ISSN 2229-5518

Design of a Honeypot Based Wireless Network Architecture and its Controlled Penetration Testing

Achin Kulshrestha, Prof. Anjali Sardana

Abstract— The convenience of 802.11-based wireless access networks has led to widespread deployment in the consumer, industrial and military sectors. Due to the borderless nature of 802.11, security is an obvious concern, mainly due to the physical aspects of the technology, and also because of weak encryption and authentication implementations. Wireless networks introduce a new point of entry into previously closed wired networks and must thus be treated as an untrusted source, just like the Internet. For wireless networking to be most useful, the wireless networks must pass data on to standard wired networks connected to the Internet which makes the wired networks vulnerable to attacks. A Honeypot acts as a supplemented active defense system for network security. Honeypots are closely monitored decoys that are employed in a network to study the trail of hackers and to alert network administrators of a possible intrusion. This paper explains honeypot architecture for the emerging mix of wired and wireless networking equipment and describes the use of this honeypot to capture various attacks by performing controlled penetration tests on the network testbed. The fundamentals will consist of an overview of 802.11b security, various attacks on the architecture and will conclude explaining ways of deception implemented in our honeypot architecture so as to make blackhats lose time in their discovery of the legitimate network.

Index Terms—Architecture, Honeypots, Networking, Penetration Testing, Security, Testbed, Wireless.

1 Introduction

Wireless technologies drive our world and have become a defacto standard for communication, entertainment and education across the planet. With wireless technologies opening avenues of change due to their ubiquity, remote capabilities, and ability to close information gaps, the human dependency on these technologies has increased to the point where one can find wireless devices almost everywhere. End users and enterprises are heavily dependent on wireless technologies because of the flexibility, mobility and freedom it provides to access and share information.

Along with this flexibiliy, though, come security issues that must be comprehensively understood. Though contemporary wireless devices support standard security methods and protocols (encryption, authentication, authorization etc) useful to thwart common attacks, many kinds of attacks are still possible but are dependent on the real level of security present and the skill of the attacker. Due to the burgeoning usage of wireless equipment and technologies today, it is imperative to get knowledge at the grassroots about the real exploitation vectors currently used to compromise wireless networks. Trying to fill this knowledge gap, the main goal of Wireless honeypots is to analyze the state of real life wireless hacking and thereby make the networks more secure.

Wireless honeypots could help to reveal real stats about attacks on the infrastructure, such as the frequency of attacks, the blackhat's skill level, his objectives and techniques. Honeypots can also help with protecting the critical networks while the attacker spends substantial effort on bogus targets.

- *Achin Kulshrestha, BTech, Dept of Computer Engineering, National Institute of Technology Kurukshetra, Institute of National Importance, India, PH-+91-8147782340. E-mail: achinkul@gmail.com*
- *Dr. Anjali Sardana, Assistant Professor, Dept. Of Electronics and Communication, Indian Institute of technology Roorkee, India. E-mail: anjlsfesc@iitr.ernet.in*

2 Problem Description

To Design a Honeypot based network architecture for the emerging mix of wired and wireless networking equipment and simulating the attack capture process by performing controlled penetration tests on the network.

The main objective of this project is to design a honeypot which deals with the attacks launched by a blackhat from a wireless machine.

The problem can be subdivided into various steps as follows:
1. Simulation of various wireless attacks
2. Capturing the Attacks
3. Storing the packets in the database
4. Analysis

In real time scenarios, based on the activity detected on the wireless honeypot designed in this architecture, following can be inferred about the intent of the attacker:

- Wired honeypot not attacked (but association with the Wireless Access Point (AP) – a wardriver merely surveying the network (this could also be for planned future activities) [1].
- Attempt to gain Internet access– An attacker trying to get free internet access (Again this might be a first

step to a sophisticated attack).

- Installation of malicious binaries – the hacker is skilled and is looking to compromise a system for future hacking activities.
- Modification of core system files – Attacker is intenting to compromise and inflict damage to critical systems.

3 Test Bed Description

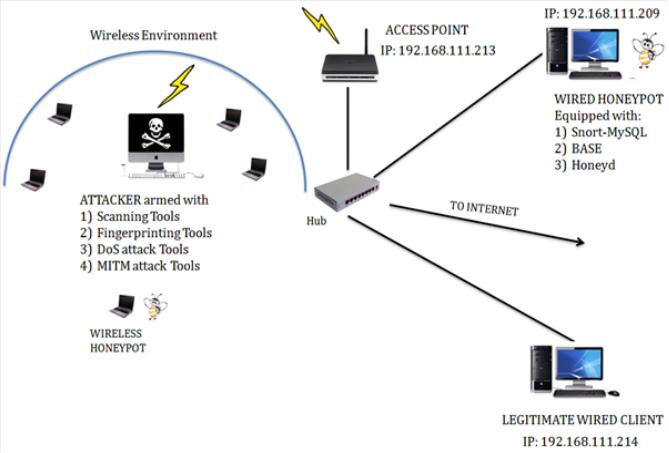

Fig.1 Test Bed

Two wired clients are connected to the access point with the help of a Hub, one of which is a honeypot and other is a legitimate wired client. The Entire configuration is as follows:

- Attacker: Dell laptop with 1395 running Win and Ubuntu (Dual-boot).
- Access point (192.168.111.213): D-Link DWL-2100, Air Plus XtremeG
- Hub: - Quantum QHM7300B-STP
- Wired Honeypot(192.168.111.209): Running Linux 2.6.31-22 generic and Windows XP(Dual-boot)
- Wired Client (192.168.111.214): Running Linux 2.6.31-22 generic
- Access point: Set up with a basic WEP 64 encryption having a 5 digit ASCII code and also with MAC address filtering.

Internet access was also provided to the network to get real time data. The wired honeypot consists of a low interaction honeypot (i.e presenting the adversary with emulators of venerable programs like ftp, Telnet and capture limited interaction). This helps in protecting the critical network assets while still gathering attack data for further analysis. There is also a facility for analysis using ACID [2] (Analysis Console for Incident databases). ACID is a PHP-based analysis engine to lookup and analyse a database of security incidents captured by Snort.

4 Components Background

4.1 Honeypots

Honeypots are an interesting piece of technology with tremendous uses in the security sphere. The honeypot concept was first brought to light by several icons in computer security, specifically Cliff Stoll in the book "The Cuckoo's Egg", and Bill Cheswick's paper "An Evening with Berferd." Since then, honeypots have continued to evolve, developing into the powerful security tools they are today.

Honeypots do not solve a specific problem, unlike other security related network components like IDS/IPS and firewalls. Due to their flexibily, they can do everything from detecting denial of service attacks to capturing an encrypted man in the middle attack. It is this versatility that has made honeypots so effective in thwarting even skilled and highly sophisticated attackers. It is this flexibility that can make honeypots a challenging entity to define and understand. As such, the following definition by Lance Splitzner defines's what a honeypot is:

A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource. [3]

All honeypots share the same concept: a security resource that should not have any production or authorized activity. In other words, deployment of honeypots in a network should not affect critical network services and applications.

A Honeypot can distract adversaries from more valuable machines on a network, they can provide early warning about new attack and exploitation trends and they allow in-depth examination of adversaries during and after exploitation of a honeypot.

4.2 Types of Honeypots

Honeypots can be of various types but at a high level they can be broken down into two general categories, low-interaction and high-interaction honeypots. According to Lance's definition of honeypot, interaction defines the level of activity a honeypot allows an attacker to perform.

Low-interaction

A low interaction honeypot emulates operating systems and services. In case of an attack they can record the time, communication protocols, source IP, source port, destination IP, destination port, and exploit type for each attack. Following features of these honeypots make them a useful tool against attackers:

- Low risk of a possible compromise, as the emulated services control what attackers can and cannot do.
- Very easy to deploy and install in a network.

High-interaction

High-interaction honeypots let the attacker interact with the system like any real operating system.

- They allow administrators to capture extensive details about the full extent of an attacker's method.
- They possess increased risk, as there is little to no restriction placed on what the hacker can do once he/she comprises the system.

A honeypot is a machine on which no legitimate activity has to be seen. This means that any traffic seen on a honeypot can be regarded as an attack or malicious activity. Since any connection to a honeypot is most likely a reconnaissance attempt by an attacker the small set of data they collect is of very high significance.

Honeypots have several clear-cut advantages:

- Honeypots protect the actual production servers from an attack through emulation of services. If a weakness in the network leads to a compromise, honeypots can help buying in sufficient time for the administrators to fix the loophole while the attackers are busy attacking the honeypot.
- Honeypots collect limited amounts of data pertaining only to them. Since any traffic directed towards them is considered malicious or unauthorized, even the small amount of data they collect is of high significance for network forensics.
- Honeypots can help to understand an attacker's tactics and methodologies. Since they are designed to capture anything thrown at them, they can help in finding tools or tactics never seen before.
- Honeypots have just one task, to capture everything directed towards them. This requires minimal resources.
- IDS/IPS systems require different configuration setting depending on the environment they are deployed in. However, honeypots work perfectly in encrypted or IPv6 environments. It does not matter what the honeypot receives, it will capture it.

4.3 Wireless LAN

A Wireless Local Area Network (WLAN) links two or more devices using some wireless distribution method (typically spread-spectrum or OFDM radio), and usually providing a connection through an access point to the wider internet. This gives users the mobility to move around within a local coverage area and still be connected to the network [4]. The IEEE 802.11 is a set of standards maintained by the IEEE LAN/MAN Standards Committee. The first version of IEEE 802.11 was released in 199, but is today obsolete. The base current version of the standard is IEEE 802.11-2007.

Following are the various versions of 802.11 released till date:

- 802.11-1997 (802.11 legacy)
- 802.11a OFDM Waveform
- 802.11b
- 802.11g
- 802.11-2007
- 802.11n
- 802.11-2012
- 802.11ac
- 802.11ad

4.4 Wireless Concepts

4.4.1 Stations and Access points

A wireless network interface card (adapter) is a device, called a station, and is used to connect radio-based computer networks. An access point (AP) is a station providing frame distribution service to stations associated with it. AP, also called a base station provides wireless access to a wired Ethernet network. It plugs into a hub, switch, or wired router and sends out wireless signals. The AP itself is usually connected by wire to a LAN.

The station and AP each contain a NIC that has a Media Access Control (MAC) address, just as wired network cards have. The MAC address is a 48-bit number, assigned to the device at the time of manufacture and is world-wide-unique. The 48-bit address is represented as a string of six octets separated by colons (e.g., 00:01:2B:19:C9:F8) or hyphens (e.g., 00-02-2A-27-C9-F8). While the MAC address as assigned by the manufacturer is printed on the device, the address can be changed in software.

Each AP also has a 0 to 32 byte long Service Set Identifier that is used for naming the wireless network[5]. The SSID is used to segment the airwaves for usage allowing each packet sent over the wireless network to arrive at the correct location. If two wireless networks are physically close, the SSIDs label the respective networks, without SSIDs, sending and receiving data in a location with multiple wireless networks would be chaotic. A separate SSID allows the components of one network to ignore those of the other.

4.4.2 Infrastructure and Adhoc Modes

A wireless network operates in one of the two defined modes for 802.11 networks, the ad hoc mode and the infrastructure mode. In the ad hoc mode, each station is a peer to the other stations and communicates directly with other stations within the network. No access point is involved and all stations can send Beacon and Probe frames. The ad hoc mode stations form an Independent Basic Service Set (IBSS).

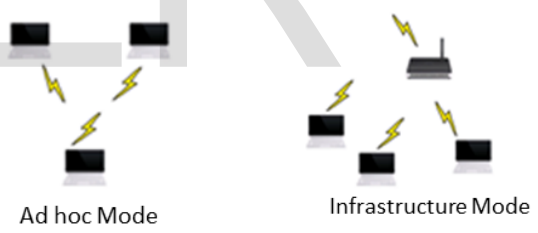

Fig. 2 Infrastructure and ad-hoc modes

The infrastructure mode of operation is slightly different. As shown in the above figure a station in the infrastructure mode communicates with an access point only. Analogous to the Ad-hoc mode IBSS structure, the infrastructure mode defines a Basic Service Set (BSS) forming a set of stations that are logically associated with each other and controlled by a single AP. Together they operate as a fully connected wireless network. Similar to a MAC address the BSSID is a 48-bit number which uniquely identifies each BSS[6]. Advantages of the Infrastructure networks include greater stability, better security and scalability than most Ad Hoc networks .

4.4.3 Frames

The format of 802.11 frames is illustrated below in figure 3. Most of the frames contain IP packets. The 802.11 MAC frame, consists of a MAC header, the frame body, and a frame check sequence (FCS). The numbers in the following figure represent the number of bytes for each field [6].

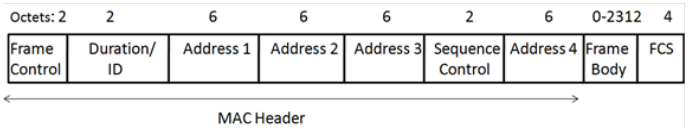

Fig. 3 IEEE 802.11 Frame

There are three different classes of 802.11 frames.

Management Frames

The management frames create and maintain communications between the components. The different messages involving management frames are:

- Association request
- Association response
- Reassociation request
- Reassociation response
- Probe request
- Probe response
- Beacon
- Announcement traffic indication message
- Disassociation
- Authentication
- Deauthentication

Most of the management frames also contain SSID. These Management messages are never encrypted, even when link encryption such as WEP/WPA/WPA-PSK is being used, so the SSID is visible to anyone who can intercept these frames.

Control Frames

The control frames help in data delivery. They perform functions of area clearing operations, channel acquisition carrier-sensing maintenance functions and positive acknowledgment of received data.

Data Frames

The data frames are the pack horses of 802.11 encapsulating the OSI Network Layer packets. These contain the source and destination MAC address, the BSSID, and the TCP/IP datagram and haul data from station to station. The payload part of the datagram is encrypted.

4.4.4 Authentication

The 802.11 authentication standard explicates that the mobile device (station) needs to establish its identity with an Access Point (AP) or broadband wireless router. For this purpose, the IEEE 802.11 standard has defined two types of authentication schemes.

Open system authentication

Open system authentication consists of two communications. The first is an authentication request from the mobile device that contains the station ID (typically the MAC address). This is followed by an authentication response from the AP/router containing a success or failure message. All stations are authenticated without any checking.

Shared Key Authentication

In the closed network architecture with shared key authentication, a shared key or passphrase is manually set on both the mobile device and the AP/router [7]. The stations must know the SSID of the AP in order to connect to the AP. There are many types of shared key authentication in use today for example WEP, WPA, WPA2 etc.

4.4.5 Association

After the completion of the authencation phase, for the station and AP to exchange data, a station has to associate with an AP in the infrastructure mode or with another station in the ad hoc mode. All the APs transmit the Beacon management frames that contain the SSID, capabilities, time and other information. Association allows the AP to record each mobile device so that frames may be properly delivered.

The association is a multi-step process. After the station and the AP mutually authenticate themselves by exchanging Authentication management frames the mobile device authenticates to an AP by sending an association Request. The AP processes the Association Request and after deciding whether or not a particular client request should be allowed, the AP responds with an Association Response frame which is basically a status code of 0 (success) and the Association ID. The latter is used to identify the station for delivery of buffered frames when power-saving is enabled. A station can be authenticated with several APs at the same time, but associated with at most one AP at any time. Association implies that authentication phase is already complete [7].

5 Penetration Testing Flow

We created an architecture which emulates a real time scenario. Wireless networks normally provide a "wedge" into a traditional wired network and in our design we directed attack based on of real risks of compromise inherent in the wireless infrastructure, and lookout for sensitive data stored elsewhere.

The following flowchart represents the flow of Penetration Testing for the Honeypot Architecture:

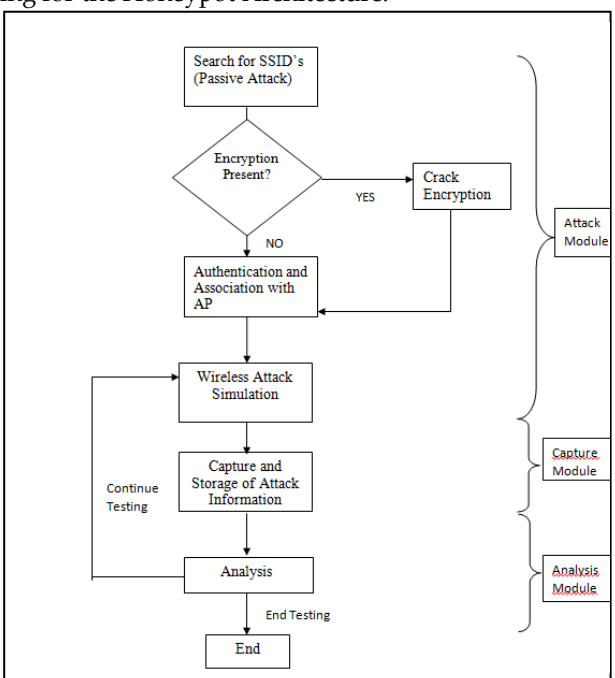

Fig. 4 Honeypot Architecture Pentesting Flow

The attack methodology can be subdivided as follows:

1. Attack and penetration of networks encrypted with WEP, WPA-PSK and WPA2-PSK
2. Network discovery and reconnaisance
3. Man-in-the-Middle (MITM) attacks
4. Identify devices interacting with the network
5. Stealing sessions and replaying
6. Traffic sniffing to capture confidential data
7. Evaluating Password weaknesses and vulnerabilities
8. SSID discovery and impersonation
9. Node detection in the network
10. Joining open/unsecured networks
11. Analysis and report generation

6 Attack Simulation

6.1 Looking for Wireless Networks – A passive attack

A passive attack occurs when someone listens to or eavesdrops on an open wireless network by using a wireless network adaptor rigged to work in promiscuous mode. All packets having SSID tokens, MAC addresses are stored for analysis. A passive attack may not be malicious at all times but it may be a stepping stone towards an active attack by an adversary.

The tool used for passive attack in this architecture was InSSIDer[8]. InSSIDer scans networks within reach of your computer's Wifi and also has the capability to determine security settings of a wireless network. NetStumbler[9] has long been a favorite for this sort of passive attack, but it doesn't work well with latest version of Windows.

Usage of a tool such as InSSIDer is only an initial step in the process of reconnaissance by the attacker. After finding the concerned SSID and related details, the attacker can connect to the wireless network to sniff and capture network traffic. This might expose a lot of details about the network and the enterprise that uses it. For example, analyzing the network traffic may reveal to the attacker DNS servers being used in the network, network names, unencrypted logon traffic etc. The attacker can decipher this information and figure out if the network is worth enough to proceed further with other attacks. Also, if the network is using a weak encryption scheme like WEP, the attacker can capture sufficient number of packets and crack the encryption to get inside the network.

6.2 WEP Key crack Simulation

For the wireless network, access points act as base stations. It is their task to receive and transmit data for the nodes to communicate within a wireless setup.

The SSID of the WLAN should be known to a client for it to join that WLAN; therefore, for the purpose of letting the clients know their SSIDs, the access points has a feature of beacon transmission, whereby it keeps transmitting a digitised signal so that any client which is in range can detect it in order to show it in the list of available wireless networks.

Data packets are continuously sent between the AP and its nodes. With the right tools in hand of an adversary, no physical access to the network is required to capture these packets. Since wireless networks can allow multiple nodes, to maintain confidentiality and integrity an authentication layer along with encryption is required before letting actual data transfer take place. It is in this layer where attackers can compromise a loophole and get inside the network.

6.2.1 Wireless Encryption

Many a time while setting up wireless networks administrators tend to leave the default key unchanged. Skilled adversaries try to detect the vendor of the access point and if they fail to do so, they try to break the pre-shared key that is used between the wireless AP and node to encrypt communication. Most of the home and small enterprise networks are encrypted using the two most popular encryption methods:

1. WEP
2. WPA

WEP or Wired Equivalent Privacy was designed to act as a default encryption method to protect link level data in wireless systems. It was introduced back in 1999 as part of the first 802.11 standard. It is a RC4 encryption based scheme supporting 3 different key lengths: 64, 128, and 256 bits also known as WEP 64, WEP 128, and WEP 256 respectively [10]. WEPsecurity is badly broken. However, due to its compatibility with older device it is still a widely popular encryption scheme for wireless networks and is used quite extensively. The WEP mechanism utilizes a user-defined or automatically generated key K and a 24 bit Initialization Vector (IV) [11] to encrypt the plaintext M and the checksum CS; the encrypted message is determined using the following formula:

$$C = [\, Msg \,||\, CS(Msg) \,] + [\, RC4(K \,||\, IV) \,]$$

WEP also comes in WEP2 and WEP+, which are not as common and still as vulnerable as the standard WEP encryption.

WPA comes in two modes WPA and WPA2, and was created as a solution to problems found in the WEP encryption scheme. Both WPA modes provide good amount of security; however, they are not compatible with older devices and therefore not as popular as WEP. By design WPA was made keeping in mind that every node has to be distributed different keys; however, it is still used quite extensively in a not so secure manner where every node has the same password for authentication encryption.

In our honeypot architecture we focused on the WEP encryption scheme and tried various new approaches for packet injection to crack the secret paraphrase in real time which are consequently captured by the honeypot having a fake AP running as an emulation service.

6.2.2 Packets and IVs

A wireless LAN may employ several security mechanisms. However, since all the wireless packets can be captured by anyone listening in promiscuous made it is imperative that the confidentiality and integrity is maintained as the attack surface in wireless is huge. In the WEP encryption scheme, every encrypted data packet would contain a 24 or 48 bit IV depending on the type of encryption. The motive behind using a random IV is to encrypt each WEP packet with a different key since the pre-shared key between the AP and the node is stat-

ic. That is to avoid a data packet from being encrypted by a twin encryption key the IV is constantly changed. Since to decrypt the encrypted WEP packet, the client needs to know the initialization vector, it is sent in plaintext.

Now here is a security issue. Theoretically, if every IV was different, it would be nearly impossible to obtain the network key; this is not the case. Since the IV is considerably small, it would wrap around eventually leading to potential reuse of the same key stream by different frames. For a 24 Bit IV only 16 million unique values can be used for encryption before repetition starts. This may seem like a very large number, but for a busy wireless network, it's miniscule.

Every IV is not unique and since the adversary knows that all the keys used to encrypt packets are related by a known IV (since the user entered parraphrase part of the key is rarely changed); the only change in the key is 24 bits. Also from the concept of probability for random variables, since the IV in WEP encryption is also randomly chosen, there is a fifty percent chance that the same IV will again reappear after just 5,000 network packets leading to a collision.

If an adversary can decipher the content of one packet, due to the collision he/she can view the contents of the other packet. If sufficient numbers of packets are collected whose IV match, the complete security of the wireless network can be breached.

6.2.3 Packet Injection

Option 1 - ARP Request Replay Attack

Address Resolution Protocol (ARP) is a required TCP/IP standard defined in RFC 826[12]. It is a TCP/IP protocol used to convert an IP address into a physical address. To be more precise ARP resolves IP addresses used by TCP/IP-based software to media access control addresses used by LAN hardware

Injection of packets into the network can be done by resending packets that have already been received. For this purpose the Aireplay tool [13] (part of the Aircrack Suite) can be used. The most efficacious way of creating initialization vectors is through the classic ARP request replay attack and most of the times it works very reliably. To achieve injection of IVs, the malicious node in our architecture keeps on listening for an ARP packet and then does a retransmission of the same packet back to the AP. This makes the AP believe that it has to send the ARP packet with a new IV. So the malicious node keeps on retransmitting the same ARP packet over and over again and the AP keeps send an ARP packet with a new IV. All these IVs collected by the malicious node allows to determine the WEP key.

Basic usage for aireplay:

aireplay-ng -3 -b 00:13:10:30:24:9C -h 00:11:22:33:44:55 eth0

Where:

- -3 means standard arp request replay
- -b 00:17:9A:82:32:51 is the access point MAC address
- -h 00:11:22:33:44:55 is the source MAC address
- eth0 is the wireless interface name

In our setup we replayed an ARP which was previously injected to save time. The trick is to use the same command plus the "-r" to read the output file from your last successful ARP replay.

aireplay-ng -3 -b 00:17:9A:82:32:51 -h 00:11:22:33:44:55 -r Achin.cap eth0

Where:

- -3 means standard arp request replay
- -b 00:17:9A:82:32:51 is the access point MAC address
- -h 00:11:22:33:44:55 is the source MAC address (either an associated client or from fake authentication)
- -r Achin.cap is the name of the file from the last successful ARP replay
- eth0 is the wireless interface name

Option 2 – Association/Disassociation

This was a unique way of getting enough IVs. We simulated the IV generation by creating a batch file to automate the authentication and deauthentication process using a legitimate client with the access point, the number of packets required to crack wep were collected very easily using this setup without using any specific tool.

```
#Windows Batch file
@echo off
for /L %%a in (1,1,1000) do (
netsh wlan connect ssid=test name=test
TIMEOUT 10
netsh wlan disconnect)
pause
```

Result

45 pcap files with 5000 packets each were collectedand were used as an input. A total of 27620 IVs were collected which subsequently helped in breaking the key.

```
[00:00:07] Tested 2742 keys (got 27620 IVs)

KB    depth   byte(vote)
 0    0/  2   31(37376) 80(33536) CB(33280) 23(33280) DB(32768)
 1    0/  1   32(40448) 14(36096) 99(34304) 68(34304) C8(33280)
 2   11/ 13   13(32000) 96(31744) 5F(31744) FF(31744) C8(31744)
 3    0/  3   34(37376) 34(34816) 90(34816) DA(33280) CD(33280)
 4    4/ 38   35(33536) 86(33280) 32(33280) 8C(33024) FE(33024)

KEY FOUND! [ 31:32:33:34:35 ] (ASCII: 12345 )
Decrypted correctly: 100%
```

6.3 ARP Cache Poisoning simulation

As discussed earlier in the ARP request replay attack, the Address Resolution Protocol serves the function of determining the mapping between IP addresses and MAC hardware addresses on local networks. For example, a machine that wants to send a message to IPv4 address 192.168.1.2 sends a broadcast ARP packet on the LAN that basically is asking for a MAC for that corresponding IP address. The host who's assigned IP is 192.168.1.2 sends back an ARP reply packet with its MAC address intact. This mapping from IP to Physical address is stored by the requesting host for future communication. This updation of cache that stores the IP mappings helps in minimizing network traffic and if in future communications, the MAC address corresponding to a given IP address has been changed, the old value in the cache is overridden. ARP replies are unicast packets, that is, only the requester receives them. However, ARP requests are always broadcasted in the entire LAN.

6.3.1 ARP Cache Poisoning in 802.11 Networks

In our architecture the access points acts as a hubs for all the hosts on the wireless network and they are doing the task of bridging traffic between the wireless network and the wired network. Now in this case, we can say that there are two separate collision domains [14]. One collision domain is defined by all hosts on the wireless subnet and the host on the wired network define the second collision domain. The AP's presence does not limit the broadcast domain and it extends to the wired network. Protocols such as DNS can be configured to only accept secured dynamic updates but since any node can send an ARP reply to another node it can update any node's cache with a new IP to MAC mapping. In our Honeypot environment this attack is even more dangerous as it is applicable to all hosts in a broadcast domain and because of the fact the access point acts as a bridge the ARP replies can propagate into the entire network.

6.3.2 Modus Operandi – ARP Cache Poisoning

Although there are tools to carry out this attack like Cain and Abel [15], as part of this research we created custom tools for ARP poisoning and demonstrated how our honeypot captures the attacks and shows alerts. We worked on the concept that if a MITM attack can be performed by a wireless attacker against two hosts present on the wired network connected to the same switch as the access point and the crafted ARP packets can reach both the victim hosts we can compromise the entire wired network from a wireless node.

It was necessary to forge the packets because ethernet frames that not addressed to the legitimate machine cannot be received by the malicious node as all NICs silently cast aside frames addressed to other MAC address (except for multicast Ethernet address). And specifically for this purpose we used Scapy [16] that allowed us to forge the packets and sending them using a raw socket.

```
Achink:$ scapy
>> ip_src="192.168.111.112"
>> ip_dst="192.168.111.101"
>> mac_src="00:00:00:00:00:AB"
>> mac_dst="00:00:00:00:00:BC"
>> ether_pack=Ether(src=mac_src,dst=mac_dst)
>>arp_pack=ARP(op=2,psrc=ip_src,pdst=ip_dst,hwdst=m
ac_dst,hwsrc=mac_src)
>> final_pack=ether_pack/arp_pack
>> sendp(final_pack,loop=1,inter=1)
```

6.3.3 Protecting against ARP Poisoning

- Creating static ARP entries with the correct IP/MAC Address matching.
- Blocking gratuitous ARP replies.
- Building custom software designed to monitor and protect your computer's ARP table.

6.4 Other Attacks

Following attacks were also simulated:
Evil Twin Attack
Dictionary Attack on WPA-PSK
Extensible Authentication Protocol Attack
De-Authentication Flooding

7 Capture Module

The designed honeypot was an amalgam of many capture mechanism. It included IDS, Open source Analysis Engine, Custom Emulation Modules for popular protocols and custom scripts.

7.1 Emulation Scripts and High privilege Processes

Emulation scripts of criticial services were created:

- SSH Emulation Script – This scripts tracks brute force attacks and logs them.
- Web Server Emulation Script – This script emulated all critical web issues such as remote file uploads, Cross site request forgert attacks and SQL injection.
- Windows services Emulation Script – This Script creates instances of windows services such as SMB and FTP.

7.2 Intrusion Detection System

Intrusion Detection is the art of detecting inappropriate, incorrect, or anomalous activity [17]. Among other tools that are employed in a network environment, an Intrusion Detection System (IDS) sole purpose is to determine if a computer network or server has experienced an unauthorized intrusion.

7.1.1 Snort

Snort was used as an IDS in our Honeypot architecture as it is one of the best open source intrusion detection system having considerable community base and also because it support three important functionalities which formed the core basic services of our honeypot :
Sniffer Mode: Snort can be used as a packet sniffer similar to wireshark and can be configured easily to display only IP headers or the payload as per the requirement.
Logger Mode: It also supports logging all the traffic into a file which can be used for forensics and analysis at a later stage.
Intrusion Detection mode: This is the core mode of' Snort. Snort maintains a database of signatures to which all packets can be compared. If any packet matches a malicious signature then alerts are sent.
Snort comes with a large repository of signatures (around 800 signatures) and it also has the ability to load additional plugins [18].
Since Snort just an Intrusion detection system at its grassworrts for our honeypot we needed a comprehensive analysis toolkit and a administration GUI. For this purpose we chose ACID (Analysis Console for Intrusion Databases). ACID is a web-based analysis toolkit that can be used to inspect Snort data (which is to be written into a database).

8. Analysis Using BASE - the Basic Analysis and Security Engine

BASE [19] is a tool that has the ability to search and process databases containing security events. It has been written in the

PHP programming language and also supports display of information from the DB in an organized manner. When BASE is used with Snort as the IDS, it has built in support to read both tcpdump binary log formats and Snort alert formats. Both layer-3 and layer-4 level packet details can be analysed once the data is logged and processed. In our analysis we generated graphs and stats for all the attacks that were simulated. The data from the custom emulation scripts was also utilized in generating the graphs in order to give a lucid picture. Custom scripts were also written to analyse and derive packet payloads corresponding to every attack so as to help in understanding attack methodologies. Since the the BASE search interface supports querying the database and generating visuals at runtime, all attacks were analysed effectively.
On top of that BASE also allowed us to easily manage alerts. We categorized high impact attacks into alert groups and minimized false positives by keeping track of previously handled alerts.

8. Conclusion

In this work, a honeypot for a network architecture having a mix of wired and wireless equipment is designed, with web-based monitoring and rule-based intrusion detection capability. The honeypot is interfaced with a SQL database, having a rich set of logging functionalities, and provided a convenient GUI for users to visualize the results.

The wireless to wired attacks simulated in our controlled penetration testing of the wireless architecture demonstrate that due to the wide range of attacks that can be carried out in such an environment, honeypots can act as a great resource in thwarting skilled adversaries and protecting critical resources from being breached. The ability of the honeypot to gather information about the attacker's tools and methodologies makes it an indispensable network component along with Firewalls and Intrusion detection systems. However, honeypot can still not be considered as a mandatory product with a fixed place in every security aware environment as firewalls or intrusion detection systems are today. There is a huge risk on the network being totally compromised if the honeypot has loopholes and therefore, there is a ever need for tight supervision and monitoring. Despite the risk, this security resource could easily become an effective way to monitor wireless intrusions attempts in critical network environments and can act as an important tool to understand a blackhat's goal.

9. Acknowledgment

The authors wish to thank Ms. Radhika Goel, Mr Emmanual Pilli and Mr. Raj Khati for their support and help. This work was supported in part by a grant from Microsoft Scholarship Award, Indian Institute of Technology Roorkee, India.

References

[1] Rick Schoeneck, "Wireless Honeypot, GIAC Security Essentials Certification (GSEC)", URL:http://www.giac.org/paper/gsec/2975/wireless-honeypot/104986, pp 7-8, June 2003
[2] ACID, "Analysis Console for Intrusion Databases", URL: http://acidlab.sourceforge.net/
[3] L. Splitzner, "Honeypots: Definitions and Value, URL:http://www.tracking-hackers.com/papers/honeypots.html," 2003
[4] Wireless networking, URL:http://cs.sru.edu/~mullins/cpsc100book/module08_networks/module08-03_networks.html.
[5] Techterms, URL: http://www.techterms.com/definition/ssid
[6] Microsoft technet Document, "How 802.11 Wireless Works", URL: http://technet.microsoft.com/enus/library/cc757419(v=ws.10).aspx
[7] Understanding IEEE* 802.11 Authentication and Association, Intel, URL: http://www.intel.com/support/wireless/wlan/sb/CS-025325.htm
[8] Insidder by Metageek, URL:http://www.metageek.net/products/inssider/
[9] Aaron Weiss, "Introduction to NetStumbler", URL: http://www.wi-fiplanet.com/tutorials/article.php/3589131, 2006
[10] Nikita Borisov, Ian Goldberg, David Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11", URL: http://www.isaac.cs.berkeley.edu/isaac/wep-draft.pdf, pp 1-3
[11] Martin Beck, Erik Tews "Practical Attacks Against WEP and WPA",URL: http://www.cs.rit.edu/~adb3160/crypto2/files/p79-tews.pdf, 2009
[12] An Ethernet Address Resolution Protocol, URL:http://tools.ietf.org/html/rfc826.html
[13] Aircrack NG,URL: http://www.aircrack-ng.org/
[14] Collision Domain, URL:http://www.inetdaemon.com/tutorials/networking/lan/collision_domain.shtml
[15] Rorik Koster, Own your LAN with ARP Poison Routing, URL:http://www.infosecwriters.com/text_resources/pdf/Arp_RKoster.pdf, pp 3-8, 2006
[16] Scapy, URL: http://www.secdev.org/projects/scapy/
[17] Sokratis K. Katsikas, "Intrusion Detection Systems ", URL:http://www.forth.gr/onassis/lectures/2010-06-28/presentations/Intrusion_Detection_Systems.pdf
[18] Reto baumann, "Honeypots", URL: http://www.rbaumann.net/download/diplomathesis.pdf, pp 11-18, 2002
[19] B. s. ideas, "Basic Analysis and Security Engine (BASE), URL:http://base.secureideas.net/," 2010.