The snort site is a valuable source of information related to the development, installation, configuration, rules and add-on packages (like ACID). In particular, be sure to check out the FAQ and discussions at this site.

Citations and referenced links:


Gaur, Nalneesh. "Snort: Planning IDS for Your Enterprise". Linux Journal. 2001. URL: http://www.securityfocus.com, (6 September 2001)

Wreski, Dave & Pallack, Christopher. "Network Intrusion Detection Using Snort". LINUXSECURITY.COM Features, 19 June 2000, URL: http://www.linuxsecurity.com/feature_stories/feature_story-49.html (6 September 2001)

Snort – The Open Source Network Intrusion Detection System. URL: http://snort.sourcefire.com (6 September 2001)

Acid Console for Intrusion Databases – AirCERT project. URL: http://www.cert.org/kb/acid (6 September 2001)

Hertel, Chris, "Samba: An Introduction". The Samba Web Pages. 10 June 1999. URL: http://us1.samba.org/samba/docs/SambaIntro.html. (6 September 2001)

Galstad, Ethan. "Netsaint Documentation, Version 0.0.7". 13 August 2001. URL: http://www.netsaint.org/docs/0_0_7. (6 September 2001)

tummy.com, ltd. URL: http://www.tummy.com (6 September 2001)

Apache Server Frequently Asked Questions – The Apache HTTPD Sever Project. URL: http://httpd.apache.org/docs/misc/FAQ.html (6 September 2001)

The MySQL web site. URL: http://www.mysql.com (6 September 2001)

The PHP web site. URL: http://www.php.net (6 September 2001)

Yank, Kevin. "Building a Database-Driven Web Site Using PHP and MySQL". 1995-2001. URL: http://www.mysql.com/articles/ddws. (30 August 2001)

INCIDENT.ORG: Database support for SNORT. URL: http://www.incident.org/snortdb (6 September 2001)

Boutell.com's graphics library. URL: http://www.boutell.com/gd (6 September 2001)

The PHPLOT web site. URL: http://www.phplot.com (6 September 2001)

ADODB for PHP – PHP Everywhere. URL: http://php.weblogs.com/adodb (6 September 2001)