

Constructing a Wireless Intrusion Detection System with Snort, MySQL, Apache, PHP, ACID & BASE on a Linux Platform

William M. Fitzgerald, *Telecommunications Software & Systems Group (TSSG), Waterford Institute Of Technology Ireland*

Abstract—Modern technology has lead to a rapid proliferation of wireless networks and mobile computing applications which has dramatically altered our understanding of network security. The traditional way of protecting networks with firewalls and encryption software is no longer sufficient and effective [1].

This paper discusses the practical side of constructing a wireless intrusion detection system with a snort traffic analyzer, MySQL database back-end and a PHP based ACID/BASE front-end on a Linux Platform. It is intended to be a one-stop-shop installation guide to help one through the labyrinth of installation and configuration processes.

Index Terms—Wireless LAN analysis, Intrusion Detection, Security

I. INTRODUCTION

THE growth of 802.11 networks has lead to the development of numerous wireless local area network (WLAN) discovery applications. These applications are designed to identify WLAN activity and network characteristics, providing enough information for an unauthorized user to gain access to the target network [2].

This paper is an in depth instruction manual to building a Wireless Intrusion Detection System (WIDS) to monitor and report intrusion attempts on ones network.

II. DOCUMENT LAYOUT

Each of the different software packages will be discussed in turn. The software sections in the paper follow an ordered flow of installation from the ground right up to the completed WIDS. The paper describes a generic installation guide of the software whereby one must substitute the correct version number for the version of software they are installing. For example: `mysql-version.tar.gz` would be replaced by `mysql-standard-4.1.9-pc-linux-gnu-i686.tar.gz`. The installation of Snort is the only non-generic installation procedure as Snort-

2.1.1 is the only supported version of wireless IDS.

Note: the installation and description process described in this paper is based on the software versions discussed in the next section. However the process should be migrateable for all software versions. Readers are advised to read the installation instructions of the respective software versions.

III. SOFTWARE REQUIRED

Listed below are the software packages required to build the WIDS system described in this paper. The list is laid out as follows, the generic software name followed by the software version that will be used in construction the system. *Note: Both ACID and PHPlot are not required if one decides to use the newer BASE software.*

- **MySQL:** `mysql-standard-4.1.9-pc-linux-gnu-i686`[3]
- **Automake:** `automake-1.6.1` [4]
- **Snort:** `snort-2.1.1` [5]
- **Snort-Wireless patches:** `Snort-2.1.1-wireless` [6]
- **Zlib:** `zlib-1.2.1` [7]
- **JPEG:** `jpeg-6b` [8]
- **Libpng:** `libpng-1.2.8` [9]
- **GD:** `gd-2.0.33` [10]
- **Apache:** `httpd-2.0.52` [11]
- **PHP:** `php-4.3.10` [12]
- **ADODB:** `adodb460` [13]
- **ACID:** `acid-0.9.6b23` [14]
- **PHPLOT:** `phplot-5.0rc2` [15]
- **JPGGRAPH:** `jpggraph-1.17` [16]
- **BASE:** `base-1.0.1` [17]
- **Linux:** Debian Linux [18]
- **Atheros wireless card** [19]

IV. MYSQL DATABASE

The MySQL software delivers a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software [3].

A. Installation of MySQL

1. Copy or save the `mysql-version.tar.gz` to `/usr/local`

Manuscript written 11 February, 2005. "Constructing a Wireless Intrusion Detection System with Snort, MySQL, Apache, PHP, Acid & BASE on a Linux Platform"

William M. Fitzgerald is with the Telecommunications Software & Systems Group, at Waterford Institute of Technology, Ireland (phone: 00353 51 302937; fax: 00353 51 302901; e-mail: wfitzgerald@tssg.org).

- directory
- 2. groupadd mysql
- 3. useradd -g mysql mysql
- 4. cd /usr/local
- 5. tar zxvf mysql-version.tar.gz
- 6. ln -s mysql-version mysql
- 7. cd mysql
- 8. scripts/mysql_install_db --user=mysql
- 9. chown -R root .
- 10. chown -R mysql data
- 11. chgrp -R mysql .
- 12. bin/mysqld_safe --user=mysql &

B. Explanation of Install Process

Steps 2 & 3 adds a login user and group for *mysqld* daemon. Steps 5 and 6 show how the *tar* command creates a directory named *mysql-standard-4.1.9-pc-linux-gnu-i686*. The *ln* command makes a symbolic link to that directory. This lets you refer more easily to the installation directory as */usr/local/mysql*.

Step 8 runs a script in the scripts directory. This directory contains the *mysql_install_db* script used to initialize the *mysql* database containing the grant tables that store the server access permissions.

Steps 9 to 11 first changes the owner attribute of the files to the *root* user. The second changes the owner attribute of the *data* directory to the *mysql* user. The third changes the group attribute to the *mysql* group.

Step 12 shows how to run the *mysqld* daemon.

C. MySQL Development Libraries

One may also need to install the *mysql* development libraries so that *Snort* and other programs can interface with. The usual place for library installation is in the */usr/local* directory. With Debian Linux you can use the *apt-get* command or with Redhat the *rpm* command. The paper focuses on the Debian installation process.

- 1. cd /usr/local
- 2. mkdir mysql
- 3. cd mysql
- 4. apt-get install mysql-dev
- 5. apt-get install libmysqlclient-dev
- 6. apt-get install libmysqlclient12-dev

D. MySQL Configuration

You will need to create a *my.cnf* file in */etc* if one does not exist.

- 1. cd /usr/local/mysql-version /support-files
- 2. cp my-medium.cnf /etc/my.cnf

You may need to use *my-large.cnf* or *my-small.cnf* instead of *my-medium.cnf* depending on your systems capabilities.

IMPORTANT: You will need to make a sym link of the *mysql.sock* file in the */tmp* directory. *Snort* will try to look for this file in the */var/run/mysqld* directory. By default the *mysql.sock* file is stored in */tmp* as set out by the *my.cnf* file.

- 1. cd /var/run
- 2. mkdir mysqld
- 3. cd mysqld
- 4. ln -s /tmp/mysql.sock mysqld.sock

V. AUTOMAKE

Automake is a tool for automatically generating *Makefile.in* files compliant with the GNU Coding Standards [4].

This software may have to be installed as was the case with the author. When the *make* command for *Snort* is executed you may receive an error like the following:

aclocal-1.6: not found WARNING: `aclocal-1.6' is needed, and you do not seem to have it handy on your system. You might have modified some files without having the proper tools for further handling them. Check the `README' file, it often tells you about the needed pre-requirements for installing this package. You may also peek at any GNU archive site, in case some other package would contain this missing `aclocal-1.6' program.

Download the version of automake that the *Snort make* command complains about.

A. Installation of Automake

- 1. Copy or save the *automake-version.tar.gz* to */usr/src/imports* or whatever
- 2. tar zxvf *automake-version.tar.gz*
- 3. cd *automake-version*
- 4. ./configure
- 5. make
- 6. make install

After automake is installed you will need to do the following:

- 1. cd *snort-version*
- 2. make distclean
- 3. aclocal
- 4. autoheader
- 5. autoconf
- 6. automake

VI. SNORT WIRELESS IDS

The *Snort-Wireless* patch adds several new features for 802.11 IDS functionality to the standard *Snort* distribution. These features allow one to specify custom rules for detecting specific 802.11 frames, rogue access points, AdHoc networks, and Netstumbler like behavior in the vicinity of the *Snort-Wireless* sensor [5,6].

The only available wireless capable *snort* is an older version of the *snort* distribution (*snort-2.1.1*) and has two available patches to turn the vanilla *snort* into a wireless IDS.

A. Installation of Wireless Snort

- 1. Copy or save the *snort-2.1.1.tar.gz* to */usr/local*
- 2. Copy or save the *snort-2.1.1-wireless.patch.gz* to */usr/local*
- 3. Copy or save the *snort-2.1.1-wireless-db.patch* to

- ```

/usr/local or whatever
4. tar zxvf snort-2.1.1.tar.gz
5. patch -p0 < snort-2.1.1-wireless.patch.gz
6. patch -p0 < snort-2.1.1-wireless-db.patch.gz
7. cd snort-2.1.1
8. ./configure --prefix=/opt/snort --enable-wireless --
 with-mysql
9. make
10. make install
11. cp etc/* /opt/snort/etc
12. cp rules/* /opt/snort/rules
13. Configure the snort.conf file to your specifications
14. Run: /opt/snort/bin/snort -c /opt/snort/etc/snort.conf
 -i ath0

```

### B. Explanation of Install Process

Step 5 and 6 apply the wireless patch code and configuration files to the vanilla snort-2.1.1 source. Step 8 configures the makefile and final destination of the binary to be in the `/opt/snort` directory. It also configures the snort compile to include both wireless sniffing and mysql logging support.

Steps 9 and 10 compile and install the final binaries. It is recommended to copy the `etc` and `rules` directory to the snort binary directory also. So copy those directories to the `opt/snort` directory.

Step 13 involves editing the `snort.conf` file to monitor and configure rules to your specification. See section ‘Configuring Snort’ for further details.

*Note: If you did not want a wireless IDS then skip steps 2 to 5 and change step 8 to `./configure --prefix=/opt/snort --with-mysql`*

### C. Configuring Snort

Change directory into `/opt/snort/etc` and edit or tailor the `snort.conf` file to your needs. The snort configuration file will only need to concern it self with layer 2 traffic analysis for the most part. Vanilla Snort by default is focused on layer 3 traffic. Below is a simple, shortened but valid wireless `snort.conf` file with wireless capability:

```

Set the network variables:
var HOME_NET any

Set up the external network addresses as well.
var EXTERNAL_NET any

Configure your wireless AP lists.
var ACCESS_POINTS 00:0a:95:f6:3c:c0
var CHANNELS 11

Configure masked MAC addresses.
var MACSPOOF_MASKED_ADDR none

RogueAP
preprocessor rogue_ap: $ACCESS_POINTS, $CHANNELS,

```

```

scan_flag 1, scan_timeout 1800, expire_timeout 3600

```

```

AntiStumbler
preprocessor antistumbler: probe_reqs 90, probe_period 30,
expire_timeout 3600

```

```

DeathFlood
preprocessor deauth_flood: deauth_threshold 20,
expire_timeout 60, target_limit 100, prune_period 30

```

```

AuthFlood
preprocessor auth_flood: auth_threshold 100, expire_timeout
60, target_limit 10, prune_period 30

```

```

MacSpoof
preprocessor macspoofer: $MACSPOOF_MASKED_ADDR,
tolerate_gap 5, threshold 10, expire_timeout 120,
spoofed_addr_limit 100, prune_period 30

```

```

Include classification & priority settings
include classification.config

```

```

Include reference systems
include reference.config

```

Consult the relevant documentation to extend and develop a personalised `snort.conf` file.

### D. Configuring Interoperation of Snort and MySQL

Tasks need to be carried out here are:

- modify the `snort.conf` file to allow for MySQL logging
- Create the snort databases and its corresponding tables

#### (1) Editing the Snort.conf:

Go to the “Configure output plugins” section and add lines similar to below:

```

Configure output plugins
output database: log, mysql, user=snu dbname=snort
host=localhost

```

#### (2) Creating the Snort database:

Change directory to point to `/usr/local/mysql/bin`. From there run the following commands:

1. `./mysql -h localhost -u root`
2. `create database snort;`
3. `create database snort_archive;`
4. `grant CREATE, INSERT, DELETE, UPDATE, SELECT on snort.* to snu@localhost;`
5. `grant CREATE, INSERT, DELETE, UPDATE, SELECT on snort_archive.* to snu@localhost;`
6. `quit`
7. `./mysql -h localhost -u snu snort < /usr/src/imports/snort-2.1.1/contrib/create_mysql`
8. `./mysql -h localhost -u snu snort_archive <`

- ```

/usr/src/imports/snort-2.1.1/contrib/create_mysql
9. ./mysql -h localhost -u snu snort <
/usr/src/imports/snort-2.1.1/contrib/snortdb-extra
10. ./mysql -h localhost -u snu snort_archive <
/usr/src/imports/snort-2.1.1/contrib/snortdb-extra

```

E. Explanation of Configuration

Create the database parameters to be used to access and log data to the database in the *snort.conf* file. In the above example a database called *snort* is used to log data and a user *snu* can access that data on the machine running MySQL which so happens to be the localhost.

Login as *root* user to MySQL on the localhost and create two databases, *snort* and *snort_archive*. Database *snort* will be used primarily be the Snort to log data while *snort-archive* will be used as an archive repository by the ACID or BASE front-end.

Grant the login user (*snu*) permissions to for those databases (steps 4 & 5). Steps 7-10 will populate the databases created with the relevant tables required by the IDS system.

VII. ZLIB

Zlib is a free, general-purpose, legally unencumbered data-compression library for use on virtually any computer hardware and operating system. The zlib data format is itself portable across platforms [7].

A. Installation of ZLib

1. Copy or save the *zlib-version.tar.gz* to */usr/local*
2. `cd /usr/local`
3. `tar zxvf zlib-version.tar.gz`
4. `mv zlib-version zlib`
5. `./configure`
6. `make`
7. `make install`

VIII. JPEG

The jpeg library installed for this system is the Independent JPEG Group's free JPEG software for JPEG compression and decompression [8].

A. Installation of JPEG

1. Copy or save the *jpegsrc.version.tar.gz* to */usr/local*
2. `cd /usr/local`
3. `tar zxvf jpegsrc.version.tar.gz`
4. `cd jpeg-6b`
5. `./configure --enable-shared`
6. `make`
7. `make test`
8. `make -n install`

IX. LIBPNG

The *libpng* library is a Portable Network Graphics (PNG) raster image files reference library [9]. It will be used

within the GD library describe later.

The *zlib* libraries must be installed prior to the installation of *libpng* libraries.

A. Installation of Libpng

1. Copy or save the *libpng-version.tar.gz* to */usr/local*
2. `cd /usr/local`
3. `tar zxvf libpng-version.tar.gz`
4. `mv libpng-version libpng`
5. `cd libpng`
6. `./configure`
7. `cp scripts/makefile.linux makefile`
8. `make`
9. `make install`

X. GD

GD is a graphics library for fast image creation. When building from source, GD requires that the following libraries also are installed, in order to produce the related image formats [10]:

- *libpng* if you want PNG
- *zlib* if you want PNG
- *jpeg-6b* or later if you want JPEG

A. Installation of GD

1. Copy or save the *gd-version.tar.gz* to */usr/local*
2. `cd /usr/local`
3. `tar zxvf gd-version.tar.gz`
4. `cd gd-version`
5. `./configure --with-png --with-jpeg`
6. `make`
7. `make install`

XI. APACHE WEB SERVER

The Apache web server is an open-source HTTP server for modern operating systems including UNIX and Windows NT [11].

A. Installation of Apache

1. Copy or save the *httpd-version.tar.gz* to */usr/local*
2. `tar zxvf httpd-version.tar.gz`
3. `cd httpd-version`
4. `./configure --prefix= /usr/local/apache2 --enable-so`
5. `make`
6. `make install`
7. `/usr/local/apache2/bin/apachectl start`

B. Explanation of Install Process

Step 4 configures the *makefile* to your specifications.

Step 5 will compile the Apache build for you and Step 6 will deploy it to its final resting place. Step 7 runs the server daemon

XII. PHP

PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be

embedded into HTML [12].

Prerequisite is to have apache already installed and optionally MySQL and GD.

A. Installation of PHP

1. Copy or save the php-version.tar.gz to /usr/local directory
2. tar zxvf php-version.tar.gz
3. cd php-version
4. ./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-mysql --with-gd --with-zlib --with-jpeg-dir
5. make
6. make install
7. restart apache server to take the new PHP installation on board

B. Explanation of Install Process

Step 5 configures PHP distribution with MySQL support, GD library support including some dependencies of GD and web server support

C. Configuring PHP

1. cp php.ini-dist /usr/local/lib/php.ini
2. cp php.ini-dist /etc/php.ini
3. cp php.ini-dist /etc/php4/apache2/php.ini
4. Edit each of the php.ini files to reflect the following change: *display_errors = Off*
5. Edit the *http.conf* file and make the following changes:
Search for *LoadModule php4_module modules/libphp4.so* and add these 2 lines below it:
AddType application/x-httpd-php .php .phtml
AddType application/x-httpd-php-source .phps
6. Restart the web server daemon in order to take affect

XIII. ADODB

ADODB is a suite of database libraries that allow you to connect to multiple databases in a portable manner [13].

A. Installation of ADODB

1. Copy or save adodb-version.tar.gz to /usr/local/apache2/htdocs
2. cd /usr/local/apache2/htdocs
3. tar zxvf adodbversion.tgz

XIV. ACID

The Analysis Console for Intrusion Databases (ACID) is a PHP-based analysis engine to search and process a database of security events generated by various IDSes, firewalls, and network monitoring tools collaboration, or archiving of alerts to transfer them between alert databases [14]. BASE is now the new Snort PHP front-end so one is advised to install it instead of ACID.

It is assumed that MySQL, Snort, Apache, PHP, PHPLOT, ADODB and GD are already installed.

A. Installation of ACID

1. Copy or save acid-version.tar.gz to /usr/local/apache2/htdocs
2. cd /usr/local/apache2/htdocs
3. tar zxvf acid-version.tar.gz

B. Configuring ACID

1. cd /usr/local/apache2/htdocs/acid
2. Modify the *acid_conf.php*:
Search for the following in the *acid_conf.php* and edit to your needs. Below is an example:

```
$DBLib_path = "/adodb";
```

```
/* The type of underlying alert database */
$DBtype = "mysql";
```

```
/* Alert DB connection parameters */
```

```
$alert_dbname = "snort";
$alert_host = "localhost";
$alert_port = "";
$alert_user = "snu";
$alert_password = "";
```

```
/* Archive DB connection parameters */
```

```
$archive_dbname = "snort_archive";
$archive_host = "localhost";
$archive_port = "";
$archive_user = "snu";
$archive_password = "";
```

```
/* Path to the graphing library */
```

```
$ChartLib_path = "/phplot";
```

XV. PHPLOT GRAPH PLOTTER

PHPPlot is graph library for dynamic scientific, business, and stock-market charts. Written in PHP and supports, PHP3, PHP4, TTF and GD versions 1.2 - latest version [15].

PHPPlot will only need to be installed if one will be using ACID support as opposed to BASE.

A. Installation of PHPLOT

1. Copy or save phplot-version.tar.gz to /usr/local/apache2/htdocs/acid
2. cd /usr/local/apache2/htdocs/acid
3. tar zxvf phplotversion.tgz

XVI. JpGRAPH

JpGraph is a fully OO (Object-Oriented) Graph creating class library for PHP. The library can be used to create numerous types of graphs on-line [16].

A. Installation of JgGraph

1. Copy or save jpgraph-version.tar.gz to /usr/local/apache2/htdocs
2. cd /usr/local/apache2/htdocs

- tar zxvf jppgraph-version.tar.gz

XVII. BASE

BASE is the Basic Analysis and Security Engine. It is based on the code from the Analysis Console for Intrusion Databases (ACID) project. This application provides a web front-end to query and analyze the alerts coming from a SNORT IDS system [17].

It is assumed that MySQL, Snort, Apache, PHP, JpGraph, ADODB and GD are already installed.

A. Installation of BASE

- Copy or save base-version.tar.gz to /usr/local/apache2/htdocs
- cd /usr/local/apache2/htdocs
- tar zxvf base-version.tar.gz
- cd base
- cp base_conf.php.dist base_conf.php

B. Configuring BASE

Edit the *base_conf.php* file and add similar lines as shown below in the example. It is similar to the ACID configuration.

```
$BASE_urlpath = "http://host_IP_Address/base";
```

```
/* Path to the DB abstraction library */
$DBlib_path = "/usr/local/apache2/htdocs/adodb";
```

```
/* The type of underlying alert database */
$DBtype = "mysql";
```

```
/* Alert DB connection parameters */
$alert_dbname = "snort";
$alert_host = "localhost";
$alert_port = "";
$alert_user = "snu";
$alert_password = "";
```

```
/* Archive DB connection parameters */
$archive_dbname = "snort_archive";
$archive_host = "localhost";
$archive_port = "";
$archive_user = "snu";
$archive_password = "";
```

C. Editing BASE Source Code

As of writing this paper neither BASE nor ACID is equipped to handle wireless layer 2 reporting graphics. In order to graph Netstumbler, RogueAP, MACSpooft traffic and so forth the BASE source code needed to be added to. Additions were made to *base_common.php* and *base_stat_common.php*.

The *base_common.php* needs to modify the *PrintProtocolProfileGraphs* function similar to the following to incorporate the wireless layer 2 intrusion alerts:

```
function PrintProtocolProfileGraphs ($db)
```

```
{
    $netstumblerscan_cnt = netstumblerPktCnt($db);
    $rogueAPscan_cnt= rogueAPPktCnt($db);
    $macSpooftscan_cnt=macSpooftPktCnt($db);
    $layer2_cnt = $netstumblerscan_cnt + $rogueAPscan_cnt
    + macSpooftscan_cnt;
    if ( $netstumblerscan_cnt > 0 )
    {
        $netstumblerscan_percent =
        round($netstumblerscan_cnt/$layer2_cnt*100);
        if ( $netstumblerscan_percent == 0 )
            $netstumblerscan_percent_show = "&lt; 1";
        else
            $netstumblerscan_percent_show =
            $netstumblerscan_percent;
    }
    else
    {
        $netstumblerscan_percent = 0;
        $netstumblerscan_percent_show = "0";
    }

    if ( $netstumblerscan_percent > 0 ) $color = "#FF0000";
    else $color="#CCCCCC";
    $rem_percent=100-$netstumblerscan_percent;
    echo '<TABLE WIDTH="100%" BORDER=0>
    <TR><TD>NetStumbler attacks
    ( '.$netstumblerscan_percent_show.'%)</A>
    </TD><TD></TD></TR></TABLE>
    <TABLE class="summarygraph" WIDTH="100%"
    BORDER=1 CELLSPACING=0 CELLPADDING=0>
    <TR><TD ALIGN=CENTER BGCOLOR=".'.$color.'"
    WIDTH=".'.$netstumblerscan_percent.'%"&nbsp;&nbsp;</TD>>;
    if ( $netstumblerscan_percent > 0 ) echo '<TD
    BGCOLOR="#CCCCCC"
    WIDTH=".'.$rem_percent.'%"&nbsp;&nbsp;</TD>>;
    echo '</TR></TABLE>';
}
```

The *base_stat_common.php* file needs to define the functions called in the *PrintProtocolProfileGraphs* function in *base_common.php*: For Example:

```
function netstumblerPktCnt($db)
{
    $result = $db->baseExecute("SELECT count(*) FROM
    acid_event WHERE sig_name LIKE 'Detected Netstumbler
    traffic from%' ");
    $myrow = $result->baseFetchRow();
    $num = $myrow[0];
    $result->baseFreeRows();
    return $num;
}
```

XVIII. PUTTING THE IDS IN MOTION

Having installed all the relevant software as described in this paper or the relevant software versioning to reflect your system, it is now time to execute the IDS system. Example:

1. Configure wireless card to operate in promiscuous mode. See Wireless card setup for instructions.
2. `/usr/local/mysql/bin/mysqld_safe --user=mysql &`
3. `/usr/local/apache2/bin/apachectl start`
4. `/opt/snort/bin/snort -c /opt/snort/etc/snort.conf -i ath0`
5. `http://domain_name/base/base_main.php` OR
6. `http://domain_name/acid/acid_main.php`

A. Wireless Card Setup:

One will need to configure the wireless card to promiscuous mode if you have not done so already. This paper makes use of an Atheros card:

1. `ifconfig ath0 up`
2. `iwpriv ath0 mode 2`
3. `iwconfig ath0 mode monitor`

XIX. CONCLUSION

This paper showed how to construct and interlink the components required to build a fully operational open source wireless IDS system. The wireless IDS described in this paper has both real-time database logging and real-time analysis and GUI front-end.

ACKNOWLEDGMENT

The author thanks **Telecommunications Software & Systems Group, Waterford Institute of Technology, Ireland** for the use of the equipment in order to put in practice IDS security knowledge.

REFERENCES

- [1] Yongguang Zhang et al, "Intrusion Detection Techniques for Mobile Wireless Networks", *Mobile Networks and Applications* (2003) 1-16
- [2] James Wright, "layer 2 Analysis of WLAN Discovery Applications for Intrusion Detection", (Nov 2002), <http://home.jwu.edu/jwright/papers/l2-wlan-ids.pdf>
- [3] MySQL: <http://www.mysql.com/>
- [4] <http://www.gnu.org/software/automake/>
- [5] Snort: <http://www.snort.org>
- [6] Snort Wireless: <http://snort-wireless.org/>
- [7] ZLIB: <http://www.gzip.org/zlib/>
- [8] JPEG: <http://www.ijg.org/>
- [9] LibPNG: <http://www.libpng.org/pub/png/libpng.html>
- [10] GD Library: <http://www.boutell.com/gd/>
- [11] Apache Web Server: <http://www.apache.org/>
- [12] PHP: <http://www.php.net/>
- [13] ADODB: <http://adodb.sourceforge.net/>
- [14] ACID: <http://acidlab.sourceforge.net/>
- [15] PHPLOTT: <http://www.phplot.com/>

- [16] JPGRAPH: <http://www.aditus.nu/jpgraph/>
- [17] BASE: <http://secureideas.sourceforge.net/>
- [18] Linux Debian: <http://www.debian.org/>
- [19] Atheros: <http://www.atheros.com/>



William M. Fitzgerald (MSc, BSc) obtained a Master of Science Degree in Computer Science from National University of Ireland Maynooth (N.U.I.M) in Maynooth in 2002 and an Honors Bachelor of Science degree (majoring in Computer Science & Mathematics) also from N.U.I.M in 2000.

He is currently employed as an applied researcher for the Telecommunications Software & Systems Group (TSSG) at the Waterford Institute of Technology in Waterford, Ireland. William is focused on the security arena within European projects such as SecurIST, Daidalos and the PASR initiative. His current research interests are security (wired & wireless) Malware (Virology, RAT's, Worms, Phyogenetic's). Prior to his current employment he was employed as an applied security researcher with Ericsson's Systems Expertise Group, Dublin, Ireland. There he researched security of Ad-Hoc networks, reputation based metrics and novel game theoretic approaches to network node cooperation.

Some of William's publications: (1) *An Approach for Network Forwarding Systems Quality*, Information Technology and Telecommunications (IT& T), Athlone, Ireland, pp 103 -111, ISSN 1649 - 1246, 2001, (2) *Performance Analysis of Host Based Routing*, Masters of Science, N.U.I. Maynooth Library, 2002, (3) *Ericsson OSS Security Architecture: Current State and Challenges Ahead*, Ericsson R&D Ireland, 2003 (Internal), (4) *Reputation and Cooperation in Ad Hoc Networks*, Ericsson R&D Ireland, Ericsson R&D Ireland, 2003 (Internal).