



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Network Monitoring and Threat Detection In-Depth (Security 503)"
at <http://www.giac.org/registration/gcia>

Detecting and Preventing Rogue Devices on the Network

Detecting and Preventing Rogue Devices on the Network

GCIA Gold Certification

Author: Ibrahim Halil Saruhan, ibrahimsaruhan@gmail.com

Adviser: John Bambenek

Accepted: August 8, 2007

Table of Contents

1. Abstract 5

2. Introduction 5

3. Architecture 9

 AP Architecture 9

 Client Architecture 10

 Server Architecture 10

 Zone Architecture 10

 Location 10

 Assignment 11

4. Set-Up 11

5. Monitoring 13

 CPE Site Survey 14

 CAU Site Survey 16

6. Detection 19

 Man in the Middle Attack 20

Detecting and Preventing Rogue Devices on the Network

Evil Twin Attack.....	27
MAC Address List Match.....	28
Client MAC Address Match.....	28
AP MAC Address Match.....	29
MAC Address Spoofing.....	30
RF Jamming.....	31
DOS Attacks.....	33
Duration Attack.....	34
Deauthentication.....	35
801.1x EAP Authentication.....	38
7. Prevention.....	39
8. Conclusion.....	44
9. Future Work.....	45
10. References.....	47
11. Appendix.....	50
Source Code.....	50

Detecting and Preventing Rogue Devices on the Network

Client MAC Address List Check	50
Client MAC Address Spoofing Check.....	50
AP MAC Address List Check	51
Main Monitor.....	52
Log Traffic.....	53
RF Jamming.....	54
Client Site Survey	54
AP Site Survey	57
Man in the Middle Attack.....	59
Evil Twin.....	60

1. Abstract

Rogue device detection is an important aspect in wireless security. Without proper detection analyzing could be difficult and error-prone. There are known techniques for rogue detection, nevertheless this paper will try to solve the problem from a different perspective. The solution in this paper can be applied to all wireless networks.

The main approach of this paper is to show how to use site survey to detect rogue devices in a wireless network. Site survey, if used correctly is extremely beneficial for detecting rogue devices. Rogue device detection can be considered the initial phase of wireless intrusion detection, in case it is not feasible to install sensors to cover all the wireless network area.

2. Introduction

Rogue access point detection is an important aspect of wireless IDS. (Potter, 2007) Rogue access points can pose significant threats. The first step in dealing with rogue devices is to find out that they exist. A radio device somewhere must note the existence of an unauthorized device. (Gast, 2005)

In this paper rogue device detection is implemented by using different techniques like site survey, MAC address list checking, noise checking and eventually wireless traffic analysis. I created some strategies by using IDS methodologies to detect rogue devices interloping on the network. The experiments are done on a broad wireless network covering a few square miles. The coverage area could have been bigger in theory, but the sample was enough

Detecting and Preventing Rogue Devices on the Network

to do analysis.

Rogue access points come in two varieties. Internal rogue access points are those that (for instance) an employee brings in and plugs into a corporate network. The access point is outside the control of IT personnel and serves as a gateway for attackers to enter the enterprise. The other type of rogue access point is more difficult to control. The external rogue access point is one that is controlled by an attacker and designed to spoof legitimate clients into connecting to it rather than the correct access point. Usually this is accomplished by setting the rogue access point SSID to the same SSID as the friendly access point and then boosting the signal of the rogue access point. This will cause client associations to come to the rogue access point. The attacker may then attempt to steal user's credentials via spoofed web pages and portals designed to trick users into giving up passwords, credit card numbers and other personal information. These types of rogue access points are generally easy to detect but difficult to turn off as the attacker then needs to be physically located. (Potter, 2007)

Rogue can be defined in many other ways. Some say anything other than legitimate clients and AP's are rogue devices. Another approach is that anything on the wireless networks that isn't authenticated is rogue. For this paper, rogue is a device, AP, or client which is trying to connect attack or interfere to the wireless network. Briefly in this paper, rogue is accepted as the malicious entity.

I did most of the testing on dedicated AP's which are not connected to the wired networks. The purpose of an AP in this paper is basically a device which provides service to its clients to connect to the internet. An AP can connect to the

Detecting and Preventing Rogue Devices on the Network

internet by using different methods like wired LAN by using Ethernet, dedicated T1 lines or some other means of connection, such as 802.11a and WiMAX.

AP's can be installed on towers or site hosts. In my testing set-up, AP's are installed on site hosts. Every client has a CPE (Customer Premises Equipment) installed at their office or home. CPE's connect client computers to the AP's. There could be instances like someone spoofing MAC address of CPE's or AP's. On the other hand someone can create noise intentionally like applying broadcast de-authentication attack to one of the AP's or unintentionally.

Moreover someone can create DDOS attack to the AP. At the same time they can start broadcasting with the same SSID. In this case they can ensure AP is not functioning anymore and at the same time spoof the MAC Address of the AP and show clients a fake login page, which is called Evil Twin attack. The point in the Evil Twin attack is the rogue device spoof's the MAC address of the legitimate AP and creates an evil twin. It also has to create DOS attack on the legitimate AP and broadcast with higher power. There might also be cases where someone can start broadcasting with the same SSID but on a different channel and only for listening purposes, which is called Monkey-in-the-middle attack or Man-in-the-middle attack. Most of these cases are tested in this paper. There is some information for each of these cases and also detection strategies.

On the other hand, there could be cases where an AP can broadcast with higher power and same SSID. MAC Address spoofing is also one of the cases that need to be checked. AP's that I was testing are working 24X7. Some enterprises turn off the access points during off hours to prevent war-driving efforts, but this

is not the case for this paper.

A rogue AP can be both hardware and software. There are soft AP's like HOSTAP or FakeAP; nonetheless I decided to use a wireless USB adapter. I bought a brand new USB wireless finder and adapter Zyxel 225-AG. It also functions as an AP. This functionality works only in XP.

Once an AP is discovered, the next step is to identify whether it is a rogue AP or not. One way to do this is to use pre-configured authorized list of APs. Any newly detected AP that falls outside the authorized list would be tagged rogue. Some of the different ways in which IT managers can populate the authorized list are:

- Authorized MAC
- Authorized SSID
- Authorized Vendor
- Authorized Media Type
- Authorized Channel (Anand, 2004)

In this paper MAC address and SSID are checked. Vendor type, media type and channel are not used to detect rogue AP's.

Detection components may be implemented through the use of a scanning feature that searches periodically for unauthorized devices or dedicated scanning devices. (Gast, 2005)

Ibrahim Halil Saruhan

The main approach in this paper is using client devices to do periodical scanning instead of using dedicated scanning devices. Real-time detection of a rogue device on the wireless network depends on continuous monitoring. Because clients are up almost all the time, doing site survey on clients and collecting data in a central server was adequate.

3. Architecture

AP Architecture

There are different kinds of equipment that can be used as an AP. To do the testing, I used specifically central access units (CAU) which supports 802.11b protocol. Each CAU is composed of three independent single board computers. One of them is the main single board computer that does most of the functionality. The other two are secondary single board computers which act only as a wireless-to-wired bridge. Each single board computer functions as an AP.

Wireless radio cards on all the single board computers are identical. They support 802.11b protocol use Prism 2.5 chipset and have 23 dBm RF output power. They also support wireless extensions. Each CAU connect to the internet through T1, multiple T1, or other solutions. On each single board computer, hostap is running to get the air traffic and forward it to the main single board computer. 802.1x is used for authentication.

Perl scripts, which send http requests to CPE and parse the http responses, run in AP's.

Client Architecture

Customer premises equipment (CPE) is a device which includes a wireless card and an antenna. It is connected to the client's computer. Each CPE has a Prism 2.5 chipset wireless card and also runs a small web server to respond http requests to do site survey. The CPE's entire purpose is to connect the client computer to the AP.

Server Architecture

A debian based Linux server is used to do logging. Apache server is installed in the server. The main project is developed in PHP by using a 2-tier application model for simplicity. It stays in the Linux server. SSH2 is installed in this server and used to send data to the AP's securely. There are also cron jobs running on the server.

Zone Architecture

Every zone constitutes a CAU (3 AP's) and multiple CPE's connecting to it. In a zip area there could be multiple zones. For example the first zone in 84095 zip area would be called 8409501 and the second zone would be called 8409502.

Location

Wireless access points can lose signals because of walls, doors, floors, insulation and other building materials. The signals may also enter into another user's airspace and connect with their wireless local area network. This is referred to as accidental associations and can occur in densely populated areas where

several people or businesses use wireless technology. (Lane, 2005)

An AP's placement and signal strength have to be calibrated or blocked to make sure the transmitting coverage is just enough to cover the correct area. The RSSI (Received Signal Strength Indicator) on a wireless card is a good way of measuring wireless coverage inside and outside of a WLAN perimeter. (Hutchison, 2004)

The signal strength needed to make a connection is much higher than that needed to just listen into the network traffic. So by its nature it's a lot easier to just listen than it is to make a legitimate connection. (Hutchison, 2004)

Based on all these, the location of AP's and their connecting clients are selected accordingly.

Assignment

Assignment of clients to the AP's is done based on physical location and signal strength.

4. Set-Up

This project has multiple entities which communicate to each other. Clients communicate with AP's and AP's communicate with the central server. Clients transfer data to AP via http response after the http request from AP and AP transfers the data to the central server via XML. Central server parses the XML data and logs it in a MYSQL database table for further analysis. There are cron

Detecting and Preventing Rogue Devices on the Network

jobs running, which check the database periodically and as soon as there is an indication of intrusion, central server communicates with the AP and orders traffic capture and logs the intrusion. The capture will be analyzed later by the WLAN administrator.

Keeping all this in mind, I did testing on 6 zones (18 AP's) with 298 clients. Each zone has around 30-70 clients. Clients were connecting randomly. Some AP's might have ACL's for authentication, but in my testing AP's don't keep ACL lists. Clients use username and password for authentication. Each CPE sends its username and password to the AP after the request from AP. EAP is implemented where supplicant is CPE, authenticator is AP and authentication server is Radius server. This is not a part of the setup and for some testing EAP is disabled. To be able solve this problem a small tool from Aegis can be installed to an XP Windows box. Aegis is a tool which can help a client to generate EAP packets. It works fine, but as soon as I installed Zyxel USB adapter, there were 3 wireless cards working in Windows which led to numerous problems. Eventually I decided to take EAP out; where I have to use my laptop as a rogue AP whereas in the other cases EAP was in place.

Each AP has a built-in memory card on it. For traffic capture, tcpdump needs to be installed in each AP. AP's, which I used for testing, don't allow me to use their wireless cards as both client and AP. Prism2.5 wireless cards have this problem. A wireless card can be set either in a monitor mode or master mode. It can't work both in monitor and master mode. Atheros chipset overcomes this problem by supplying Virtual AP functionality.

Detecting and Preventing Rogue Devices on the Network

Basically, to be sure that the set-up works fine, we have to check each entity in the system one-by-one. Central server should have the PHP interpreter installed and also PHP code for the logic, database connectivity, and XML parsing and sending notifications. SSH2 is needed for secure communication between AP's and the central server. Cron jobs should be running and have to make sure that everything is running in their environment by using "env". Also command line interface "php.ini" should be modified according to the SSH2 package. AP's should have site survey Perl script running and CPE's should also have their web server running.

5. Monitoring

To be able to monitor the scenarios I developed a simple small GUI. Its code is in the appendix.

Wireless Rogue Device Detection	
CAU Mac Address Control	CAU Mac Address Control
CPE Mac Address Control	CPE Mac Address List Control
CPE Mac Spoofing Control	Rogue Client Mac Control
Evil Twin	Evil Twin
Evil Twin based on Log Date	EvilTwin Log Date
RF Jamming	RF Jamming
Log The Traffic	----- v 1 v Log The Traffic
CPE Site Survey	----- v CPE Site Survey
CAU Site Survey	----- v CAU Site Survey
Insert Evil	----- v 1 v Insert Evil

The best monitoring environment is one which has sensors in place to detect unauthorized activity; and services that can notify authorities via an alarm, analyze data, and provide reports. (Lane, 2005)

I used site survey both from client view and AP view, Arp cache table and CPE log table to do monitoring and collecting data. I leveraged iwlist while doing site survey from AP and I set up a system by using PHP+SSH2+Perl and developed a few programs to do site survey on the client. Clients can't run iwlist or any other wireless commands. I developed a few cron jobs which run periodically and in case of intrusion, AP's are triggered to start logging traffic. In this section, AP and client site survey methods are presented along with some analysis based on the data.

CPE Site Survey

I leveraged a Perl script, which basically creates an http request and sends it to the CPE. CPE runs a small web server on it. It performs a site survey requested by the AP and responds with an http response packet. The same Perl script parses the response packet, gets the data out of it, creates an XML file with the site survey data, and sends it to the original requester which is central server.

Here is a screen shot done on AP while the program traverses all the online clients one by one and runs site survey on them. This data is collected and kept in the database.

Detecting and Preventing Rogue Devices on the Network

```
The XML file is : /home/cpestatus/cpe8409503_30193.xml

CPE INFORMATION
SITE SURVEYS
SITESSID BBC_8409503_3 MACADDRSITE 00:02:6F:07:7D:91 SITECHANNEL 1 SITESTRENGTH -55
SITESSID hpsetup MACADDRSITE 62:77:F5:27:A6:80 SITECHANNEL 10 SITESTRENGTH -93
SITESSID NETGEAR MACADDRSITE 00:18:4D:18:76:82 SITECHANNEL 3 SITESTRENGTH -87
SITESSID TMH321 MACADDRSITE 00:0F:B5:E2:1B:14 SITECHANNEL 11 SITESTRENGTH -79

CPE Status run successfully!!!

The XML file is : /home/cpestatus/cpe8409503_30218.xml

CPE INFORMATION
SITE SURVEYS
SITESSID BBC_8409503_3 MACADDRSITE 00:02:6F:07:7D:91 SITECHANNEL 1 SITESTRENGTH -54
SITESSID oasis_ccj2 MACADDRSITE 00:0B:6B:34:34:3E SITECHANNEL 10 SITESTRENGTH -72
SITESSID Handrahan MACADDRSITE 00:18:4D:00:29:16 SITECHANNEL 3 SITESTRENGTH -56
SITESSID C4Tech-2 MACADDRSITE 00:11:50:53:88:14 SITECHANNEL 11 SITESTRENGTH -81
SITESSID BBC_8409503_2 MACADDRSITE 00:02:6F:07:CF:19 SITECHANNEL 11 SITESTRENGTH -81
SITESSID WIZBANG MACADDRSITE 00:14:6C:DB:64:1E SITECHANNEL 11 SITESTRENGTH -88
SITESSID NETGEAR MACADDRSITE 00:0F:B5:28:72:74 SITECHANNEL 11 SITESTRENGTH -92

CPE Status run successfully!!!
```

To be able to make it work continuously I created cron jobs like this:

```
4 ,23,41 * * * * cpesitesurvey env -i PATH=/usr/bin/ php5 /var/www/secure.
GiacProject.net/Wids/CPESitesurvey8406502.php > /dev/null
6 ,25,44 * * * * cpesitesurvey env -i PATH=/usr/bin/ php5 /var/www/secure.
GiacProject.net/Wids/CPESitesurvey8402001.php > /dev/null
8 ,27,46 * * * * cpesitesurvey env -i PATH=/usr/bin/ php5
/var/www/secure.GiacProject.net/Wids/CPESitesurvey8409503.php > /dev/null
```

Here is the table structure for the site survey data. For 298 clients running site survey 3 times in an hour, the number of records in the table was more than a million after one week. Each site survey process runs from 12-18 seconds. In 20 minutes it can run up to 80 clients. In the testing system, none of the AP's has that many connected clients at any time.

```
CREATE TABLE `cpesitesurveys` (
```

Detecting and Preventing Rogue Devices on the Network

```
`id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
`zoneid` int(7) unsigned zerofill NOT NULL default '0000000',  
`customerid` INT(10) NOT NULL,  
`site_ssid` varchar(50) default NULL,  
`mac` VARCHAR(50) NOT NULL,  
`channel` INT(10) NOT NULL,  
`LogDate` datetime NOT NULL default '0000-00-00 00:00:00',  
`signal` INT(10) NOT NULL  
) TYPE = MYISAM;
```

Here is an excerpt from the CPE site survey table:

```
mysql> select * from cpesitesurveys limit 3;  
+-----+-----+-----+-----+-----+-----+-----+-----+  
| id | zoneid | customerid | site_ssid | mac | channel | LogDate | signal |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
| 650077 | 8412803 | 12908 | belkin54g | 00:11:50:9D:FF:60 | 2 | 2007-07-13 18:25:33 | -76 |  
| 650078 | 8412803 | 12908 | 001601142DEA | 00:16:01:14:2D:EB | 1 | 2007-07-13 18:25:33 | -89 |  
| 650079 | 8412803 | 12908 | ACTIONTEC | 00:0F:B3:23:CA:BD | 9 | 2007-07-13 18:25:33 | -72 |  
+-----+-----+-----+-----+-----+-----+-----+-----+
```

On the other hand, I can't run iwlist on clients but in AP's iwlist works fine.

CAU Site Survey

CPE site survey is a few programs running together to collect data from clients. CAU site survey on the other hand is simple. Basically, I used iwlist on the AP's and recorded the responses to the database.

Here is a screen shot done on AP. This data is collected and kept in the database.

Detecting and Preventing Rogue Devices on the Network

Connected to the Zone: 8408401

Channel	SNR	BcnInt	Capab	Rate	Mac	ATIM	SupRates	SSID
9	26	100	0x11	20	00:06:25:60:73:59	0	<82><84><0b><16>	ahn
2	19	100	0x431	10	00:13:46:f0:4e:bc	0	<96>	ROCCO
6	33	100	0x401	10	00:16:b6:c8:c8:1e	0	<82><84><8b><96><24><30><48><6c>	linksys
6	25	100	0x401	10	00:14:bf:78:0a:8e	0	<82><84><8b><96><24><30><48><6c>	linksys
6	4	100	0x411	10	00:16:b6:30:b9:dd	0	<82><84><8b><96><24><30><48><6c>	My House
6	21	100	0x401	10	00:16:b6:c8:c8:1e	0	<82><84><8b><96><24><30><48><6c>	linksys
6	9	100	0x01	110	00:02:6f:07:cf:18	0	<96>	BBC_8409503_1
6	13	100	0x401	10	00:13:10:0f:e6:2d	0	<82><84><8b><96><24><30><48><6c>	linksys
6	14	100	0x431	20	00:18:3f:f7:d7:01	0	<82><84><8b><96><0c><12><18><24>	ZWIRE170
10	19	100	0x21	10	00:0b:6b:34:34:3e	0	<82><84><0b><16>	oasis_ccj2

A beacon is a small broadcast data packet that reports the characteristics of the wireless network, with information such as supported data rate (max data rate), capabilities (encryption on or off), Access Point MAC address, SSID (wireless network name), etc. (Wong, 2004)

When you check the response from iwlist, you can see the capability information and some more data. For example, capability 0X431 in the second line can be analyzed and it will show whether the AP supports WEP or not and also any other features. All the capability bits can be checked by parsing this 0X431 value.

Here is the table structure for the site survey data collected at the AP:

```
CREATE TABLE `causitesurveys` (  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `channel` INT(10) NOT NULL,  
  `snr` INT(10) NOT NULL,  
  `capab` varchar(10) default NULL,  
  `rate` INT(10) NOT NULL,  
  `mac` VARCHAR(20) NOT NULL,  
  `suprates` VARCHAR(80) NOT NULL,  
  `zoneid` int(7) unsigned zerofill NOT NULL default '0000000',  
  `ssid` varchar(25) default NULL,  
  `LogDate` datetime NOT NULL default '0000-00-00 00:00:00'  
) TYPE = MYISAM;
```

Here is an excerpt from the CAU site survey table.

Detecting and Preventing Rogue Devices on the Network

```
mysql> select * from causersurveys limit 3;
```

id	channel	snr	capab	rate	mac	suprates	zoneid	ssid	LogDate
1	2	18	0x431	10	00:13:46:f0:4e:bc	<96>	8408401	ROCCO	2007-07-23 00:01:22
2	6	18	0x11	10	00:13:10:fa:2d:6e	<82><84><8b><96>	8408401	Casa	2007-07-23 00:01:22
3	6	22	0x401	10	00:14:bf:78:0a:8e	<82><84><8b><96><24><30><48><6c>	8408401	linksys	2007-07-23 00:01:22

```
mysql> select channel, snr, zoneid, mac, ssid from causersurveys where ssid like 'BBC_8%' or ssid like 'TBZ%' group by mac;
```

channel	snr	zoneid	mac	ssid
6	21	8409503	00:02:6f:04:2e:e7	BBC_8407002_2
11	67	8408401	00:02:6f:07:be:39	BBC_8408401_2
6	61	8408401	00:02:6f:07:c1:6f	BBC_8408401_1
11	54	8401509	00:02:6f:07:c3:7e	BBC_8401509_1
1	63	8404303	00:02:6f:07:cc:be	BBC_8404303_1
11	67	8404303	00:02:6f:07:cc:d3	BBC_8404303_2
6	75	8401509	00:02:6f:07:cf:13	BBC_8401509_2
6	57	8409503	00:02:6f:07:cf:18	BBC_8409503_1
11	61	8409503	00:02:6f:07:cf:19	BBC_8409503_2
11	8	8408401	00:02:6f:07:d0:60	BBC_8408804_1
6	13	8409503	00:15:6d:63:02:f9	TBZ_8402002_2
6	13	8409503	00:15:6d:63:11:6e	BBC_8402005_1

Above, I tried to see whether someone is trying to spoof MAC address of one of the AP's. The results were negative, as you can't see multiple SSID's on the list.

According to the results our test AP's have SNR from 87 to 53. After 53 tests shows other AP's around. For this sample set we can say that all the 15 AP's tested are not affected by noise nor is there any MAC Spoofing.

From both the CPE and CAU site survey tables we can also observe that most of the users leave default names "Linksys, NETGEAR and belkin54g" on the wireless routers and most of these AP's also don't support WEP.

Detecting and Preventing Rogue Devices on the Network

```
mysql> select count(*) as TOTAL, ssid from causersurveys group by ssid order by TOTAL DESC limit 10;
+-----+-----+
| TOTAL | ssid |
+-----+-----+
| 204   | linksys |
| 56    | NETGEAR |
| 36    | ACTIONTEC |
| 34    | digis-000 |
| 31    | belkin54g |
| 30    | hpsetup |
| 29    | My House |
| 29    | wireless |
| 28    | BBC_8404303_1 |
+-----+-----+
```

6. Detection

This section starts by explaining 2 attacks namely Man-in-the-middle and Evil Twin attacks. After that I will give examples of MAC address list, spoofing, RF jamming, de-authentication and authentication detection.

My intention to do site survey in the first place was to be able to detect existence of an Evil Twin attack. I wasn't able to detect one actually. I wrote a script where I can do it through adding data to the database, however I wasn't satisfied. Therefore I bought a hardware USB adapter which can work as AP, went to the field and tried to create a rogue by myself and see whether the system would catch me. The experiment went smooth. My novice USB adapter turned to be a good one and was detected by many clients. The program on the cron job notified the AP my existence and the AP automatically started capturing wireless traffic and logged it automatically to the central server.

Man-in-the-middle and Evil Twin attacks are pretty much similar attacks. Both of them use a rogue AP and intend to create denial of service attack to the legitimate AP. Evil Twin attack creates an evil twin by spoofing the MAC address of the legitimate AP and also tries to forward the client to fakes login pages and

trying to gather login information, whereas Man-in-the-middle attack tries to hide its existence to work at least 5 channels away where the legitimate AP is functioning and tries to listen the traffic.

I will start with what the others say about the attacks, showing the experiment and its results and end up with some analysis on what happened by checking the wireless traffic.

The strategy is simple. A cron job checks the CPE site survey table. In case it detects a rogue AP from this data, it triggers the AP to log the traffic. While capturing the traffic AP moves from master mode to monitor mode and then back. The source code for all these steps is in the appendix.

Man in the Middle Attack

An attacker can successfully implement a Man-in-the-middle attack by first configuring a rogue AP to imitate a legitimate AP, then coerce wireless clients to connect to the rogue AP by performing a denial of service attack against the legitimate AP or by providing a stronger signal than the targeted AP. Wireless clients will normally associate to the AP with the strongest signal or lowest signal to noise ratio (SNR). To make the intercepted connection appear seamless to victims, the rogue AP could then bridge connections to another network connection. If successfully executed an attacker will have complete control of the wireless client's network connection and may perform any inline attack they wish (Deckerd, 2006)

AP impersonation attacks can be done for several purposes, including as a

Detecting and Preventing Rogue Devices on the Network

Man-in-the-middle attack, as a rogue AP attempting to bypass detection, and as a possible honeypot attack. In such an attack, the attacker sets up an AP that assumes the BSSID and ESSID of a valid AP. A successful Man-in-the-middle attack will insert an attacker into the data path between the client and the AP. In such a position, the attacker can delete, add, or modify data, provided he has access to the encryption keys. Such an attack also enables other attacks that can learn a user's authentication credentials. (Aruba Networks Technical Brief, 2007)

After going to the field and running my USB wireless adapter as a rogue AP, I checked the existence of the rogue AP's from the Monitor Program that I mentioned earlier. The program showed 2 cases, the second one being what I was looking for. The first one actually shows that the MAC Address of the AP is changed because it is replaced. 00:02:6F series are 802.11b AP series and 00:15:6D series are the 802.11g series. Basically, the AP changed in that zone. This program also showed me that I need to update the AP MAC list in the system.

09:32:35 PM

Different Mac Address Same SSID							
SSID	BBC_8402001_3	Mac Address	00:02:6F:05:52:D0	Log Date	2007-07-21 16:44:26	Signal	-68
SSID	BBC_8402001_3	Mac Address	00:15:6D:63:29:31	Log Date	2007-07-13 19:45:12	Signal	-57
SSID	BBC_8409503_2	Mac Address	00:13:49:AF:CB:53	Log Date	2007-07-22 21:23:45	Signal	-84
SSID	BBC_8409503_2	Mac Address	00:02:6F:07:CF:19	Log Date	2007-07-13 20:50:00	Signal	-80

Done Internet

Right after the detection which we can see from the third and fourth lines,

Detecting and Preventing Rogue Devices on the Network

we see that it logged the event. The fourth line in the log below is actually the incident traffic. I stopped the logging at 20.41 and at the time I started rogue AP it logged it. I allowed it to log after the incident periodically for a while and moved it back to steady state again.

```
PriRadius:/home/cpestatus# ls -l DUMP_8409503_2_2007-07-22*
-rw-r--r-- 1 cpestatus cpestatus 229753 Jul 22 20:23 DUMP_8409503_2_2007-07-22-20-23-11
-rw-r--r-- 1 cpestatus cpestatus 232001 Jul 22 20:32 DUMP_8409503_2_2007-07-22-20-31-40
-rw-r--r-- 1 cpestatus cpestatus 232126 Jul 22 20:41 DUMP_8409503_2_2007-07-22-20-40-29
-rw-r--r-- 1 cpestatus cpestatus 234971 Jul 22 21:32 DUMP_8409503_2_2007-07-22-21-32-15
-rw-r--r-- 1 cpestatus cpestatus 230886 Jul 22 21:40 DUMP_8409503_2_2007-07-22-21-40-28
-rw-r--r-- 1 cpestatus cpestatus 229313 Jul 22 21:51 DUMP_8409503_2_2007-07-22-21-50-30
-rw-r--r-- 1 cpestatus cpestatus 232532 Jul 22 22:01 DUMP_8409503_2_2007-07-22-22-00-28
-rw-r--r-- 1 cpestatus cpestatus 223231 Jul 22 22:11 DUMP_8409503_2_2007-07-22-22-10-28
-rw-r--r-- 1 cpestatus cpestatus 237435 Jul 22 22:20 DUMP_8409503_2_2007-07-22-22-20-28
-rw-r--r-- 1 cpestatus cpestatus 210370 Jul 22 22:31 DUMP_8409503_2_2007-07-22-22-30-31
```

When we check the database we see the multiple SSID's in the table, which triggered the whole process. The second line below is done manually by me to test the cases. The third line actually is from the USB Adapter AP and the fourth line is from the legitimate AP. Even though the legitimate AP was working on channel 11 I set up the rogue AP to work on channel 1. A Man-in-the-middle attack works when the rogue AP is at least 5 channels away from the legitimate AP.

656991	8402003	10887	BBC_8409501_3	00:15:6D:63:29:36	11	2007-07-13 21:20:25	-89
1210133	8409503	99999	BBC_8409503_1	DD:DD:DD:DD:DD:DD	1	2007-07-21 17:00:21	-99
1304010	8409503	567	BBC_8409503_2	00:13:49:AF:CB:53	1	2007-07-22 21:23:45	-84
655936	8406502	12566	BBC_8409503_2	00:02:6F:07:CF:19	11	2007-07-13 20:50:00	-80
655733	8402001	7214	BBC_8409503_3	00:02:6F:07:7D:91	1	2007-07-13 20:47:40	-80

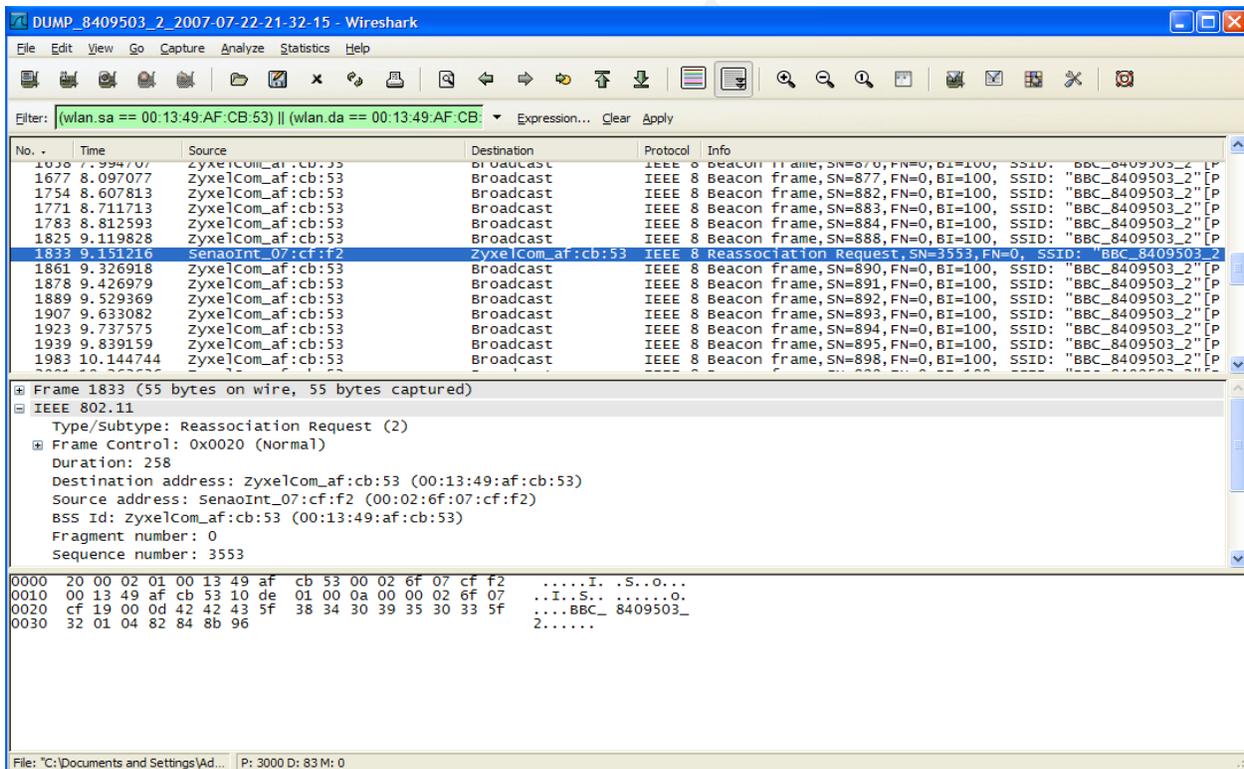
Here are the clients who noticed the rogue AP at the time of the incident. The rogue AP worked for 23 minutes and 3 different clients noticed the existence of the rogue AP. I didn't even boost the signal by using an external antenna.

Detecting and Preventing Rogue Devices on the Network

```
mysql> select * from cpesitesurveys where mac = '00:13:49:AF:CB:53';
+----+-----+-----+-----+-----+-----+-----+-----+
| id   | zoneid | customerid | site_ssid | mac           | channel | LogDate           | signal |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1304010 | 8409503 | 567 | BBC_8409503_2 | 00:13:49:AF:CB:53 | 1 | 2007-07-22 21:23:45 | -84 |
| 1304228 | 8409503 | 8365 | BBC_8409503_2 | 00:13:49:AF:CB:53 | 1 | 2007-07-22 21:26:33 | -54 |
| 1304413 | 8409503 | 567 | BBC_8409503_2 | 00:13:49:AF:CB:53 | 1 | 2007-07-22 21:28:15 | -87 |
| 1305672 | 8409503 | 168 | BBC_8409503_2 | 00:13:49:AF:CB:53 | 1 | 2007-07-22 21:46:22 | -81 |
| 1305798 | 8409503 | 567 | BBC_8409503_2 | 00:13:49:AF:CB:53 | 1 | 2007-07-22 21:47:42 | -87 |
+----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.93 sec)
```

After the incident I transferred the capture file “DUMP_8409503_2_2007-07-22-21-32-15” to my laptop and opened it with Wireshark. I applied the filter “wlan.sa == Rogue AP MAC” or “wlan.da == Rogue AP MAC”

Here is the screen shot:

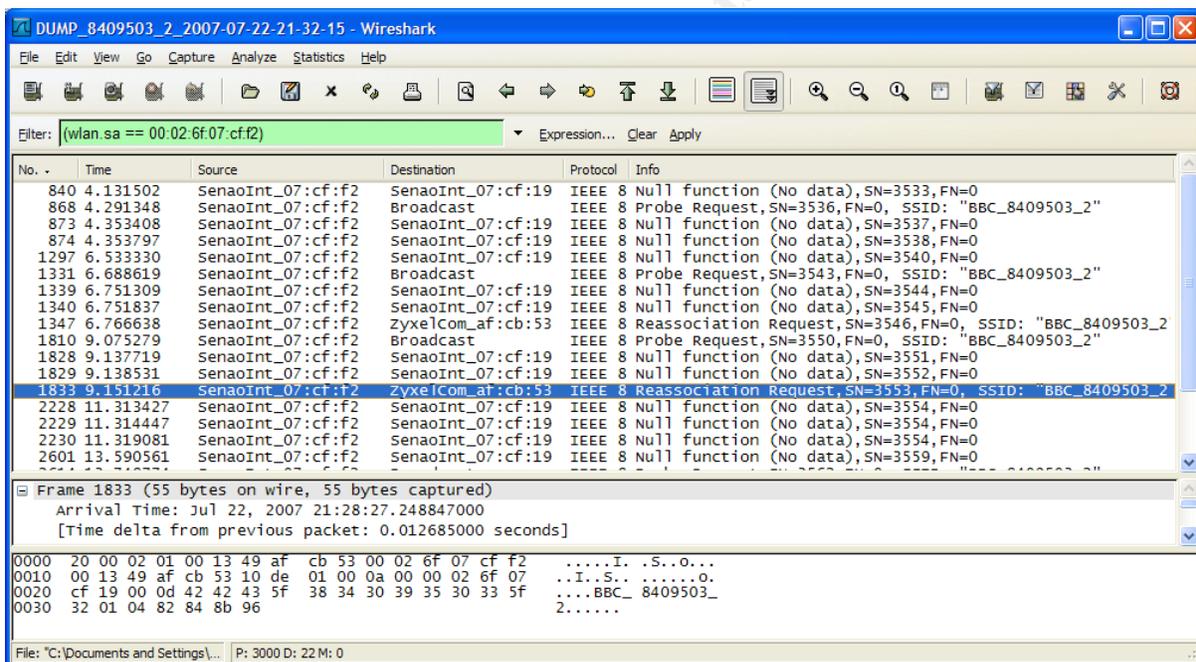


We can see that rogue 00:13:49:AF:CB:53 Zyxel AP is sending beacons to the network. Only one client, 00:02:6f:07:cf:f2, started to communicate with the

Detecting and Preventing Rogue Devices on the Network

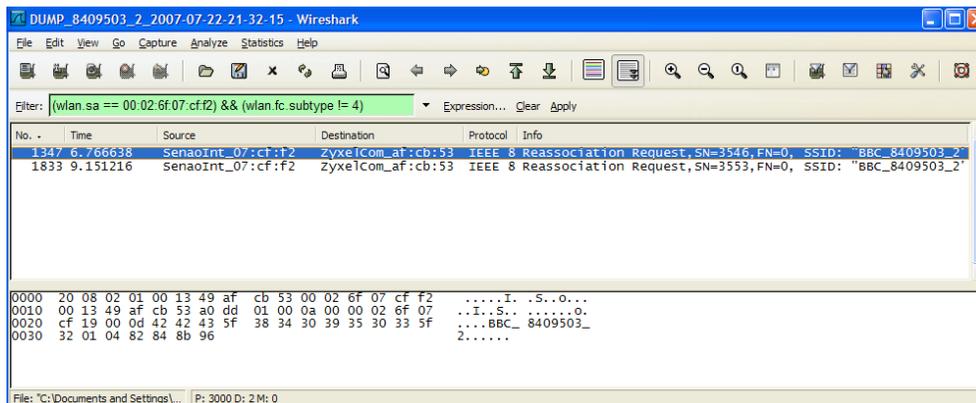
rogue AP. In this traffic we can also notice that all the frames that the rogue AP is sending are beacon frames. Beacon Frames can be transmitted by the AP for polling purposes. The beacon frame sent by the AP contains control information and can be used by mobile stations to locate an AP if it is on active scanning mode. Briefly, rogue AP wants to let everyone know that it exists.

After identifying the only client connecting to the rogue AP I checked the traffic originated from this client by using this filter **"wlan.sa == 00:02:6F:07:CF:F2"**



After that I checked the subtype of the traffic for identification. After adding **"wlan.fc.subtype!=4"** filter all the probe request traffic is gone and only re-association packets left.

Detecting and Preventing Rogue Devices on the Network

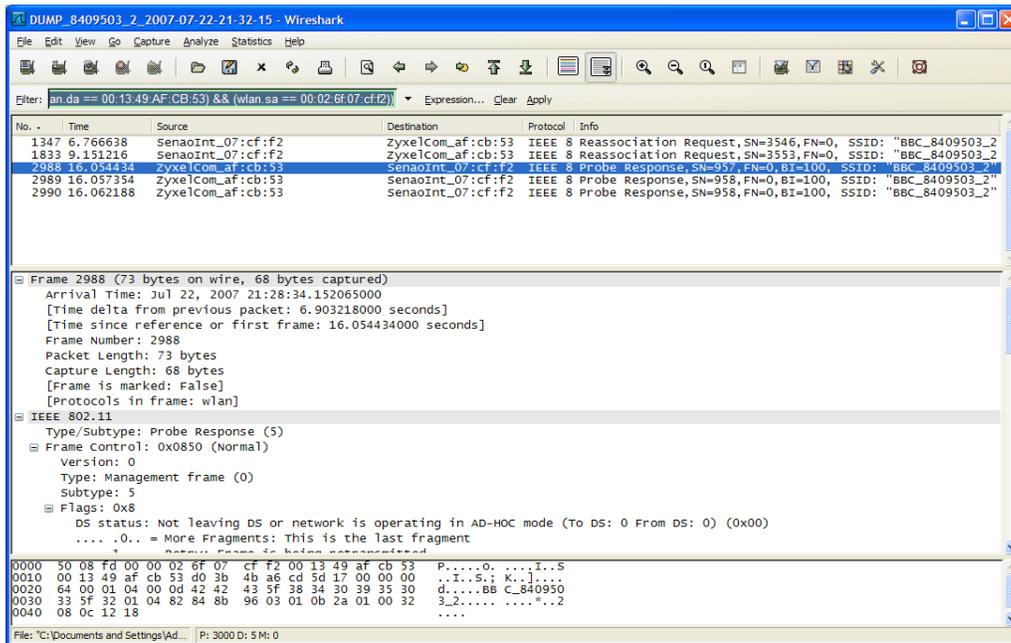


On the other hand, to be able to show the communication between the rogue AP and legitimate client I used the following filter...

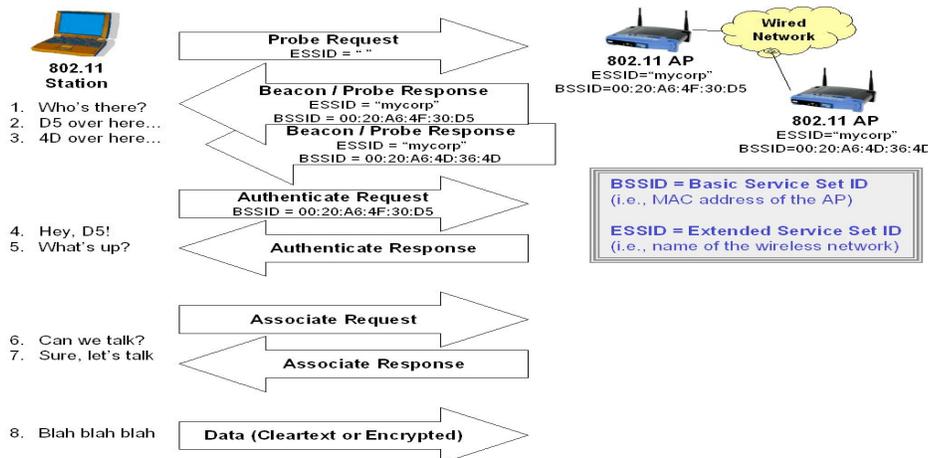
"((wlan.sa == 00:13:49:AF:CB:53) && (wlan.da == 00:02:6f:07:cf:f2)) || ((wlan.da == 00:13:49:AF:CB:53) && (wlan.sa == 00:02:6f:07:cf:f2))"

Here is the screen shot of the result:

Detecting and Preventing Rogue Devices on the Network



Here is a graphical representation of the connection establishment between an AP and a station.



This image is retrieved from <http://www.watchguard.com/archive/images/APhandshake.jpg>

From all the above I came to the following conclusion. The client is continuously sending probe request packets to the rogue AP on a certain channel

Detecting and Preventing Rogue Devices on the Network

but the rogue AP is ignoring those requests. On the other hand the client went through all the probe request/response, authentication, and association phases with the legitimate AP, de-associated and now wants to re-associate again to start transfer traffic. At this point the legitimate AP is down and the rogue AP doesn't fulfill its request. AP in this case somehow locks the client from connecting to the internet.

My intention was to implement Evil Twin attack but eventually I noticed that I essentially implemented Man-in-the-middle attack. The program detected Rogue AP right at the moment. The wireless traffic shows the existence of the rogue AP, struggling client sending probe requests and also trying to re-associate. Meanwhile, the rogue AP is sending beacons. Also site survey shows that 3 clients see the rogue AP too. In case I had set up the rogue AP with the same channel of the legitimate AP, I would have seen successful connections.

Evil Twin Attack

Here are the SSID's and the total number of occurrences in the CPE site survey table.

```
mysql> select count(*) as TOTAL, site_ssid from cpesitesurveys group by site_ssid order by TOTAL DESC limit 10;
+-----+-----+
| TOTAL | site_ssid |
+-----+-----+
| 40811 | ACTIONTEC |
| 34234 | NETGEAR   |
| 24243 | BBC_8412803_3 |
| 23715 | belkin54g |
| 23006 | BBC_8412803_1 |
| 22406 | BBC_8411804_3 |
| 20454 | BBC_8411902_3 |
| 18799 | linksys   |
| 18271 | BBC_8411804_2 |
| 17054 | BBC_8411902_2 |
+-----+-----+
10 rows in set (2.37 sec)
```

Both queries below returned nothing. There was no indication of Evil Twin

Detecting and Preventing Rogue Devices on the Network

attack. What I was trying to do was finding multiple logs for the same exact time from the same client with the same MAC address. It returned nothing, which means there weren't any real Evil Twin attacks occurring in those 18 AP's I was testing.

```
mysql> select customerid, LogDate, mac from cpesitesurveys where site_ssid='ACTIONTEC'  
group by customerid,LogDate, mac having count(*)>1 ;
```

```
mysql> select customerid, LogDate, mac from cpesitesurveys where site_ssid='linksys' group  
by customerid,LogDate, mac having count(*)>1 ;
```

Another method for detecting Evil Twin attack would be to use kismet to find evil twins. PRISM2 header contains Received Signal Strength Indication (RSSI) for every packet. RSSI is a measurement of the received radio signal strength. It is possible to log the traffic and use Wireshark to see the signal on the packets. We need per-packet signal strength for that to work because it watches all the beacon and probe frames and keeps track of their signal strength.

MAC Address List Match

In order to detect rogue access points, the IDS utilize a list of authorized access points then alerts when a detected AP does not match the list. (Vladimirov, Gavrilenko, and Mikhailovsky, 2004)

I kept the lists in the database at the Central Server.

Client MAC Address Match

I used Arp on AP's and a MAC table for clients in the central server. I had a cron job running every 5 minutes which sends an email notification if a MAC

Detecting and Preventing Rogue Devices on the Network

address in the Arp table on the AP doesn't match with the database. I created an email account as GiacMonitor@gmail.com and turned of the authentication for a while and this was the result.



As we can see from the e-mail someone is connected to the AP and we can also tell with proximity of 5 minutes when it happened.

AP MAC Address Match

Another attribute that can be implemented on the wireless IDS is to create a listing or be pre-configured with all known and authorized WAPs, so that whenever an unidentified or rogue WAP is found in the network, the wireless IDS can quickly detect and alert. (Poblete, 2005)

For AP MAC Address Match, I used MAC addresses of the AP's from CPE site survey table and an AP MAC List table in the database at the central server. As long as the list is green there is no indication of spoofing.

Detecting and Preventing Rogue Devices on the Network

Here is a screen shot of the MAC address checking for the AP:

SSID	BBC_7406505_2	Mac Address	00:02:6F:07:D0:3E
SSID	BBC_7406505_3	Mac Address	00:02:6F:07:BE:61
SSID	BBC_7406506_3	Mac Address	00:02:6F:04:2E:E0
SSID	BBC_7406509_1	Mac Address	00:02:6F:05:52:FA
SSID	BBC_7408401_3	Mac Address	00:02:6F:05:52:DB
SSID	BBC_7408403_1	Mac Address	00:02:6F:07:C1:90
SSID	BBC_7408403_3	Mac Address	00:02:6F:07:7F:63
SSID	BBC_7408405_3	Mac Address	00:02:6F:07:7D:A4
SSID	BBC_7408406_1	Mac Address	00:02:6F:07:CF:17
SSID	BBC_7408406_3	Mac Address	00:02:6F:05:4C:16
SSID	BBC_7408701_4	Mac Address	00:02:6F:04:65:EE
SSID	BBC_7408801_1	Mac Address	00:02:6F:07:C3:82
SSID	BBC_7408804_1	Mac Address	00:02:6F:07:D0:60
SSID	BBC_7408804_3	Mac Address	00:02:6F:07:C9:87
SSID	BBC_7409303_2	Mac Address	00:02:6F:07:C9:EF

MAC Address Spoofing

Impersonation attacks in a wireless network typically involve an attacker taking on the address of a valid client or AP and trying to obtain access or services typically reserved for those valid clients or APs. (Aruba Networks Technical Brief, 2007)

MAC address Spoofing is very easy to do. The following command will let someone to change the MAC address of their network card.

```
ifconfig eth1 hw ether AA:BB:CC:DD:00:11
```

Detecting and Preventing Rogue Devices on the Network

Where eth1 is the network interface and AA:BB:CC:DD:00:11 is the MAC address.

The test program checks multiple MAC addresses in the Arp table on the AP. I created a cron job and checked the logs but didn't see a single MAC address spoofing case. I believe people do not bother to sniff the wireless traffic, grab the MAC address of a connected client and try to connect simultaneously or wait until that client finishes its connection and connect afterwards. The only time a rogue client can not be detected is when it spoofs a legitimate MAC address of a wireless network and when the actual client is not connecting. In this case there is no authentication like 802.1x. As long as the list is green there is no indication of spoofing.

```
00:02:6f:07:a4:e2
00:02:6f:07:c8:e4
00:02:6f:07:d3:7d
00:02:6f:07:c2:98
00:02:6f:07:c0:8b
00:02:6f:07:7e:41
```

RF Jamming

RF Jamming is counted as DOS attacks by many, however I think DOS attacks are intentional, but RF Jamming can be unintentional by nature. Therefore I didn't show RF Jamming as a part of DOS attacks.

Baby monitors and other devices that operate on the 2.4 GHz band can disrupt a wireless network using this frequency. These denials of service can

Detecting and Preventing Rogue Devices on the Network

originate from outside the work area serviced by the access point, or can inadvertently arrive from other 802.11b devices installed in other work areas that degrade the overall signal. (Internet Security Systems White Paper, 2007)

Disruptions can be caused by noise from microwaves, cordless phones, or other appliances that operate on the 2.4 GHz radio frequency on which 802.11b wireless LANs also operate. Disruptions can also be caused by hackers using access points to send dissociate commands. (Lane, 2005)

RF technology is very dynamic in nature as it changes in response to the real-world conditions. Noise, interference, and increased traffic load, signal attenuation – these are some of the factors that can cause RF topology to change from one moment to the next. For most network managers RF technology has remained something that is always in the air. (Anand, 2004)

RF jamming is used to take down an entire wireless LAN by overwhelming the radio environment with high-power noise. Channel or power adjustments to overcome the interference can be done (Aruba Networks Technical Brief, 2007)

I developed 2 small programs to check noise both on the CPE level and CAU level. I leveraged another Perl script which also is doing site survey and logs the data in a CPE log table. The main difference between this table and the CPE site survey table is that this can also check the noise around it. Here are two screen shots from these logs showing noisy AP's and clients. All the screen shots shown in this paper have small programs behind them and the source code is attached in the appendix.

Detecting and Preventing Rogue Devices on the Network

Noisy AP List		
Access Point	UserName	CAUNoise
BBC_8411804_3	chapmanpwr	-92
BBC_8412803_3	twinslow	-94
BBC_8411902_2	loengo	-95
BBC_8402001_1	kcrump	-95
BBC_8411902_1	walter	-95
BBC_8402001_2	rijorgensen	-95
BBC_8402001_3	debthompson1	-95
BBC_8411902_3	bbarton	-95
BBC_8412803_1	brtregemba	-95
BBC_8406502_2	ciegbert	-96
BBC_8406502_1	cevensen	-98
BBC_8411804_1	edwinjohn	-99
BBC_8406502_3	ataylor	-100
BBC_8402003_1	cdendtsen	-100
BBC_8402003_3	lanes	-100
BBC_8402003_2	markhansen	-100
BBC_8412803_2	jessegreen	-100
BBC_8411804_2	tmetcalf	-100

Noisy Client List		
MAC Address of the Client	UserName	NoiseLevel
00:02:6F:07:7D:4D	lloydron	-83
00:02:6F:07:9C:1A	dfowlerw	-88
00:02:6F:07:A6:C9	etchief2005	-89
00:02:6F:07:A4:FE	jmadrighal	-90
00:02:6F:07:BE:95	smilefamily	-91
00:02:6F:07:BD:EE	kjl	-91
00:02:6F:07:9D:AF	bbarton	-91
00:02:6F:07:A2:66	shaehealey	-91
00:02:6F:07:7F:59	jaesunh	-91
00:02:6F:07:C6:95	dtilby	-92
00:02:6F:07:BE:1C	wwoodard	-92

DOS Attacks

DOS attacks are designed to prevent or inhibit legitimate users from accessing the network. This includes blocking network access completely, degrading network service, and increasing processing load on clients and network

equipment. (Aruba Networks Technical Brief, 2007)

Denial of service attacks are also easily applied to wireless networks, where legitimate traffic can not reach clients or the access point because illegitimate traffic overwhelms the frequencies. An attacker with the proper equipment and tools can easily flood the 2.4 GHz frequency, corrupting the signal until the wireless network ceases to function. (Internet Security Systems White Paper, 2007)

Saturating an AP or client with requests is commonly used to deny service. To ensure WLAN high availability, flooding attacks need to be detected. These attacks include associate, re-associate, and disassociate requests, probe requests, and other reserved management frame flooding (Subtypes 6, 7, D, E, and F). (3Com Corporation White Paper, 2007) Some of the common DoS attacks are RF jamming attack, FataJack attack, Duration attack, Authentication storm, De-authentication storm, Association storm, Disassociation storm etc. (Anand, 2004)

Duration Attack

When a client sends frames with prolonged duration, other clients in the network have to wait till the specified duration to use the RF medium. If the client continuously sends frames with such high duration, then it can prevent other clients from using RF medium and remain unassociated forever. WLAN devices perform virtual carrier sensing prior to using the RF medium. Carrier sense minimizes the likelihood of two devices transmitting simultaneously. Wireless nodes reserve the right to use the radio channel for the duration specified in the frame. (Anand, 2004)

Detecting and Preventing Rogue Devices on the Network

The duration value in the frame indicates the duration in milliseconds for which the channel is reserved. The Network Allocation Vector (NAV) stores this duration information and is traced for every node. The basic rule is that any node can transmit only if the NAV reaches zero or in other words no one has reserved the channel at that time. Attackers take advantage of the NAV. An attacker can send frames with huge duration values. This would force other nodes in the range to wait till the value reaches zero. (Anand, 2004)

Deauthentication

Spoofed deauthenticate frames form the basis for most denial of service attacks, as well as the basis for many other attacks such as Man-in-the-middle attack. A Linux driver called AirJack typically forms the basis for this type of attack, with tools such as WLAN-Jack and Fake-Jack actually carrying out the attack. Broadcast deauthentication attack generates spoofed deauthenticate frames with a broadcast destination address instead of disconnecting a single station, the intent is to disconnect all stations attached to a given AP. Typically, a Linux tool known as “Hunter-Killer” is used to generate this attack. FakeAP is a tool originally created to thwart war drivers by flooding beacon frames containing hundreds of different addresses. This would appear to a war driver as though there were hundreds of different APs in the area, thus concealing the real AP. (Aruba Networks Technical Brief, 2007)

Deauthentication attacks are one of the most common DOS attacks. This is usually the initial step an attacker takes to create DOS attack against an AP with the intention to make it not respond and force the clients to use the rogue AP. I

Detecting and Preventing Rogue Devices on the Network

implemented it on one of the test AP's by using aireplay-ng.

Here is the command.

```
aireplay-ng -0 1 -a 00:14:6C:7E:40:80 -c 00:0F:B5:34:30:30 ath0
```

Where:

- -0 means deauthentication
- 1 is the number of deauths to send (you can send multiple if you wish); 0 means send them continuously
- -a 00:14:6C:7E:40:80 is the MAC address of the access point
- -c 00:0F:B5:34:30:30 is the MAC address of the client to deauthenticate; if this is omitted then all clients are deauthenticated
- ath0 is the interface name

(Aircrack-ng Contributors, 2007)

I used a live BackTrack2 CD to implement the attack. I believe it is the simplest way of running the attack. I had both Atheros (installed by default) and external Prism card in my laptop. After booting from a live BackTrack2 CD, I set up my Prism card to monitor and my Atheros card to create the attack.

While in RFMON mode, wireless clients are unable to transmit any frames; their cards are only able to receive, and therefore capture traffic. This limits the client to reporting only current or recorded network traffic. For instance, a client using passive monitoring would be able to report on the MAC addresses and number of associations to a discovered AP, but would be unable to probe the discovered AP for SNMP MIB information. (Wright, 2002)

Detecting and Preventing Rogue Devices on the Network

I put my Prism card to monitor mode like this:

```
iwconfig eth1 mode monitor
```

And after that I started the tcpdump packet capture for analysis:

```
tcpdump -i eth1 -w DeAuthtest
```

Meanwhile I destroyed my Atheros card interface first, then created a new virtual interface assigning it to wifi0 and putting it in the monitor mode like this

```
bt ~ # wlanconfig ath0 destroy
bt ~ # wlanconfig ath0 create wlandev wifi0 wlanmode monitor
```

I checked whether the card is up and made sure that it is up and in monitor mode.

```
bt ~ # ifconfig ath0 up
bt ~ # iwconfig

ath0 IEEE 802.11g ESSID:"" Nickname:""
Mode:Monitor Frequency:2.462 GHz Access Point: 00:19:7D:4A:97:96
Bit Rate:0 kb/s Tx-Power:31 dBm Sensitivity=0/3
Retry:off RTS thr:off Fragment thr:off
Encryption key:off
Power Management:off
Link Quality=0/94 Signal level=-95 dBm Noise level=-95 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

I moved to the folder where aireplay-ng is and started the attack. -0 means that it is a deauthentication attack and 0 means that it will be continuously. On the other hand, -a means the AP's MAC address and -c means the client's MAC address and ends with the interface that the attack would generate from. To be able to start the attack the interface should be up and in monitor mode. Omitting -c will make the attack broadcast and it effects all the connected clients at the AP.

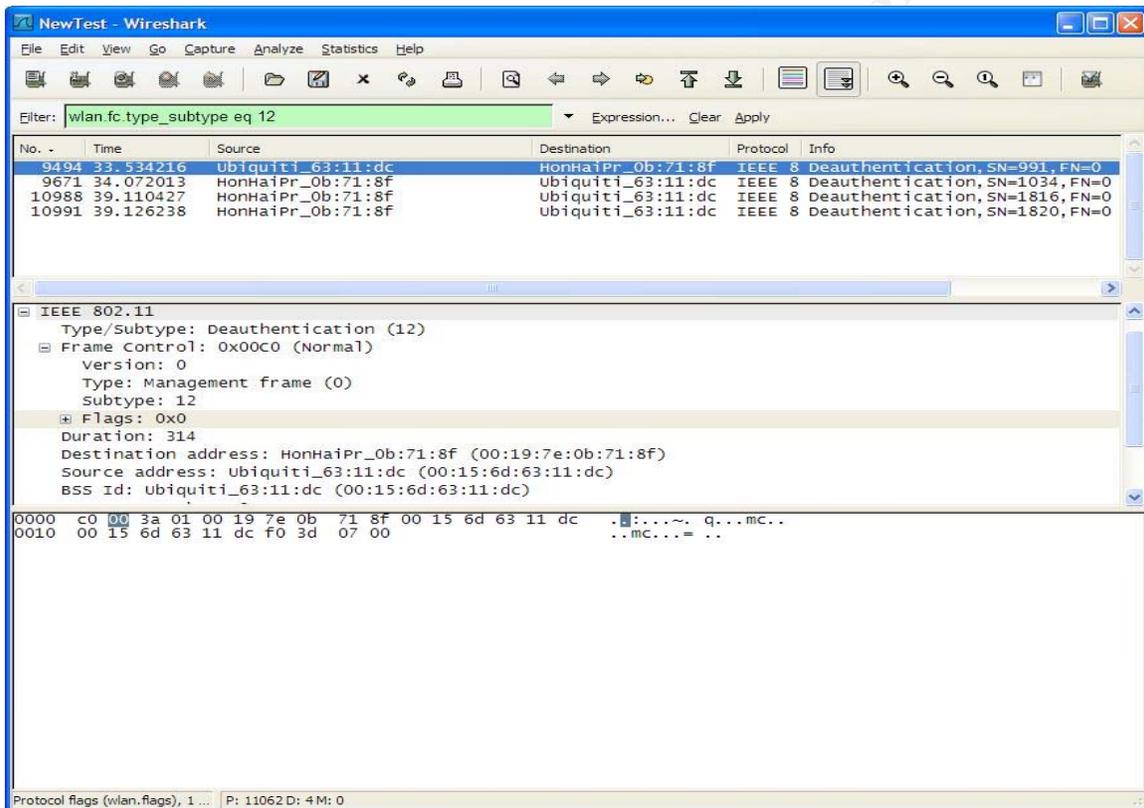
```
bt ~ # cd /pentest/wireless/aircrack-ng/
bt aircrack-ng # aireplay-ng -0 0 -a 00:15:6D:63:11:DC -c 00:19:7E:0B:71:8F ath0
```

```
19:04:29 Sending DeAuth to station -- STMAC: [00:19:7E:0B:71:8F]
19:04:30 Sending DeAuth to station -- STMAC: [00:19:7E:0B:71:8F]
```

Detecting and Preventing Rogue Devices on the Network

```
19:04:31 Sending DeAuth to station -- STMAC: [00:19:7E:0B:71:8F]
19:04:32 Sending DeAuth to station -- STMAC: [00:19:7E:0B:71:8F]
19:04:34 Sending DeAuth to station -- STMAC: [00:19:7E:0B:71:8F]
19:04:35 Sending DeAuth to station -- STMAC: [00:19:7E:0B:71:8F]
19:04:36 Sending DeAuth to station -- STMAC: [00:19:7E:0B:71:8F]
19:04:37 Sending DeAuth to station -- STMAC: [00:19:7E:0B:71:8F]
```

Here is Wireshark capture that the attack is actually working against the client 00:19:7E:0B:71:8F



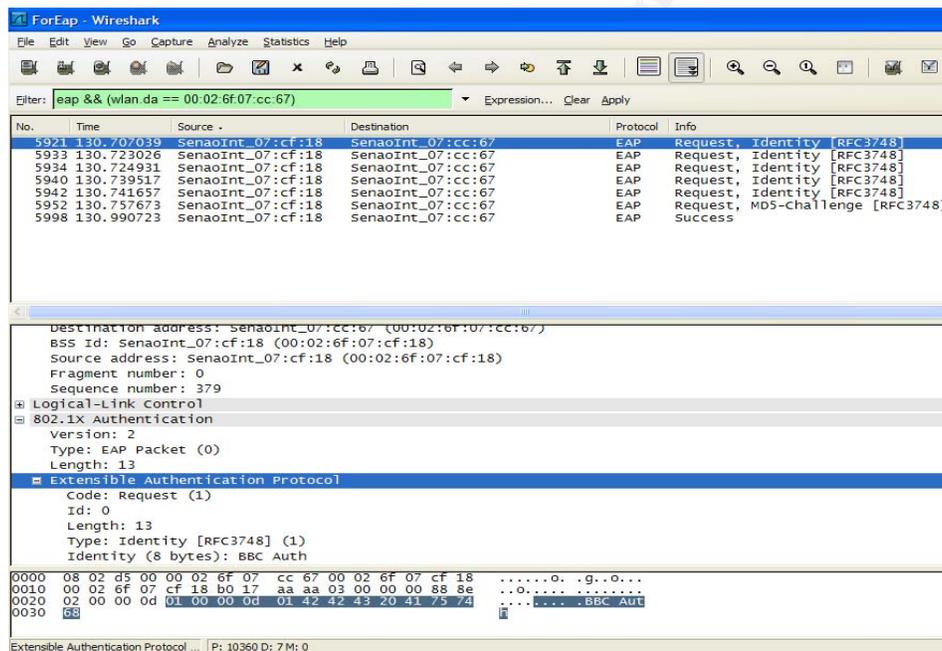
801.1x EAP Authentication

A skilled attacker captures wireless traffic using techniques similar to those employed on wired networks. Many of these tools capture the first part of the connection session, where the data would typically include the username and password. An intruder can then masquerade as a legitimate user by using this

Detecting and Preventing Rogue Devices on the Network

captured information to hijack the user session and issue unauthorized commands. (Internet Security Systems White Paper, 2007)

I logged in to one of the test AP's and forced it to reboot. On another AP I captured the traffic. As we can see after the initial 3 steps (Probe Request and Response, Authentication, Association) EAP Authentication phase starts. AP starts the authentication and after it gets the MD-5 Challenge it checks it and replies with a success message.



No.	Time	Source	Destination	Protocol	Info
5921	130.707029	SenaoInt_07:cf:18	SenaoInt_07:cc:67	EAP	Request, Identity [RFC3748]
5933	130.723026	SenaoInt_07:cf:18	SenaoInt_07:cc:67	EAP	Request, Identity [RFC3748]
5934	130.724931	SenaoInt_07:cf:18	SenaoInt_07:cc:67	EAP	Request, Identity [RFC3748]
5940	130.739517	SenaoInt_07:cf:18	SenaoInt_07:cc:67	EAP	Request, Identity [RFC3748]
5942	130.741657	SenaoInt_07:cf:18	SenaoInt_07:cc:67	EAP	Request, Identity [RFC3748]
5952	130.757673	SenaoInt_07:cf:18	SenaoInt_07:cc:67	EAP	Request, MD5-Challenge [RFC3748]
5998	130.990723	SenaoInt_07:cf:18	SenaoInt_07:cc:67	EAP	Success

Destination address: SenaoInt_07:cc:67 (00:02:6f:07:cf:18)
BSS Id: SenaoInt_07:cf:18 (00:02:6f:07:cf:18)
Source address: SenaoInt_07:cf:18 (00:02:6f:07:cf:18)
Fragment number: 0
Sequence number: 379

Logical-Link Control
802.1X Authentication
Version: 2
Type: EAP Packet (0)
Length: 13

Extensible Authentication Protocol
Code: Request (1)
Id: 0
Length: 13
Type: Identity [RFC3748] (1)
Identity (8 bytes): BBC Auth

0000 08 02 d5 00 00 02 6f 07 cc 67 00 02 6f 07 cf 180..g..o...
0010 00 02 6f 07 cf 18 b0 17 aa aa 03 00 00 00 88 8e
0020 02 00 00 0d 01 00 00 0c 01 42 42 45 20 41 75 74BBC Auth
0030

Extensible Authentication Protocol ... | P: 10360 D: 7 M: 0

7. Prevention

Once a rogue AP is discovered the next immediate step is to block the AP from the network so that the authorized clients don't associate with it. There are two ways of blocking the rogue APs. 1. Tit for Tat: Launch a Denial-of-service (DOS) attack on the rogue AP and make it deny wireless service to any new client.

Ibrahim Halil Saruhan

39

2. Pull it out of the network: Either the WLAN administrator can manually locate the AP and pull it physically off the LAN OR block the switch port to which the AP is connected. (Anand, 2004)

In our testing case choice number 1 is reasonable. By using tools like void11, hunter-killer or aireplay-ng we can create DOS attack on the rogue AP so that we can make it sure that it denies wireless service to any new client.

The access control list is the simplest security measure we can find in a wireless network. The protection offered by this mechanism mainly consists of filtering out unknown users and requires a list of authorized client's MAC addresses to be loaded in the Access Point. Only those registered MAC addresses will be able to communicate with the Access Point, and will drop any communication that come from others not registered MAC addresses. (Wong, 2004)

Once a rogue client is detected, WLAN administrator should shut down the client from the network. The most common method of keeping rogue clients away is by configuring their MAC address in the Access Point's Access Control List (ACL). ACL determines whether to deny or allow a client to connect to the AP. WLAN administrators can specify the rogue client's MAC address in the ACL of all authorized Access Points to keep the rogue client off the network for ever. (Anand, 2004)

Keeping ACL's in an AP is a way of keeping rogue clients away. This project on the other hand checks Arp cache table, get's the IP-MAC pairs and compare them with the MAC addresses from the database at the Central Server. So the difference is between keeping the list in AP or a Central Server. I think both ways

Detecting and Preventing Rogue Devices on the Network

are pretty much protective.

802.11 frames are trivial to forge. WPA RADIUS looks the most secure solution with PEAP as the authentication method or EAP-TLS, which are equivalently secure.

Infrastructure mode is the most common operation mode in which we could find wireless networks. In this operation mode, each wireless client connects directly to a central device called Access Point; there is no direct connection between others wireless clients. (Wong, 2004)

Two wireless clients can talk directly to each other, bypassing the access point. Users therefore need to defend clients not just against an external threat but also against each other. (Internet Security Systems White Paper, 2007)

Preventing clients to talk to each other is called Client Isolation. Client isolation should be activated on the AP. It prevents wireless clients from talking to one another. The typical way for someone to attack one of the clients is to inject packets into the AP that the AP dutifully repeats to the client and they should be prevented from doing so because they have no business talking to each other directly. This is also called as "AP isolation". In our system it is implemented by using hostapd. This way an attacker can now only attack the AP. To attack the client directly they'd have to spoof frames from the AP and be close enough to the client that the client can hear them which significantly reduces the attacker's range and potential targets

Clever deployers of rogue access points have been known to purchase

Detecting and Preventing Rogue Devices on the Network

unauthorized 802.11a devices on the theory that the existing network is not capable of detecting them. (Gast, 2005)

The current set-up in this paper recognizes only 802.11b and 802.11g AP's because it works on the open 2.4 GHz frequency. 802.11g is an upgrade to 802.11b, and supports up to 54Mbps. 802.11 also supports 54Mbps, but on 5.0 GHz, which makes it impossible for the 802.11b CPE's to recognize it. There should be other measures to detect 802.11a AP's.

When a client exceeds a pre-specified rate of 802.11 associate, disassociate, or re-associate packets, it should automatically be put on a blacklist for a pre-specified amount of time to allow an administrator's assessment of the situation. Since spoofed de-authenticate frames are the basis for DOS and Man-in-the-middle attacks, detecting these frames provides a security alert. Detecting probe responses containing a null SSID that will disable a number of popular network interface cards can also deter attacks. (3Com Corporation White Paper, 2007)

Most client intrusion attempts are handled by higher-layer security functions. However, one serious lower-layer attack that exploits client weaknesses is the honeypot AP. A "honeypot" has a number of connotations in the security world. When discussing wireless LANs, one meaning is an attacker's AP that is set up in close proximity to an enterprise, advertising the ESSID of the enterprise. The goal of such an attack is to lure valid clients to associate to the honeypot AP. From that point, a Man-in-the-middle attack can be mounted, or an attempt can be made to learn the client's authentication credentials. Most client devices have no way of distinguishing between a valid AP and an invalid one – the devices only look for a

particular ESSID and will associate to the nearest AP advertising that ESSID.

(Aruba Networks Technical Brief, 2007)

In my testing system, each AP has a different ESSID based on their location. Clients perform site surveys and collect data about AP's around them. This data is stored in the database. A rogue AP will be recognized by multiple clients and site survey table will show whether it is an AP with a different MAC Address, possibly indicating a Man-in-the-middle attack, or a rogue AP seen by many clients at the same time and same MAC address, which most probably indicates the existence of the Evil Twin attack.

Rogue detection and blocking is a continuous process involving at least three components:

- A dedicated piece of hardware probe/sensor to monitor the air and identify network behavior
- A central IDS engine that gathers inputs from many such probes/sensors and helps in pinpointing a device as rogue.
- A network management software that can talk to the wired network, identify the switch port to which the rogue AP is connected and shutdown the port.

(Anand, 2004)

8. Conclusion

Site survey on the client side is a very effective method. By using this data I'm able to implement rogue device detection without installing dedicated devices on the network.

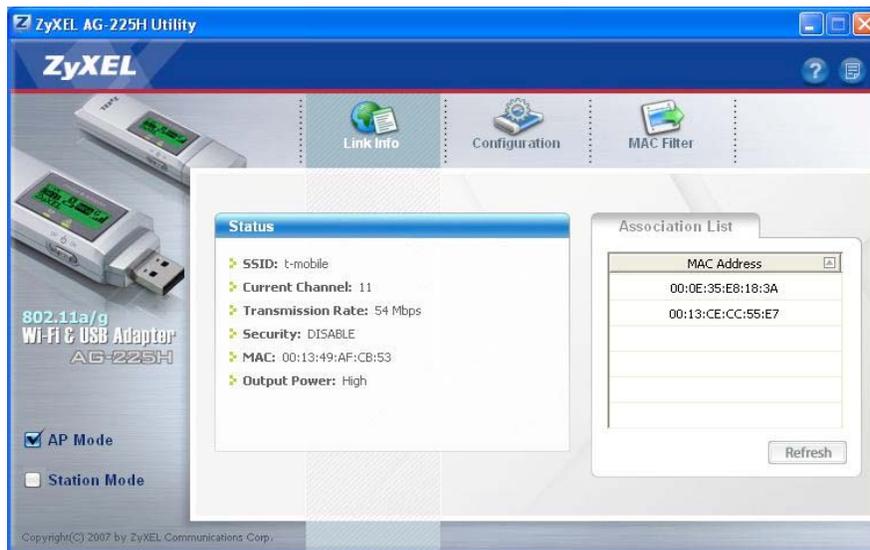
All the other techniques like MAC list checking, MAC spoofing checking, and RF jamming checking are additional tools to detect rogues on the network.

I believe this set-up allows me to watch as many AP's as I want for rogues. It is scalable. This paper approached the problem of rogue detection and prevention from a different perspective.

Actually, I checked the database for the weeks of logs and didn't see a single rogue AP which has a spoofed MAC address. I also checked the CPE site survey results coming from almost 300 clients for more than 2 weeks and didn't find a single case there either. I also checked for other well known default SSID's like Linksys and NETGEAR and I wasn't able to see a single case there either.

I think Rogue AP's are mostly implemented in airports or Starbucks coffee shops. I was checking connectivity in the airport yesterday and wasn't able to connect any AP. I decided to change the mode of my wireless USB adapter from station mode to AP mode and used the broadcast SSID t-mobile. Suddenly, I saw connecting clients on the MAC list of my AP. Here is the screen shot of it:

Detecting and Preventing Rogue Devices on the Network



I did a few experiments on the field by using my USB wireless adapter and created rogue AP's and detected them. The solution is working as it is intended to work.

9. Future Work

Detection of the physical location of an attack is a critical aspect of a wireless IDS. For the most common standard 802.11, attacks are often carried out from a close proximity to a WAP, and most likely are carried out in an extremely short period to further avoid detection. (Poblete, 2005)

With the current set-up we can guess where the rogue device is, but a sophisticated solution to the project which allows the system not only detect the rogue, but also detect it's physical location would be a nice addition.

By using tools like void11 or hunter-killer we can create DOS attack on the rogue AP. I believe it would be a nice addition to test each of these along with

Detecting and Preventing Rogue Devices on the Network

other tools. After that, we should check the site survey and logged traffic with the proposed system. Some of these tools are card dependent. For example void11 works with Prism cards but not with Atheros cards.

The Snort-Wireless project is an attempt to make a scalable, and not to mention free 802.11 intrusion detection system that can easily be integrated into an IDS infrastructure. It is completely backwards compatible with Snort 2.0.x and adds several additional features. Currently it allows for 802.11 specific detection rules through the new "WiFi" rule protocol, as well as rogue AP, AdHoc network, and Netstumbler detection (Poblete, 2005)

Snort-wireless is a wireless IDS designed to integrate into a Snort 2.x environment. Snort is the most widely deployed open source IDS, so a wireless plug-in makes sense for many enterprises. Snort-wireless allows for custom rules to be created based on framing information from a wireless packet. It also contains rules to attempt to find rogue access points, wardrivers, and ad hoc networks. (Potter, 2007)

The next step might be adding Snort-Wireless to the current project. Site survey, MAC address list control, spoofing control and all the other cases already logs the intrusions, but they can be more centralized, log the traffic and automatically forward the traffic to snort-wireless.

Monitoring interface should hop between the 12 channels available to wireless networks. Several wireless attacks work by utilizing a rogue AP on a different channel. For instance monkey in the middle attacks utilize a rogue AP that is at least 5 channels away from the target AP. Without channel hoping wireless IDS

would be blind to attacks that function on other channels. (Deckerd, 2006)

This project doesn't do channel hopping while logging the traffic. However site survey functionality logs the channel data. In case the system detects a rogue AP, it will also know on which channel the rogue is working. It currently logs the traffic on the channel of the legitimate AP; however it would be a nice addition to log the traffic on all channels at the time of intrusion.

10. References

1. Ellingson, Jorgen. (2001). Layers One & Two of 802.11 WLAN Security. Retrieved July 22,2007, from http://www.giac.org/certified_professionals/practicals/GSEC/0996.php
2. Aircrack-ng Contributors. (07/22/2007). Aircrack-ng Deauthentication . Retrieved July 22,2007, from <http://www.aircrack-ng.org/doku.php?id=deauthentication>
3. Gast, Matthew. 802.11 Wireless Networks: The Definitive Guide. Sebastopol, CA: O'Reilly, 2005
4. Vladimirov, Andrew A., Konstantin V. Gavrilenko, and Andrei A. Mikhailovsky. Wi-Foo: The Secrets of Wireless Hacking. Boston: Addison-Wesley, (2004)
5. Lane, Heather D. (02/6/2005). Security Vulnerabilities and Wireless LAN Technology. Retrieved July 22,2007, from http://www.giac.org/certified_professionals/practicals/GSEC/4383.php

Detecting and Preventing Rogue Devices on the Network

6. Wireless LANs: Assuring Enterprise Security and Identity Awareness. (07/22/2007). 3Com Corporation White Paper. Retrieved July 22,2007, from http://www.3com.com/corpinfo/en_US/technology/tech_paper.jsp?DOC_ID=230943
7. Potter, Bruce. (07/22/2007). Wireless Intrusion Detection. Retrieved July 22,2007, from <http://www.itsec.gov.cn/webportal/download/88.pdf>
8. Poblete, Oliver. (01/24/2005). An Overview of the Wireless Intrusion Detection System. Retrieved July 22,2007, from http://www.giac.org/certified_professionals/practicals/GSEC/4296.php
9. Wireless LAN Security 802.11b and Corporate Networks. (07/22/2007). Internet Security Systems White Paper. Retrieved July 22,2007, from http://www.iss.net/documents/whitepapers/wireless_LAN_security.pdf
10. Anand, Dev. (2004). Effective WLAN Management With Distributed RF Sensors. An Adventnet Technical Whitepaper. Retrieved July 22,2007, from http://manageengine.adventnet.com/products/wifi-manager/rfsensor_whitepaper.pdf
11. Anand, Dev. (2004). Rogue Detection and Blocking. An Adventnet Technical Whitepaper. Retrieved July 22,2007, from <http://manageengine.adventnet.com/products/wifi-manager/rogue-detection-and-blocking.pdf>

Detecting and Preventing Rogue Devices on the Network

12. Wright, Joshua. (11/08/2002). Layer 2 Analysis of WLAN Discovery Applications for Intrusion Detection. Retrieved July 22,2007, from http://www.rootsecure.net/content/downloads/pdf/wlan_ids.pdf
13. Deckerd, Gary. (11/23/2006). Wireless Attacks from an Intrusion Detection Perspective. Retrieved July 22,2007, from http://www.giac.org/certified_professionals/practicals/GCIA/0878.php
14. Hutchison, Ken. (10/18/2004). Wireless IntrusionDetection Systems. Retrieved July 22,2007, from http://www.giac.org/certified_professionals/practicals/GSEC/4210.php
15. Wong, Luis Carlos. (10/21/2004). An Overview of 802.11 Wireless Network Security Standards & Mechanisms. Retrieved July 22,2007, from http://www.giac.org/certified_professionals/practicals/GSEC/4185.php
16. Gast, Matthew. 802.11 Wireless Networks: The Definitive Guide. Sebastopol, CA: O'Reilly, 2005
17. Wireless Intrusion Protection (07/22/2007). Aruba Networks Technical Brief. Retrieved July 22,2007, from http://www.arubanetworks.com/pdf/technology/tb_wip.pdf
18. Beacon Frame. (07/22/2007). In Wikipedia, The Free Encyclopedia. Retrieved July 22, 2007, from http://en.wikipedia.org/wiki/Beacon_frame
19. Phifer, Lisa. (07/22/2007). Anatomy of a Wireless "Evil Twin" Attack. Retrieved July 22, 2007, from

<http://www.watchguard.com/infocenter/editorial/27061.asp>

11. Appendix

Source Code

Client MAC Address List Check

```
function CPeMacAddressCheck($CpeZoneId){

    $result      = mysql_fetch_assoc(mysql_query("SELECT NasIPAddress as CauIpAddress FROM
    RadiusNas WHERE (zone = $CpeZoneId)"));
    $CauIpAddress = $result[CauIpAddress];
    $conn        = ssh2_connect($CauIpAddress, 22);

    if (ssh2_auth_password($conn, 'UserName', 'Password'))
        echo "<br><center>Connected to the CAU at $CpeZoneId.</center><br>";
    else
        die('<br><center>Authentication is Failed...</center><br>');

    $arpcommand = " /usr/sbin/arp -e | grep '172.16.[4-6]' | sort | /usr/bin/awk
    '/ether/ {print \$3}'";
    $stream      = ssh2_exec($conn, $arpcommand);

    $Dump       = '';
    print "<table align= 'center' border= '1px solid #BDBABD' border-collapse='collapse'
    bgcolor='#F7F7F7'>";

    while($line = fgets($stream)){
        $line = trim(strtolower($line));
        $result = mysql_query("SELECT count(*) as TOTAL FROM MacList WHERE CPE_MAC =
        '$line'");
        $note = mysql_fetch_assoc($result);
        if ($note[TOTAL] > 0)
            $line = '<font color="green">' . $line . '</font>' ;
        else
            $line = '<font color="red">' . $line . '</font>' ;
        $Dump = $Dump . '<tr><td>' . $line . '</td></tr>' ;
    }
    print "$Dump";
    print "</table>";
}
```

Client MAC Address Spoofing Check

```
function CPESpoofingCheck ($CpeZoneId){

    $result      = mysql_fetch_assoc(mysql_query("SELECT NasIPAddress as CauIpAddress FROM
    RadiusNas WHERE (zone = $CpeZoneId)"));
    $CauIpAddress = $result[CauIpAddress];

    $conn        = ssh2_connect($CauIpAddress, 22);

    if (ssh2_auth_password($conn, 'UserName', 'Password'))
```

Detecting and Preventing Rogue Devices on the Network

```
    echo "<br><center>Connected to the CAU at $CpeZoneId.</center><br>";
else
    die('<br><center>Authentication is Failed...</center><br>');

$arpcommand = " /usr/sbin/arp -e | grep '172.16.[4-6]' | /usr/bin/awk '/ether/
{print $3}' | sort";
$stream = ssh2_exec($conn, $arpcommand);
$Dump = '';
print "<table align= 'center' border= '1px solid #BDBABD' border-collapse='collapse'
bgcolor='#F7F7F7'>";

$previousline = 'none';

while($line = fgets($stream)){
    $line = trim(strtolower($line));
    if ($line == $previousline)
        $line = '<font color="red">' . $line . '</font>' ;
    else
        $line = '<font color="green">' . $line . '</font>' ;
    $Dump = $Dump . '<tr><td>' . $line . '</td></tr>' ;
    $previousline = $line;
}
print "$Dump";
print "</table>";
}
```

AP MAC Address List Check

```
function CAUMacAddressCheck ($site_ssid, $mac){

    $zoneid = substr($site_ssid, 4, 7);
    $radio = substr($site_ssid, 12, 1);
    $ReturnValue = false;

    if ($radio == 1) $tempRadio = 'radio1_mac';
    if ($radio == 2) $tempRadio = 'radio2_mac';
    if ($radio == 3) $tempRadio = 'radio3_mac';
    if ($radio == 4) $tempRadio = 'radio4_mac';

    $SQLQuery = "SELECT" . ' ' . $tempRadio . ' ' . "FROM serial WHERE ssid = '$zoneid'
ORDER BY install_date DESC";
    $result = mysql_query($SQLQuery);
    $note = mysql_fetch_assoc($result);
    if (strtoupper($note[$tempRadio]) == strtoupper($mac))
        $ReturnValue = true;
    return ($ReturnValue);
}

function CAUMacAddressControl(){

    $result = mysql_query("SELECT * FROM cpesitesurveys WHERE site_ssid like '%BBC%'
OR site_ssid like '%TBZ%' GROUP BY mac ORDER BY site_ssid");
    print "<table align= 'center' border= '1px solid #BDBABD' border-collapse='collapse'
bgcolor='#F7F7F7'>";
    print "<tr><td align='center' colspan=4><b>Mac Address Control</b></td></tr>";
    while($note = mysql_fetch_assoc($result))
        if (CAUMacAddressCheck ($note[site_ssid], $note[mac]))
            print "<tr><td><b>SSID</b></td><td><font
color='green'><b>$note[site_ssid]</b></td><td><b>Mac
Address</b></td><td><font
color='green'><b>$note[mac]</b></font></td></tr>";
        else
            print "<tr><td><b>SSID</b></td><td><font
color='green'><b>$note[site_ssid]</b></td><td><b>Mac
Address</b></td><td><font
color='green'><b>$note[mac]</b></font></td></tr>";
}
```

Detecting and Preventing Rogue Devices on the Network

```
print "<tr><td><b>SSID</b></td><td><font
color='red'><b>$note[site_ssid]</b></td><td><b>Mac
Address</b></td><td><font color='red'><b>$note[mac]</b></font></td></tr>";
print "</table>";
}
```

Main Monitor

```
include ("wids/EvilTwin.func.php");
include ("wids/CAUMacAddressControl.func.php");
include ("wids/CPeMacAddressControl.func.php");
include ("wids/RFJamming.func.php");
include ("wids/Traffic.func.php");

if(isset($_POST['EvilTwin'])){echo date('h:i:s A');EvilTwinLogDate();echo date('h:i:s A');}
elseif(isset($_POST['InsertEvil'])){ echo date('h:i:s A');
InsertEvilTwin($_POST['ZoneInsertEvil'], $_POST['RadioInsertEvil']);echo date('h:i:s A');}
elseif(isset($_POST['CAUMacAddressControl'])){ echo date('h:i:s A');CAUMacAddressControl();echo
date('h:i:s A');}
elseif(isset($_POST['CPeMacAddressControl'])){echo date('h:i:s A');
CPeMacAddressControl($assignedzone); echo date('h:i:s A');}
elseif(isset($_POST['RogueClientMacControl'])){echo date('h:i:s A');
RogueClientMacControl($assignedzone); echo date('h:i:s A');}
elseif(isset($_POST['RFJamming'])){echo date('h:i:s A');RFJamming();echo date('h:i:s ');}
elseif(isset($_POST['CAUSiteSurvey'])){SiteSurveyCAU($assignedzone)}
elseif(isset($_POST['CPESiteSurvey'])){CPESiteSurveys()}
elseif(isset($_POST['LogTheTraffic'])){ echo date('h:i:s A');
LogTraffic($_POST['ZoneLogTraffic'], $_POST['RadioLogTraffic']);echo date('h:i:s A');}
elseif(isset($_POST['MonkeyintheMiddle'])){echo date('h:i:s A');EvilTwin();echo date('h:i:s
A');}else{
print "<BR><BR><form name='GiacForm' method='POST' action='$_SERVER[PHP_SELF]'>";
print "<table align='center' border='1px solid #BDCBD' border-collapse='collapse'
bgcolor='#0044FF'>";
print "<tr class='bigsort'><td colspan='3'><center><font color='red' size=6>Wireless
Rogue Device Detection</font></center></td></tr>";
print "<tr><td><b>CAU Mac Address Control</b></td><td colspan='2'><font
color='blue'><center><input type='submit' name='CAUMacAddressControl' value='CAU Mac
Address Control' size='15'></center></font></td></tr>";
print "<tr><td><b>CPE Mac Address List Control</b></td><td colspan='2'><font
color='blue'><center><input type='submit' name='CPeMacAddressControl' value='CPE Mac
Address List Control' size='15'></center></font></td></tr>";
print "<tr><td><b>Rogue Client Mac Control</b></td><td colspan='2'><font
color='blue'><center><input type='submit' name='RogueClientMacControl' value='Rogue
Client Mac Control' size='15'></center></font></td></tr>";
print "<tr><td><b>Evil Twin</b></td><td colspan='2'><font color='blue'><center><input
type='submit' name='EvilTwin' value='Evil Twin' size='15'></center></font></td></tr>";
print "<tr><td><b>Evil Twin based on Log Date</b></td><td colspan='2'><font
color='blue'><center><input type='submit' name='EvilTwinLogDate' value='EvilTwin Log
Date' size='15'></center></font></td></tr>";
print "<tr><td><b>Monkey in the Middle</b></td><td colspan='2'><font
color='blue'><center><input type='submit' name='MonkeyintheMiddle' value='Monkey in the
Middle' size='15'></center></font></td></tr>";
print "<tr><td><b>RF Jamming</b></td><td colspan='2'><font color='blue'><center><input
type='submit' name='RFJamming' value='RF Jamming' size='15'></center></font></td></tr>";
print "<tr><td><b>Log The Traffic</b></td><td><select name='ZoneLogTraffic'>";
print "<option value='0'>-----";
$sql = "SELECT * FROM somezones WHERE zone like '84%' OR zone like '32%' OR zone like
'33%' OR zone like '34%' ORDER BY zone";
$r = mcq($sql,$db);
while($note = mysql_fetch_array($r))
if (strlen($note[zone])>0)
print "<option value='$note[zone]','$note[zone]'";
print "</select>";
```

Detecting and Preventing Rogue Devices on the Network

```
print "<select name='RadioLogTraffic'>";
    print "<option value=1>1";
    print "<option value=2>2";
    print "<option value=3>3";
print "</select></td>";
print "<td><font color='blue'><center><input type='submit' name='LogTheTraffic'
value='Log The Traffic' size='15'></center></font></td></tr>";
print "<tr><td><b>CPE Site Survey</b></td><td><select name='Zone'>";
print "<option value='0'>-----";
$sql = "SELECT * FROM somezones WHERE zone like '84%' OR zone like '32%' OR zone like
'33%' OR zone like '34%' ORDER BY zone";
$r = mcq($sql,$db);
while($note = mysql_fetch_array($r))
    if (strlen($note[zone])>0)
        print "<option value='$note[zone]'>$note[zone]";
print "</select></td>";
print "<td><font color='blue'><center><input type='submit' name='CPESiteSurvey'
value='CPE Site Survey' size='15'></center></font></td></tr>";

print "<tr><td><b>CAU Site Survey</b></td><td>";
print "<select name='Zone'>";
    print "<option value='0'>-----";
    $sql = "SELECT * FROM somezones WHERE zone like '84%' OR zone like '32%' OR
zone like '33%' OR zone like '34%' ORDER BY zone";
    $r = mcq($sql,$db);
    while($note = mysql_fetch_array($r))
        if (strlen($note[zone])>0)
            print "<option value='$note[zone]'>$note[zone]";
print "</select></td><td><font color='blue'><center><input type='submit'
name='CAUSiteSurvey' value='CAU Site Survey' size='15'></center></font></td></tr>";
print "<tr><td><b>Insert Evil</b></td><td>print "<select name='ZoneInsertEvil'>";
print "<option value='0'>-----";
    $sql = "SELECT * FROM somezones WHERE zone like '84%' OR zone like '32%' OR
zone like '33%' OR zone like '34%' ORDER BY zone";
    $r = mcq($sql,$db);
    while($note = mysql_fetch_array($r))
        if (strlen($note[zone])>0)
            print "<option value='$note[zone]'>$note[zone]";
print "</select>";

print "<select name='RadioInsertEvil'>";
    print "<option value=1>1";
    print "<option value=2>2";
    print "<option value=3>3";
print "</select></td>";
print "<td><font color='blue'><center><input type='submit' name='InsertEvil'
value='Insert Evil' size='15'></center></font></td></tr></table></form>";
}
```

Log Traffic

```
function LogTraffic($zoneid, $radioid){

    if ($radioid == 1) $port = 23;
    if ($radioid == 2) $port = 24;
    if ($radioid == 3) $port = 22;

    $result = mysql_fetch_assoc(mysql_query("SELECT NasIPAddress as CauIpAddress FROM
RadiusNas WHERE (zone = $zoneid)"));
    $CauIpAddress = $result[CauIpAddress];

    $conn = ssh2_connect($CauIpAddress,$port);
```

Detecting and Preventing Rogue Devices on the Network

```
if (ssh2_auth_password($conn, 'UserName', 'Password'))
    echo "<br><center>Connected to the CAU at $zoneid.</center><br>";
else
    die('<br><center>Authentication is Failed...</center><br>');

$iwconfigmonitorcommand = '/sbin/iwconfig wlan0 mode monitor;';
$Now = date("Y-m-d-H-i-s");
$dumppcommand = 'DUMP' . '_' . $zoneid . '_' . $radioid . '_' . $Now;
$tcpdumpcommand = '/usr/sbin/tcpdump -i wlan0 -c 3000 -w ' . $dumppcommand . ' ';
$sleepcommand = '/bin/sleep 2;';
$iwconfigmastercommand = '/sbin/iwconfig wlan0 mode master;';
$scppcommand = '/usr/bin/scp ' . $dumppcommand . ' cpestatus@XX.XX.XXX.XX:.';

$allcommands = $iwconfigmonitorcommand . $tcpdumpcommand . $sleepcommand .
    $iwconfigmastercommand . $sleepcommand . $scppcommand ;

$stream = ssh2_exec($conn, $allcommands);
sleep(20);
}
```

RF Jamming

```
function RFJamming(){
    $Now = date("Y-m-d H:i:s");
    $FourHoursAgo = date("Y-m-d H:i:s",strtotime("$Now -4 hour"));

    $result = mysql_query("SELECT CPEssid, UserName, CAUNoise FROM logCPE WHERE
    LogDate>'$FourHoursAgo' AND NoiseLevel<>0 AND CAUNoise<>0 AND CPEssid<>' ' GROUP BY
    CPEssid ORDER BY CAUNoise DESC limit 20");

    print "<table align= 'center' border= '1px solid #BDBABD' border-collapse='collapse'
    bgcolor=#F7F7F7><tr><td align='center' colspan=3><font color='red'><b>Noisy AP
    List</b></font></td></tr><tr><td><b>Access Point</b></td><td><b>UserName</b></td>
    <td><b>CAUNoise</b></td></tr>";
    while($note = mysql_fetch_assoc($result))
        print "<tr><td><b>$note[CPEssid]</b></td>
        <td><b>$note[UserName]</b></td><td><b>$note[CAUNoise]</b></td></tr>";

    print "</table><BR><BR>";

    $result = mysql_query("SELECT MACAddress, UserName, NoiseLevel FROM logCPE WHERE
    LogDate>'$FourHoursAgo' AND NoiseLevel<>0 AND CAUNoise<>0 GROUP BY MACAddress ORDER BY
    NoiseLevel DESC limit 20");

    print "<table align= 'center' border= '1px solid #BDBABD' border-collapse='collapse'
    bgcolor=#F7F7F7><tr><td align='center' colspan=3><font color='red'><b>Noisy Client
    List</b></font></td></tr>";

    print "<tr><td><b>MAC Address of the Client</b></td><td><b>UserName</b></td>
    <td><b>NoiseLevel</b></td></tr>";
    while($note = mysql_fetch_assoc($result))
        print "<tr><td><b>$note[MACAddress]</b></td><td><b>$note[UserName]</b></td>
        <td><b>$note[NoiseLevel]</b></td></tr>";
    print "</table>";
}
}
```

Client Site Survey

```
error_reporting(E_ERROR);
```

Detecting and Preventing Rogue Devices on the Network

```
$Connectedarray = ConnecttoZone($AssignedZoneid, $db);
$conn           = $Connectedarray[0] ;
$Connected      = $Connectedarray[1] ;
if ($Connected == true)
    SiteSurveyCPE($conn, $zoneid);

function ConnecttoZone($zoneid, $db){
    $sql          = "SELECT NasIPAddress FROM RadiusNas WHERE (zone = '$zoneid')";
    $result       = mysql_query($sql,$db);
    if (!$result) die('Connection Problem:' . mysql_error());
    $result       = mysql_fetch_assoc(mysql_query($sql));
    $resultIPAddress = $result[NasIPAddress];
    $conn         = ssh2_connect($resultIPAddress,22);
    $Connected    = false;

    if (ssh2_auth_password($conn,'UserName','Password')){
        echo "<br><center>Connected to the CAU.</center><br>";
        $Connected = true;
    }else{
        die('<br><center>Authentication is Failed...</center><br>');
        $Connected = false;
    }
    return array($conn, $Connected);
}

function SiteSurveyCPE($conn, $zoneid){
    $ListeningIP = "XX.XX.XXX.XX";
    $result       = mysql_query("SELECT id, CPE_IP, zone, CPE_MAC, user FROM Clients WHERE
    (zone = $zoneid) AND STATUS='enabled' ");

    while($note = mysql_fetch_assoc($result)){
        $CpeIpAddress = $note[CPE_IP];
        $MACAddress    = $note[CPE_MAC];
        $CpeZoneId     = $note[zone];
        $CustomerId    = $note[id];
        $notification  = WaitOnServer($ListeningIP, '1', $CpeIpAddress, $MACAddress,
        $conn, '1');
        if ($notification == 1)
            $filename = LogSiteSurvey($CpeIpAddress, $CpeZoneId, $conn, $CustomerId);
    }
}

function make_seed() {
    list($usec, $sec) = explode(' ', microtime());
    return (float) $sec + ((float) $usec * 100000);
}

Function WaitOnServer($ListeningIP, $ListeningType, $CpeIpAddress, $MACAddress, $conn,
$Channel){
    set_time_limit(40);
    $socket = socket_create(AF_INET, SOCK_STREAM, 0) or die("Could not create socket\n");
    srand(make_seed());
    $ListeningPort = rand();
    $ListeningPort = ($ListeningPort % 25)+5501;
    while (socket_bind($socket, $ListeningIP, $ListeningPort) != 1) {
        $ListeningPort = rand();
        $ListeningPort = ($ListeningPort % 25)+5501;
        socket_shutdown($socket, 2);
        socket_close($socket);
        $socket = socket_create(AF_INET, SOCK_STREAM, 0) or die("Could not create
        socket\n");
    }

    $snrcommand = '/bin/bash -C ' . '\'/mnt/mon/sitesurvey.sh\' \' ' .
    $CpeIpAddress . '\ ' -p\ ' \' . $ListeningPort . '\ ' \' . $MACAddress . '\ ' ' .
    '\ '&\';
}
```

Detecting and Preventing Rogue Devices on the Network

```
$stream          = ssh2_exec($conn, $snrcommand);
$result          = socket_listen($socket, 1) or die("Could not set up socket
listener\n");
socket_set_option($socket,SOL_SOCKET, SO_RCVTIMEO, array("sec"=>30,
"usec"=>100));
if ($spawn      = socket_accept($socket)){
    $notification = socket_read($spawn,8) or die("Could not read input\n");
    if ($notification == 1)
        print " <center> CPE Status run successfully!!!</center><BR>";
    return($notification);
    socket_shutdown($socket, 2);
    socket_close($socket);
}
}

function LogSiteSurvey ($CpeIpAddress, $CpeZoneId, $conn, $CustomerId){
    $list          = split("\.", $CpeIpAddress);
    $XmlCpeIpAddress= $list[2] . $list[3];
    $snrstring     = 'scp /var/spool/Client' . $CpeZoneId . '_' . $XmlCpeIpAddress .
    '.xml cpestatus@XX.XX.XXX.XX:.';
    $stream        = ssh2_exec($conn, $snrstring);
    $filename      = '/home/cpestatus/cpe' . $CpeZoneId . '_' . $XmlCpeIpAddress .
    '.xml';

    while (!(file_exists($filename)))
        usleep(2000);
    XMLParserSiteSurvey($CpeZoneId, $filename, $CustomerId);
    return ($filename);
}

function XMLParserSiteSurvey($CpeZoneId, $filename, $CustomerId){

    global $AllData, $AddData, $CustId, $CpeZone;

    $CustId        = $CustomerId;
    $CpeZone       = $CpeZoneId;

    $parser        = xml_parser_create();
    $AllData       = "<table align='center'>";
    $AllData       = $AllData . '<tr><td align="center" colspan="8"><b>CPE
INFORMATION</b></td></tr>';
    $AddData       = false;

    xml_set_element_handler($parser,"startsurvey","stopsurvey");
    xml_set_character_data_handler($parser,"processingsurvey");

    sleep(1);

    $fp=fopen($filename,"r");

    while ($data=fread($fp,4096)){
        xml_parse($parser,$data,feof($fp)) or die (sprintf("XML Error: %s at line %d",
xml_error_string(xml_get_error_code($parser)),
xml_get_current_line_number($parser));
    }

    $AllData = $AllData . '</tr></table>'; print "$AllData <BR>";
    xml_parser_free($parser);
}
}
```

Detecting and Preventing Rogue Devices on the Network

AP Site Survey

```
SiteSurveyCAU($AssignedZoneId);

function SiteSurveyCAU($zoneid){

    $sql          = "SELECT NasIPAddress FROM RadiusNas WHERE (zone = $zoneid)";
    $result       = mysql_fetch_array(mcq($sql,$db));

    $conn        = ssh2_connect($result[NasIPAddress],22);

    if (ssh2_auth_password($conn,'UserName','Password'))
        echo "<br><center>Connected to the Zone: $zoneid </center><BR>";
    else
        die('<br><center>Authentication is Failed...</center><br>');

    $iwconfigmanaged = '/sbin/iwconfig wlan0 mode managed;';
    $sleepcommand    = '/bin/sleep 15;';
    $iwlist           = '/sbin/iwlist wlan0 scan;';
    $iwconfigmaster  = '/sbin/iwconfig wlan0 mode master;';
    $catcommand      = 'cat /proc/net/hostap/wlan0/scan_results;';

    $allcommands    = $iwconfigmanaged . $sleepcommand . $iwlist . $iwconfigmaster .
    $catcommand ;

    $stream         = ssh2_exec($conn, $allcommands);

    sleep(20);

    $Dump = ''; $passed = false; $TotalLines = '';

    while($line = fgets($stream)){

        $findme      = 'CHID ANL SL BcnInt Capab Rate BSSID ATIM SupRates SSID';
        $pos         = strpos($line, $findme);

        if (!$pos == false)
            $passed = true;
        if ($passed == true)
            ParseLine($line, $zoneid);
    }

    print "</table>";
}

function ParseLine($line, $zoneid){

    $findme        = ' ';
    $pos           = strpos($line, $findme);
    $channel       = substr($line, 0, $pos);

    if (is_numeric($channel)){
        print      "<tr><td>$channel</td>";
        $newline   = substr($line, $pos+1);

        $pos       = strpos($newline, $findme);
        $noise     = substr($newline, 0, $pos);
        $newline   = substr($newline, $pos+1);

        $pos       = strpos($newline, $findme);
        $signal    = substr($newline, 0, $pos);
        $SNR       = $signal - $noise;
    }
}
```

Detecting and Preventing Rogue Devices on the Network

```
print          "<td>$SNR</td>";
$newline      = substr($newline, $pos+1);

$pos          = strpos($newline, $findme);
$BcnInt      = substr($newline, 0, $pos);
print        "<td>$BcnInt</td>";
$newline     = substr($newline, $pos+1);

$pos          = strpos($newline, $findme);
$Capab       = substr($newline, 0, $pos);
print        "<td>$Capab</td>";
$newline     = substr($newline, $pos+1);

$pos          = strpos($newline, $findme);
$Rate        = substr($newline, 0, $pos);
print        "<td>$Rate</td>";
$newline     = substr($newline, $pos+1);

$pos          = strpos($newline, $findme);
$mac         = substr($newline, 0, $pos);
print        "<td>$mac</td>";
$newline     = substr($newline, $pos+1);

$pos          = strpos($newline, $findme);
$atim        = substr($newline, 0, $pos);
print        "<td>$atim</td>";
$newline     = substr($newline, $pos+1);

$pos          = strpos($newline, $findme);
$suprates    = substr($newline, 0, $pos);
print        "<td>$suprates</td>";
$ssid        = substr($newline, $pos+1);

print        "<td>$ssid</td></tr>";

$LogDate     = date("Y-m-d H:i:s");

$sql         = "INSERT INTO causitesurveys ";
$sql         = $sql . "(channel, snr, capab, rate, mac, suprates, zoneid, ssid,
LogDate) ";
$sql         = $sql . "VALUES ($channel, $SNR, '$Capab', $Rate, '$mac',
'$suprates', $zoneid, '$ssid', '$LogDate')";

$result      = mysql_query($sql);

}else{
print "<br><table align='center' class='querytable'>";
print "<tr>";
print "<td>Channel</td>";
print "<td>SNR</td>";
print "<td>BcnInt</td>";
print "<td>Capab</td>";
print "<td>Rate</td>";
print "<td>Mac</td>";
print "<td>ATIM</td>";
print "<td>SupRates</td>";
print "<td>SSID</td>";
print "</tr>";
}
}
```

Detecting and Preventing Rogue Devices on the Network

Man in the Middle Attack

I decided to detect Evil Twin attack and coded it everywhere like that but all I detected was Man-in-the-middle attack. Therefore I changed the subject from Evil Twin to Man-in-the-middle attack.

```
function EvilTwin(){

    $result          = mysql_query("select * from cpesitesurveys where site_ssid like '%BBC%'
    OR site_ssid like '%TBZ%' group by mac order by site_ssid");

    $previous_ssid = 'none';
    $previousmac   = 'none';

    print "<table align= 'center' border= '1px solid #BDBABD' border-collapse='collapse'
    bgcolor='#F7F7F7'>";
    print "<tr><td align='center' colspan=8><b>Different Mac Address Same
    SSID</b></td></tr>";

    while($note = mysql_fetch_assoc($result)){
        if ($previous_ssid == $note[site_ssid]){
            print "<tr>";
            print "<td><b>SSID</b></td>";
            print "<td><b>$previous_ssid</b></td>";
            print "<td><font color='blue'>Mac Address</font></td>";
            print "<td><b>$previousmac</b></td>";
            print "<td><font color='blue'>Log Date</font></td>";
            print "<td><b>$previousLogDate</b></td>";
            print "<td><font color='blue'>Signal</font></td>";
            print "<td><b>$previoussignal</b></td>";
            print "</tr>";

            print "<tr>";
            print "<td><b>SSID</b></td>";
            print "<td><b>$note[site_ssid]</b></td>";
            print "<td><font color='red'>Mac Address</font></td>";
            print "<td><b>$note[mac]</b></td>";
            print "<td><font color='red'>Log Date</font></td>";
            print "<td><b>$note[LogDate]</b></td>";
            print "<td><font color='red'>Signal</font></td>";
            print "<td><b>$note[signal]</b></td>";
            print "</tr>";
        }else {
            $previous_ssid          = $note[site_ssid];
            $previoussignal         = $note[signal];
            $previousmac            = $note[mac];
            $previousLogDate        = $note[LogDate];
        }
    }
    print "</table>";
}

function InsertEvilTwin($ZoneId, $Radio){

    $Now          = date("Y-m-d H:i:s");
    $SSID         = 'BBC_' . $ZoneId . '_' . $Radio;
    $tempmac      = 'AA:AA:AA:AA:AA' . ':' . $Radio . $Radio;
}
```

Detecting and Preventing Rogue Devices on the Network

```
$SQL          = "INSERT INTO cpesitesurveys (zoneid, customerid, site_ssid, mac,
channel, LogDate, signal) VALUES ($ZoneId, 99999, '$SSID', '$tempmac', 1, '$Now', -99)";
mysql_query($SQL) OR die(mysql_error());
print"<BR><center><b>Evil Twin Inserted into zone : $ZoneId, radio :
$Radio!</b></center><BR>";
}
```

Evil Twin

This code is similar to the Man-in-the-middle attack but checks log dates and MAC addresses. This code catches evil twin attack if there is really one. I inserted data to the DB and saw that it is working, but couldn't be able to change the MAC Address of my USB Wireless Adapter's MAC address and couldn't be able to test it on the field.

```
function EvilTwinLogDate(){

$result      = mysql_query("SELECT site_ssid FROM cpesitesurveys WHERE site_ssid like
'%BBC_8%' OR site_ssid like '%TBZ_8%' group by mac order by site_ssid");

while($note = mysql_fetch_assoc($result)){
    $AnHourAgo    = date("Y-m-d H:i:s",strtotime("$Now -1 hour"));
    $resultinner  = mysql_query("SELECT * FROM cpesitesurveys WHERE
LogDate>'$AnHourAgo' AND site_ssid='$note[site_ssid]' order by customerid,
LogDate ASC ");
    print "<table align= 'center' border= '1px solid #BDBABD' border
collapse='collapse' bgcolor='#F7F7F7'>";

    print "<tr>";
        print "<td><b>Customer Id</b></td>";
        print "<td><b>Site SSID</b></td>";
        print "<td><b>Channel</b></td>";
        print "<td><b>Log Date</b></td>";
        print "<td><b>Signal</b></td>";
        print "<td><b>Attack Type</b></td>";
    print "</tr>";

    $previouscustomerid    = '0';
    $previousLogDate       = '0000-00-00 00:00:00';
    $previouschannel       = '0';
    $previousmac           = '00:00:00:00:00:00';
    $previoussignal        = '0';

    while($noteinner = mysql_fetch_assoc($resultinner)){
        if ($previouscustomerid == $noteinner[customerid]){
            if ($previousLogDate == $noteinner[LogDate]){

                if ($previouschannel == $noteinner[channel])
                    $AttackType = 'Evil Twin' ;
                else
                    $AttackType = 'Man in The Middle' ;
            }
        }
    }
}
```

