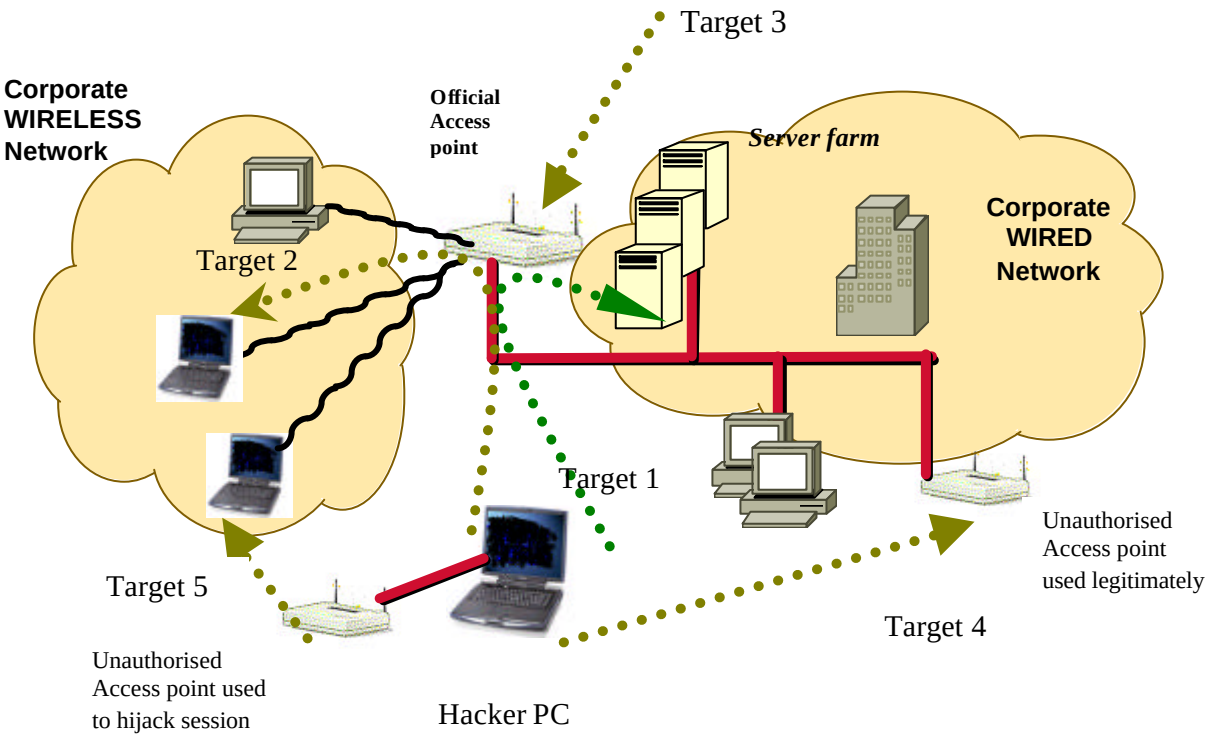
WIDZ - The Wireless Intrusion detection system

The design and implementation of Version TWO

1 Introduction

WIDZ is still a proof of concept – it does work and it will run quite happily in a stable environment but I have really developed this for demonstration purposes. Although it was designed with features that would enable corporate deployment, it hasn't be coded to an exacting standard nor tested with sufficient rigour.

1.1 The wireless network –what is attacked



The diagram above shows the various elements of the wireless network and how these can be attacked. When I first produced this diagram, many practioners were only talking

www.loud-fat-bloke.co.uk

about wireless attacks being pointed at the AP. For those of us that know the guy that caught slammer over an 802.11 link, we can feel justifiably smug.

- The corporate network and servers

This is what most people generally recognised as a 802.11 hacking target. An attempted penetration through the official access points(target 1) into the corporate network. Very simple and very basic.

At this location, only 802.11 scans and DOS attacks should remain un-detected by a good traditional infrastructure because inherently most other attacks will be TCP/IP based. If any more gets by, there is a more general network security problem.

- The wireless clients

The wireless clients(Target 2) is generally not recognised as a target. Here the Access point behaves as a hub connecting the authorised wireless clients directly to the bad buys – inevitably this will expose a connecting pc to a huge array of IP based attack.

- The unauthorised Access point

Unofficial access points installed by user departments (target 4) represent a huge risk as the security configuration is often questionable, providing an effective yet unmonitored back-door to the network.

Bogus Access points (Target 5) represent a different threat as these can be used to hijack sessions at the data link layer and steal valuable information.

- Target 3 – The legitimate Access point

Services like SNMP and web-based configuration tools on the Access point are often targeted by attackers. Again protecting the internal interface is a typical network security challenge – the same as any router or switch.

I decided that Target 3 could be adequately provided for by normal wired security mechanisms. I also decided that target 1 and target 2 functions could be combined into one module so we came up with the WIDZ two module design.

2 WIDZ two module design

WIDZ was therefore designed with two modules, the AP monitor and the Probe (or attack monitor). This are deployed separately to match the particular environment.

2.1 - Unauthorised AP monitor (widz_apmon.c)

This module covers two types of attack:-

- Bogus APS which are designed to steal the associations . Once this is achieved, login credentials can be retrieved or a man in the middle attacks can be performed. This is not a theoretical attack as some writers have claimed (please refer to the Monkey-jack attack).
- Unauthorised APs are the ones that are installed by, say, the marketing department after they have visited the local PC superstore. They usually allow all and sundry access to the corporate LAN without a password.

Addressing this need was simple - any reasonable administrator in a reasonably sized organisation could identify an official Access point and record their details in a file. Even a poor one could examine an automatically generated file and verify that its content are correct. These details would include Mac Address, channel and ESSID. All this module had to do was to do an AP scan and compare the results with our file – any that were found and did not appear are either bogus or unauthorised. Consideration was given to detecting changes in signal strength and frequency but this was considered over-kill at the time.

2.2 802.11b Traffic monitor(widz_probemon)

A file containing Mac Addresses and ESSIDs is obviously not going to provide a real world solution to drive this modules detection capabilities. This is because:

- Any self-respecting hacker or war-driver knows how to change a MAC-Address – making the identification process flimsy at best.
- Whilst many organisation would have less than say 20-30 access-points, the same organisation may have hundreds or thousands of network cards some of which will need to be replaced as they become faulty – this would create an administrative nightmare if each were to be record in a file manually.

None-the-less, it has been suggested that such facilities are essential so MAC blacklist and white-lists have been added, along with an ESSID blacklist to reduce noise. Other methods of detection include:

- **probe monitoring** - Picks up probe requests which don't have the ESSId field set in the probe or that have an essid that matches the ESSId blacklist.
- **Flood detection** - Picks up attempts to flood the network. The floods detected include
  - ?? Auth fail floods
  - ?? Asso floods

- ?? Deauth floods
- ?? General volume floods
- ?? Fatajack

?? **General failures** – most AP's do not provide any error recording, widz detects non zero return code on authentication, association and disassociation and reports on them

?? **Wireless Client attack** – (picking up of wireless attacks on associated wireless client) although considered was never implemented

2.3 Alert management

Each new security product seems to need its own management console – that cannot be right! For both these modules it was planned to report in a way that could be processed by Snort, swatch or any SNMP manager.

3 Implementation

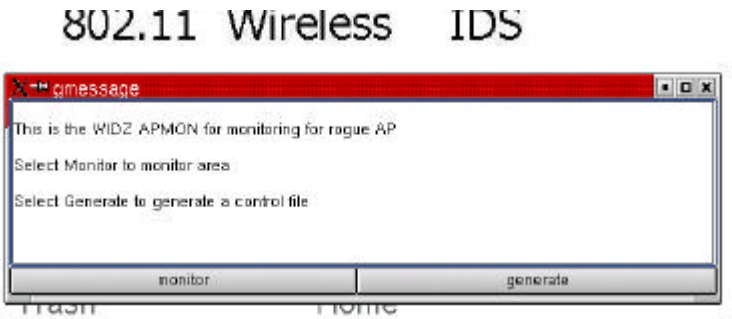
3.1 Widz_apmon

This little program monitors an area for Access Points. If it finds an access point it compares it to a list of Authorised APs in a configuration file called *widz_apmon.conf*. If the detected AP isn't listed in the configuration file, the program raises an Alert with an appropriate message.

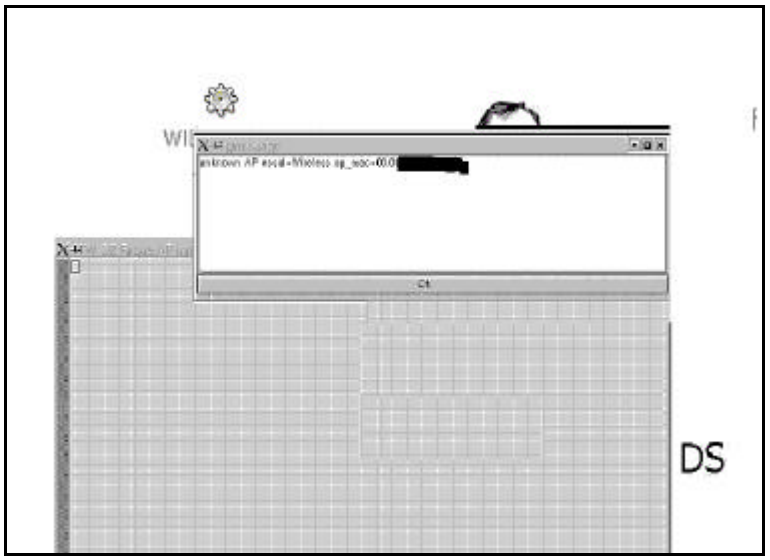
The original command line interface is still available but now simple scripts have been written to easy the process (I wrote them for InfoSec so that people could come to the stand and play)

1. From the WIDZ desktop, click WIDZ_APMON

2. This produce a dialog box that will allow you to select monitor – (It did have fully functioning generate code included but various Muppets kept on selecting it by mistake and eradicating the config files so it was removed).



3. Click monitor and the probemon monitor will start (as below – sorry for the poor graphic, I edited out an address and it lost quality)



(mac addresses have been blanked to protect the identity and my wallet)

3.1.1 WIDZ_APMON.CONF

By default the control file to the AP monitor is called *widz_apmon.conf*. This file can be generated automatically or created manually. An example file is shown below

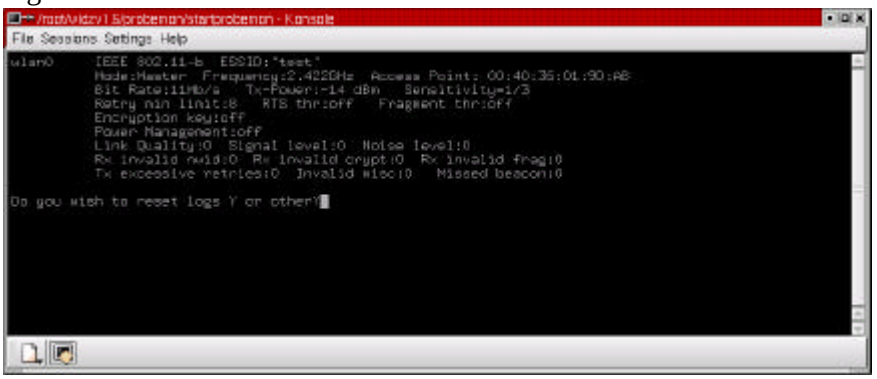
```
essid=tesz ap_mac=00:06:25:53:e0:8e
essid=Wirelesz ap_mac=00:30:ab:1b:9b:cc
```

3.2 widz_probemon.c

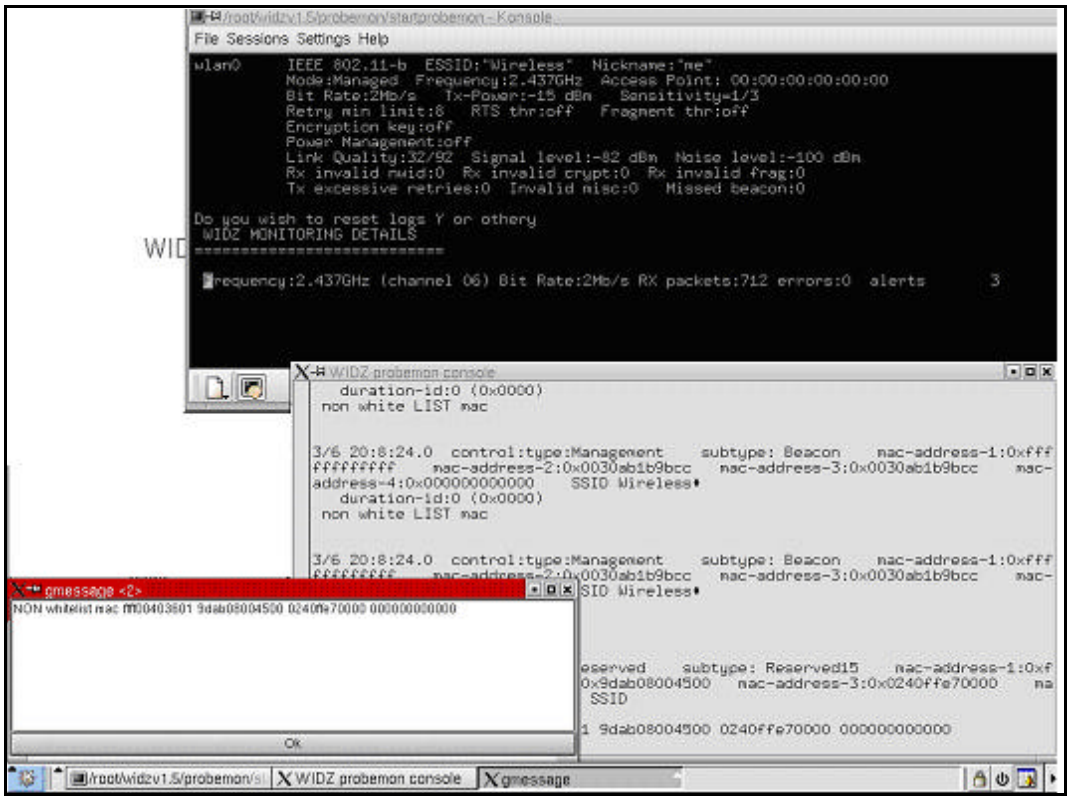
This program originally was a traditional UNIX filter that read the output of a program like *prismdump*, *tethereal* or our custom version. Now it reads the info directly from a raw socket.

1. To start probemon click on the ICON

2. You will see some interface information and have to respond “Y” to clear the logs



3. Subsequently, a console will be displayed showing any alert info



3.2.1 Tailoring widz_probemon

Like ap_mon, widz_probemon.c has a simple text control file called *probemon.conf*. An example is shown below:

```
# probemon.conf
# takes the following parms
#usebadmacs=y|n
#usebadssids=y|n
#usescripts=y|n
usebadmacs=n
usebadssids=n
usegoodmacs=y
```

It contains the following Stanzas each that take the value y or n:

- usebadmacs If set to n, this feature is NOT activated.

  If set to y, the probe monitor will report and call the alert subroutine for any packet that contains a mac address defined as a “bad mac” in any of the (up to) four address fields in an 802.11 packet.

  Bad macs are defined in a text file named in the constant (i.e. #define) BADMACFILE. Typically, it has the value "./badmacs"

- usebadssids If set to n, this feature is NOT activated.

  If set to y, the probe monitor will report and call the alert subroutine for any probe or probe-response packet that contains a SID defined as a “bad sid”.

  Bad SIDS are defined in a text file named in the constant (i.e. #define) BADSSIDFILE. Typically, it has the value "./badsids".

- usegoodmacs If set to n, this feature is NOT activated.

  If set to y, the probe monitor will IGNORE any packet that contains a mac address defined as a “good mac” in any of the (up to) four address fields in an 802.11 packet.

  Bad macs are defined in a text file named in the constant (i.e. #define) GOODMACFILE. Typically, it has the value "./goodmacs".

- usescripts If set to n, this feature is NOT activated.

  If set to y, the probe monitor will call a number of custom script for every record intercepted. The scripts must be stored in a directory defined by the expression stored in the constant WIDZSCRIPTS. An example is provide in the WIDZ 1.6 distribution.

3.3 Alert

A program named Alert will be executed each time an Alert is raised. The WiDZ package provides an example script which shows how to

- send a syslog message
- write to the console or current terminal
- send an snmp trap
- send an email

4 WIDZ IMPROVEMENTS

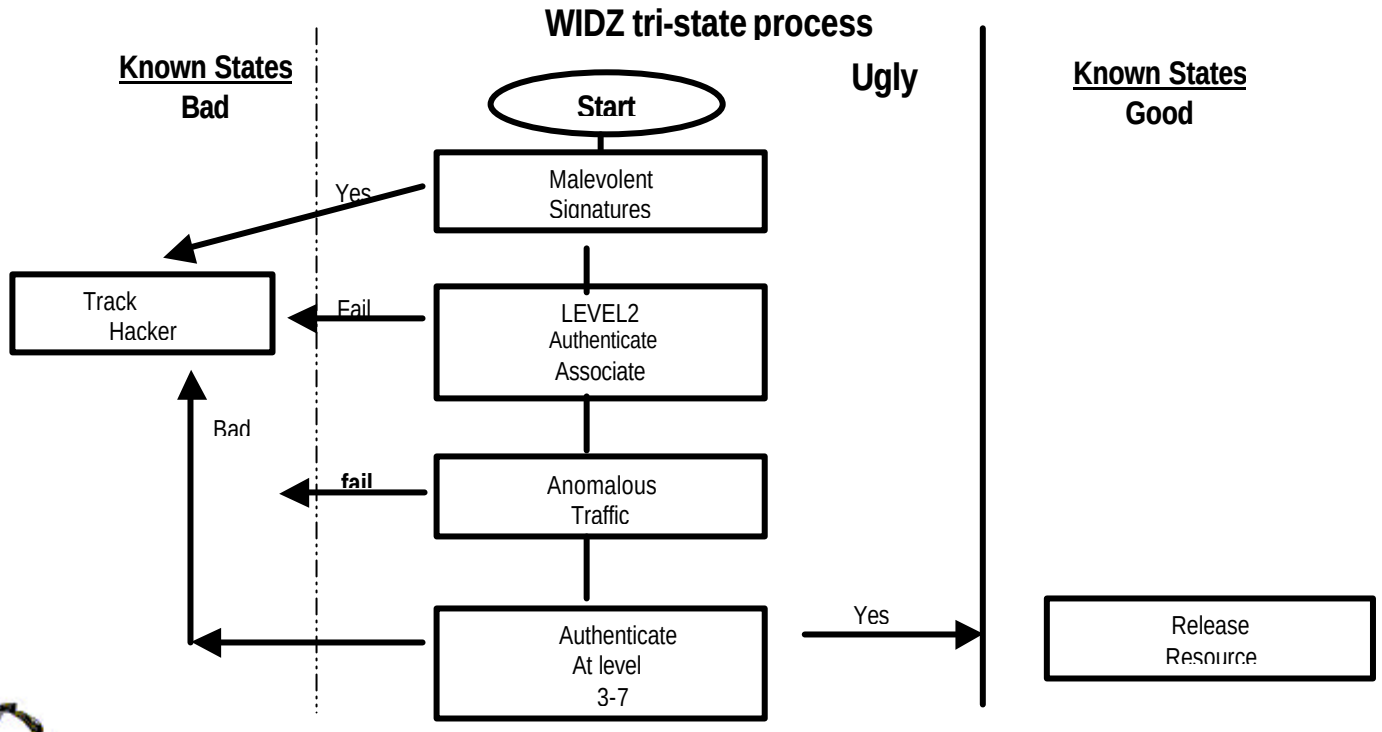
Apart from better coding and testing, the following improvements are considered

Good Traffic Authenticate Model

This model is shown below by the state diagram. All traffic has three states represented as stacks of wireless clients.
Suspect (or ugly) – All clients enter the system at this stage, they will remain in this state until they do something that classifies them as bad, or they authenticate on a corporate server using traditional authentication which will make them good, or until they have been suspect for a defined period. The assumption is that after a certain period of time they remain unauthenticated, they must be alien/unable to authenticate and therefore bad.

Good – This client and its traffic been authenticated at by a traditional means and should be considered as good. This can be established by a query of the existing servers.

Bad – Traffic considered alien or malevolent.



www.loud-fat-bloke.co.uk

www.loud-fat-bloke.co.uk