

# HoneySpot: The Wireless HoneyPot

## Monitoring the Attacker's Activities in Wireless Networks

*A design and architectural overview*

*Raúl Siles*

*The Spanish HoneyNet Project (SHP)*

<http://www.honeynet.org.es>

Last Modified: *December 17, 2007*

### INTRODUCTION

Wireless technologies drive our world, a world without cables where information is available from anywhere at anytime. The huge expansion of wireless technologies, and specifically 802.11 wireless data networks, in the last years have provided a new battle field for information access. It is hard to find a place today in the main cities and surrounding areas of most first-world countries where there are no wireless data networks spreading around. End users and corporations are heavily interested in taking advantage of the flexibility, mobility and freedom offered by wireless technologies to access and share information. These allow connecting to data networks, like the Internet. Along with this freedom, though, come security issues that must be thoroughly understood and addressed.

Since its origins in 1999, when the HoneyNet Project ([www.honeynet.org](http://www.honeynet.org)) was founded, honeypot and honeynet solutions have been extensively used to monitor the attacker's activities in different IT environments. These solutions have evolved to offer support for multiple technologies, from pure TCP/IP network communications to more advanced application-level or focused attacks, such as Google hacking attacks or attacks against SCADA infrastructures. Surprisingly, honeypot solutions have not been widely applied to wireless technologies. This implies that there is a significant lack of knowledge about the current state-of-the-art of wireless attacks effectively used to break into wireless networks. Only after a wireless security breach, like in the famous TJX case [1] where WEP vulnerabilities lead to the biggest public theft of credit card numbers in history, it is possible to get more details about the real methods currently used by the attackers.

Due to the exponential usage of wireless equipment and technologies today, it is required to get an in-depth knowledge about the real exploitation vectors currently used to compromise wireless networks. Trying to fill this knowledge gap, the main goal of this research is to analyze the state of real life wireless hacking, and introduce and promote the design and deployment of wireless honeypots.

In this paper we will first provide an overview of wireless honeypots along history, to further analyze the wireless honeypots objectives and its taxonomy. The paper then mainly focuses on the details of the design and architecture of an 802.11 wireless honeypot, providing an extensive overview of its different components and their requirements. Some guidance is provided from a deployment and implementation perspective, and finally, how this solution can be further enhanced, and extended to other wireless environments, is detailed.

## WIRELESS HONEYPOTS HISTORY

Kevin Poulsen published in 2002 one of the first news about the existence of wireless honeypots [2], a new way of trapping hackers, covering what was called the first organized wireless honeypot. The Wireless Information Security Experiment, or WISE, was launched in June 15<sup>th</sup>, 2002, by Science Applications International Corporation (SAIC) in Washington DC (US). The focus of this initial research was driven by the inherent insecurity of wireless networks at that time, and the fact that most of them were deliberately open. Unauthorized network access, use, and eavesdropping, were and, are still today, the major threats against wireless networks.

WISE, led by Rob Lee <sup>1</sup>, was “an 802.11b network based at a secret location in Washington D.C. and dedicated to no other purpose than being hacked from nearby.”, and closely monitoring the attacker’s activities. At that time the concept of client honeypots didn’t exist, so “like conventional honeypots, the WISE network has no legitimate users, so anything that crosses it is closely scrutinized.”

At the end of 2002, other organizations like Tenebris published the results of collecting data from a wireless honeypot [3] deployed in Ottawa (Canada), and confirmed the huge war driving activity taking place at that time, and the existence of targeted intrusions. There are just a few references in 2003[4], 2004 [5] and 2005 [6], covering the results of wireless honeypots deployments around the city of London, promoting the idea of using wireless honeypots as a deception mechanism, and investigating the unauthorized use of wireless networks in Adelaide (South Australia), respectively.

Finally, one of the more detailed articles published on wireless honeypots, “Wireless Honeypot Countermeasures”, was released at the beginning of 2004 [7] by Laurent Oudot, a HoneyNet Project member. It was focused on providing an introduction to the goals, design and limitations of wireless honeypots, and provided practical examples using honeyd and FakeAP. Surprisingly, since the initial news on this topic in 2002, and the slight follow up in 2004, there has not been any relevant public research about wireless honeypots.

Recently in 2006, the MAP project [8] was born using a three-point approach, (MAP - Measure, Analyze, Protect), to develop an integrated and extensible framework to address existing and future attacks on WiFi networks. Wireless honeypots are referenced in the Measurement component: “We will extend our state-of-the-art honeypot technology to build wireless honeypots that, when attacked and subsequently used to attack the wireless network itself, allow us a detailed look at the attacker’s methods and activities. Finally, we will develop novel honeypots that emulate real VoIP handsets, in anticipation that these handsets will be the target of future attacks.” Unfortunately, not too many details have been released about this project, and specifically, about the wireless honeypot solutions involved.

The latest news about wireless honeypots point to Raytheon, that in 2007 sponsored a wireless honeypot research project, dubbed “The Hive” [9], at the University of Florida, to help address wireless threats. The project is based on a Linux Live CD environment that provides access point capabilities and entire networks simulations through honeyd.

---

<sup>1</sup> The author could gather most of the details of the WISE project directly from Rob Lee in a presentation held during a SANS conference in 2003, under the scope of the HoneyNet Project.

## HONEYSPOT: DEFINITION AND TAXONOMY

The research presented in this paper is focused on introducing and analyzing the options available for the design and implementation of a wireless honeypot. Specifically, the paper goes in depth into WiFi honeypots, that is, honeypots for wireless data networks, or local area networks (LAN), based on the IEEE 802.11 standards. Similar ideas as the ones presented can be applied to other wireless technologies, such as MAN/WAN wireless communications based on WiMAX, or personal wireless communications (PAN) based on Bluetooth.

A new term has been coined to refer to wireless honeypots for IEEE 802.11 technologies, **HoneySpot**.

### HoneySpot

The **HoneySpot** term has been coined mixing two terms, **Honeypot** and **Hotspot**. Based on the HoneyNet Project definition (<http://www.honeynet.org/misc/faq.html>), “A honeypot is a system whose value is being probed, attacked, or compromised, you want the bad guys to interact with it”. Based on the Wikipedia definition ([http://en.wikipedia.org/wiki/Hotspot\\_%28Wi-Fi%29](http://en.wikipedia.org/wiki/Hotspot_%28Wi-Fi%29)), a hotspot is “A hotspot is a venue that offers Wi-Fi access. The public can use a laptop, WiFi phone, or other suitable portable device to access the Internet”.

Therefore the HoneySpot term accurately summarizes the concept around wireless honeypots: “A **HoneySpot** is a venue that offers Wi-Fi access whose value is being probed, attacked, or compromised, you want the bad guys to interact with it”.

Most of the previous wireless honeypot projects and research (see the “Wireless Honeypots History” section) were based on easily providing wireless connectivity to the attacker in order to research the activities carried over the wireless network against local or remote resources. Their main purpose was to dispel the myth that attacks on wireless networks were simply an attempt to obtain free Internet access. Additionally, for those studies where some kind of wireless hacking analysis was performed, it was focused on the wireless network itself, that is, the access points.

There are different attack scenarios to consider when talking about wireless technologies and honeypots, as represented in Figure 1:

- A. Attacks directed towards the wired network to which the wireless network connects. These attacks use the wireless network as a medium but the primary target is the network or information systems beyond it.
- B. Attacks directed towards the wireless users. These attacks may use the wireless network as a medium to target the user’s device wireless capabilities, and exploit the fact that the wireless device is enabled. The user may or may not be connected to a wireless network.
- C. Attacks directed towards the wireless network infrastructure. These attacks focus on gaining control of the access point or wireless controllers, that is, the wireless infrastructure devices.

In the first two scenarios, although malicious users might direct attacks against the wireless network itself (trying to gain access to it by breaking the access control mechanisms), the wireless network is not the goal but the first step.

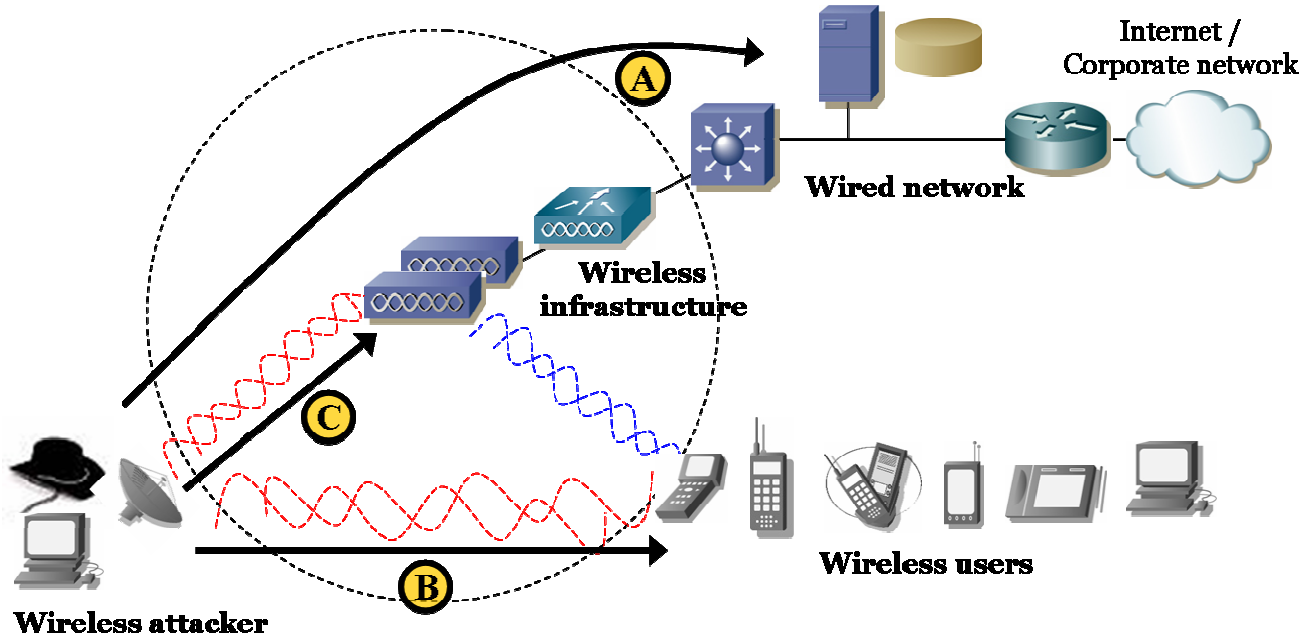


Figure 1. Wireless attack scenarios

The research proposed in this paper forgets about previous research conceptions and focuses on pure layer-2 wireless attacks, that is, those focused on trying to break into a secure wireless network. For this same reason, this research does not consider plain open wireless networks or default AP configurations scenarios which do not have any control mechanism in place, as there is no effort for the attacker to break into them and use their capabilities. Additionally, the paper tries to help the research of direct wireless client attacks, one of the main new threats now that the wireless infrastructures can be highly secure; attackers are directing their focus onto vulnerable clients. Besides, previous projects only deployed a small amount of wireless honeypots, so the conclusions obtained are not relevant enough to announce conclusive results. The purpose of this paper is to raise awareness and interest on the topic, so that multiple HoneySpots are deployed worldwide and it is possible to reach “definitive” conclusions about the wireless hacking scene.

Therefore, it is important to clarify that the main goal of a HoneySpot is to gather information about the attacks performed on the wireless network and the associated technologies. Specifically in those attacks that exploit the wireless technologies weaknesses, subvert the security mechanisms in place, and that are mainly focused on the radio frequency (RF) and 802.11-based vulnerabilities. Therefore, the goal is not to analyze the IP-level attacks that are carried out over HoneySpot wireless networks.

Although potentially very similar to the attacks performed over wired networks, attacks carried out over wireless networks could be analyzed in future related research using the same wireless honeypot design described here. Such research could focus on analyzing and collecting statistics about the type and nature of IP-level attacks executed over wireless networks, like targeted remote system exploitation, phishing attacks, SPAM generation, botnets traffic and control, etc. Other security mechanism at the IP-layer, commonly used in wireless networks,

such as VPNs (IPSec or SSL-based) could be added to the analysis. The results obtained could be contrasted with the data collected from wired honeynets with the goal of deriving some conclusions and comparisons about the attacker's profiles and objectives in both environments, wireless and wired. This extended IP analysis could allow confirming if wireless networks are really used to launch other attacks due to the anonymity they provide. The major difference with honeypots connected to a globally available network, such as the Internet, is that wireless honeypots are not globally accessible and can only be attacked from malicious users "nearby"<sup>2</sup>, therefore, HoneySpots might be used to research if different locations might see different types of attacks.

Although in the past some literature referred to FakeAP or Hotspotter [10] as wireless honeypot tools, this is not the concept of a wireless honeypot referred in this paper. These tools, and others like Karma [10], allow the creation of fake access points with the goal of attracting the wireless user attention. They can be used for evil, to attract and compromise benign wireless users, or can be used for good to attract and monitor malicious users. However, the lack of flexibility and capabilities to simulate and monitor in-depth a wireless network do not make them the preferred solution.

In order to reduce the HoneySpot definition and scope, two types of HoneySpots have been defined:

- A **Public HoneySpot** simulates a public wireless data network, that is, a pure hotspot. Hotspots are commonly available at hotels, airports, coffee shops, libraries, as well as other public places where there is a high interest in offering Internet connectivity to visitors and customers.
- A **Private HoneySpot** simulates a private wireless data network, such as those available in corporations or at home. Typically, a private network offers access to a wired network (corporate or home network) to legitimate wireless clients without the physical barriers associated to wired connections.

The type of threats and attacks a HoneySpot is exposed to in both environments are very different:

- On the one hand, in a public wireless network or hotspot, any user is allowed to access the wireless network in order to get basic connectivity and be able to interact with the network infrastructure. These types of networks have no access control mechanism at the wireless level. Once IP access has been achieved, it is possible (and required from the perspective of this research) to have other controls in place to limit the access to the network only to subscribers or customers. Typically these controls (called Universal Access Method, UAM) are implemented in the form of a controller, a firewall plus a web-based captive portal, that allow users to authenticate using a web browser in order to get full network access. Since the main purpose of these hotspots is offering full Internet access, a Public HoneySpot is mainly interested in:
  - o Direct client attacks focused on exploiting client vulnerabilities at the operating system level, such as vulnerabilities in the wireless administration client software, or the wireless drivers.
  - o Attacks focused on the wireless infrastructure, such as the access point.
  - o Attacks focused on bypassing the security controls imposed by the controller (firewall and captive portal), and obtaining full network access without being a legitimate subscriber.

One of the major challenges in a Public HoneySpot is the need to differentiate between intruders, and those who simply stumble onto nearby networks for convenient and free Internet access, sometimes even

---

<sup>2</sup> The attacker needs to be within the range of the wireless network. This means he could potentially be kilometers away from the target network depending on the equipment being used.

unknowingly due to the automatic wireless connection capabilities of their wireless client software. This does not apply to Private HoneySpots, closed networks by definition, where layer-2 access is restricted.

- On the other hand, in a private wireless network only authorized users are allowed to connect to the network. These types of networks will typically have access control mechanisms restricting access to the wireless network itself and, in the case of corporations, might have additional access control layers, such as Network Access Control (or NAC) to limit the access of mobile devices to the private network. The main purpose of a Private HoneySpot is to identify if unauthorized attackers can get access to the network overcoming the wireless-related security mechanisms in place. Therefore, it is mainly focused on:
  - o Direct client attacks focused on exploiting client vulnerabilities at the operating system level, such as vulnerabilities in the wireless administration client software, or the wireless drivers.
  - o Attacks focused on the wireless infrastructure, such as the access point.
  - o Attacks focused on exploiting wireless vulnerabilities at the protocol level, such as weaknesses on WEP or WPA/2, or on the authentication mechanisms offered by WEP, WPA/2 pre-shared keys, or 802.1X/EAP.
  - o Attacks focused on bypassing other security mechanisms available on wireless deployments, like MAC address filtering, turned off SSID broadcast, etc, with the goal of getting full network layer-2 access.

The initial HoneySpot design was mainly focused on layer-2 wireless attacks against the wireless infrastructure (access point, RADIUS server, etc), wireless clients, and wireless technologies: 802.11, its security mechanisms (WEP, WPA/2, etc), and EAP/802.1X protocols. However, an overall evaluation of the current wireless scene emphasized the need to cover the widely deployed and extensively used hotspot environments, and create two HoneySpot types.

The main difference between the two types is that a Public HoneySpot focuses on the wireless attacks once IP connectivity is available (they are “open” networks), and a Private HoneySpot focuses on how an attacker tries to get access to a supposedly “close” network. In the former case, there is a grey line between wireless specific attacks at the IP level (for example, specific to the web-based captive portal), and generic IP-level attacks common to wireless and wired environment. The research focus is on the first group of attacks.

The purpose of each HoneySpot type is focused on analyzing the specific threats and attacks associated to the type of wireless network provided. There are multiple security mechanisms that can be implemented in a wireless environment. A HoneySpot can be deployed with different security levels depending on the level of complexity of the attacks the researcher is interested in. These security levels are described in the “HoneySpot Design” section.

Similarly to the deployment of other honeynet technologies, legal issues must be addressed with wireless honeypots. Although the legal connotations are beyond the scope of this paper, it is strongly recommended to get detailed information about the legal implications (especially in reference to privacy, entrapment or liability issues) of deploying this kind of technology in your environment, specific to your country and organization. For example, international wiretap laws prohibit the interception of electronic communications and traffic monitoring, with some exceptions [14].

## HONEYSPOT DESIGN

The latest wireless security threats not only try to exploit vulnerabilities on the wireless infrastructure, but on the weakest link in a wireless network - the clients. A HoneySpot must cover attacks that target both infrastructure and client systems. The complexity of a HoneySpot is noteworthy compared with a traditional honeypot. A HoneySpot is focused on wireless attacks, no matter if they are addressed against the clients or the servers, the access point or wireless infrastructure in this case. Therefore, it combines the requirements of traditional server honeypots with the requirements of the recent client honeypots research, or honeyclients. The HoneySpot must provide the wireless infrastructure plus the client capabilities required to simulate the presence of wireless clients in the wireless network.

From a conceptual perspective, any traffic going to or coming from the HoneySpot is likely a probe, attack or compromise attempt, except the self-generated traffic that simulates the HoneySpot wireless clients. Wireless honeypots present the same weaknesses as traditional honeypots, where skilled attackers may identify the presence of the honeypot based on the differences with a real wireless network. Consequently, the goal is to perfectly simulate the reality. One of the main design differences between a Public and a Private HoneySpot is the need to provide access to, or emulate, the Internet on a Public HoneySpot. While the main purpose of a public environment or hotspot is providing Internet access, this is not the case for a Private HoneySpot. Due to the fact that the private version is focused in layer-2 attacks, there is no need to provide a complete IP infrastructure.

The main goal of a HoneySpot from a design perspective is to reveal real statistics about wireless attacks, such as type of attacks, frequency of attacks, the attacker's skill level, goals and methods, and even help to determine the hacking tools being used. Although from an enterprise perspective a HoneySpot could help to protect a network, because the attacker spends a significant effort on it versus on the real wireless network, this is not the initial design purpose of a HoneySpot.

The current HoneySpot proposal is focused on the deployment of wireless infrastructure networks, where an access point provides and controls the access to the medium. Future research could also investigate the threats associated to dynamic ad-hoc (or peer-to-peer) wireless networks, where the access to the medium is managed by one of the participants.

A HoneySpot is designed to help to answer questions such as:

- What attack techniques are more extensively used to break into and exploit the weaknesses of WEP-based wireless networks? WEP key cracking (with or without traffic injection), chop-chop attacks, WEP fragmentation attacks, etc.
- What attack techniques are more extensively used to bypass hotspot access controls? Session hijacking through MAC and IP address spoofing, web-based session hijacking, service theft through protocol tunneling, etc.
- What attack techniques are more extensively used to compromise wireless clients? Wireless driver vulnerabilities exploitation, hotspot IP traffic injection, AP impersonation, etc.

The identification of all these attack types would allow evaluating the real skills of wireless attackers today, and help to deploy effective and realistic security countermeasures for wireless networks in the future.

## Wireless attacks

The goal of a HoneySpot is to detect as much wireless attacks as possible, those well-known today, as well as new ones the could be discovered in the future. As a reference, a HoneySpot must be able to detect and discern an extensive set of well-known wireless attack types, listed on Appendix C. Apart from the pure wireless threats mentioned, there are other IP-layer attacks common to wireless networks (see Appendix C) that should also be detected by a HoneySpot <sup>3</sup>.

Some of the wireless attacks are completely passive, such as the eavesdropping of traffic required to crack the WEP encryption key, so there is no opportunity to detect them. However, if the attacker wants to speed up the WEP cracking process he needs to use active techniques, like traffic replay or injection. These additional actions make possible to detect the presence of the attacker. From a threat perspective, we assume the attacker can passively capture the wireless network traffic.

In a wireless network it is possible to impersonate any other client by using MAC address spoofing techniques. In order to differentiate the traffic coming from the real user and from the attacker, both using the same source MAC address, it is required to use advanced techniques focused in the analysis of the signal strength and other particularities in 802.11 frames, like sequence numbers.

## HoneySpot Security Levels

Wireless attackers present different security skills. The type and the complexity of the attacks we are interested in researching about must drive the type and security level of the HoneySpot deployed. Several HoneySpot security levels have been identified for both the public and private versions:

- Public HoneySpot:
  - o Level 0: Open wireless network (with IP-layer controls)
- Private HoneySpot:
  - o Level 0: WEP-based wireless network
  - o Level 1: WPA-based wireless network
  - o Level 2: WPA2-based wireless network

Some providers, like T-Mobile (<http://hotspot.t-mobile.com>), offer public wireless access in hotspot environments with advanced security mechanism in place, such as WPA and 802.1X [11]. Wisely, they have decided to authenticate the hotspot subscribers at layer-2 using 802.1x/EAP, instead of using a layer-3 mechanism like a web-based captive portal. The main benefit is that the legitimate user needs to authenticate prior to getting any network access, while in the old model, anyone can get network access (IP connectivity) because it is required to go through the authentication process using the web browser. This type of advanced wireless hotspot networks fit in the Private HoneySpot category as previously defined since they use exactly the same technologies and infrastructure

---

<sup>3</sup> Wireless IP-level attacks are mainly associated to hotspot, or Public HoneySpot, environments where IP connectivity is available to any visitor. They can also occur in Private HoneySpots once the attacker has obtained access to the protected wireless network, although they are out of the scope of Private HoneySpots.

typical of corporate wireless networks, although they are still considered hotspots because they provide Internet access to their subscribers.

For every HoneySpot deployment, independently of the type and security level, it is required to document the specific security mechanisms that have been implemented. The following list details the most common security mechanism available in commercial AP's, and therefore available for the HoneySpot:

	Open	WEP	WPA	WPA2
<b>Authentication method</b>	Open	Open/Shared <sup>1</sup>	PSK/Enterprise	
- 802.1X/EAP type	N/A	Dynamic WEP <sup>2</sup>	PEAP, EAP/TLS, TTLS, etc. <sup>3</sup>	
<b>Encryption method</b>	No encryption	WEP (RC4)	TKIP (RC4)	TKIP/CCMP (AES)
<b>MAC address filtering</b>	yes/no			
<b>SSID broadcast</b>	yes/no			
<b>PSPF <sup>4</sup></b>	yes/no			
<b>Captive portal <sup>5</sup></b>	yes/no			

**NOTES:**

- <sup>1</sup> The authentication for WEP-based shared, or closed, networks is based on the WEP-key.
- <sup>2</sup> Dynamic WEP (DWEAP) makes use of 802.1X/EAP to generate and renew the WEP keys. It is required to know the EAP type implemented with DWEAP prior to its deployment.
- <sup>3</sup> When WPA/WPA2 networks use Enterprise (802.1X/EAP) mode, versus PSK (pre-shared key) mode, it is required to know the specific EAP type to be used prior to its deployment.
- <sup>4</sup> PSPF, Publicly Secure Packet Forwarding, is a wireless station isolation technique to deliver a private VLAN for each 802.11 client on a single AP, so that clients cannot communicate directly with each other.
- <sup>5</sup> Captive portals are typically used in open wireless networks as the main authentication and access control mechanism. However, although there is no technical reason why they cannot be used with WEP or WPA/2 based networks, it is not common to use them on WPA/2 Enterprise networks where the authentication is provided by a more robust security mechanism, like 802.1X/EAP.

The HoneySpot deployment process must include the documentation of the HoneySpot being deployed, detailing (between others) the type, security level, and the security features implemented. Appendix A provides a “HoneySpot Deployment Cheat-Sheet” to gather all this information.

### HONEYSPOT ARCHITECTURE

Once the HoneySpot design principles and main objectives have been defined, it is mandatory to identify the main components and their requirements from an architectural perspective. The following modules (or components) are mandatory in HoneySpot architectures, represented in Figure 2:

- **Wireless Access Point (WAP) module:** The WAP module provides the wireless network infrastructure for clients and attackers to connect to. This infrastructure is mainly conformed by one or multiple access points (AP's) that offer a wireless network. The network is the main target for attackers on a HoneySpot.
- **Wireless Client(s) (WC) module:** The WC module represents legitimate and automated end user devices that connect to the HoneySpot wireless network. It is required to simulate real wireless traffic and provide the minimum traffic required by the attacker to launch some specific wireless attacks, such as accelerated WEP key cracking through traffic replay techniques.
- **Wireless Monitor (WMON) module:** The WMON module is a device that collects all the wireless traffic on the HoneySpot for real-time and offline analysis. This is a crucial component as it provides the capability to capture all wireless activities taking place inside the HoneySpot.
- **Wireless Data Analysis (WDA) module:** The WDA module provides the capabilities required to analyze all the network traffic collected by the WMON module. Once the collected traffic is transferred from the WMON to the WDA, the WDA goal is to identify malicious activities and all their details.
- **Wired Infrastructure (WI) module (optional):** On the one hand, a HoneySpot can be deployed without any associated wired networking infrastructure. It will simply provide wireless connectivity with no additional networking capabilities. On the other hand, trying to accurately simulate real-world scenarios, a HoneySpot can have an associated wired networking infrastructure that simulates the internal network of an organization or a hotspot Internet connection. The WI module is the module that provides these extended capabilities and it is an optional component because a HoneySpot, particularly a Private HoneySpot, can be made of just wireless components, where no wired or Internet infrastructure is required for the analysis of layer-2 wireless attacks.

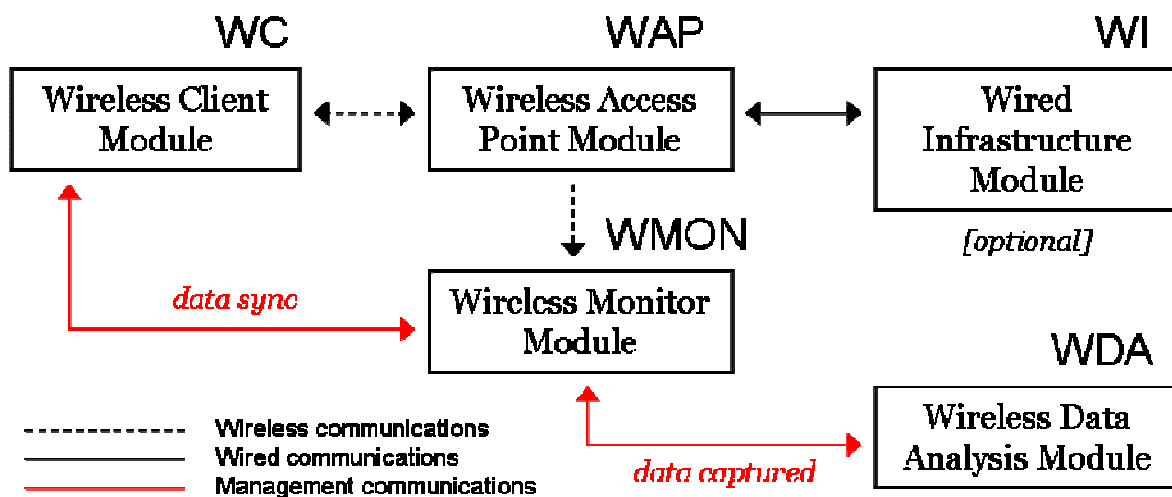


Figure 2. HoneySpot Architecture

The detailed HoneySpot architecture including the different modules described below is presented in Figure 3.

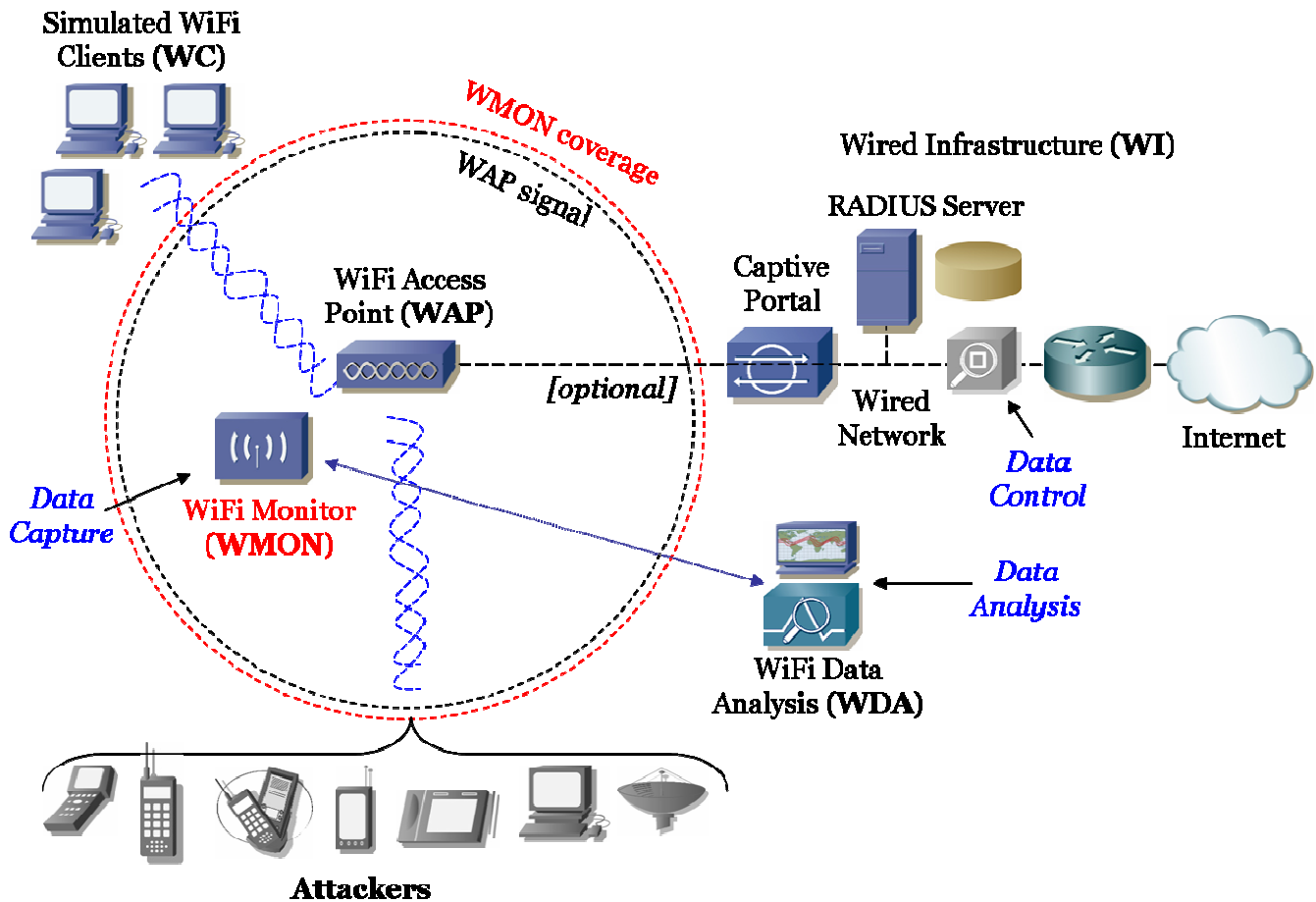


Figure 3. Detailed HoneySpot Architecture

The following sections provide an overview of the design and deployment considerations for each module.

### Wireless Access Point (WAP) Module

The WAP module provides the wireless network infrastructure. It must simulate a standard Wi-Fi access point, so it should be configured with the default transmission (TX) power and omnidirectional antennas (unless a higher coverage range is required or the area of interest to be covered is specifically delimited).

The WAP can be implemented using a real wireless access point or a generic computer with a wireless card in master mode. In any case, it must be configured for the specific HoneySpot type, security level, and the specific security features selected (see the “HoneySpot Deployment Cheat-Sheet” in Appendix A).

The WAP channel selection must be established after performing a wireless signal survey around the HoneySpot location, trying to avoid conflicts with nearby wireless network and reduce the extra traffic coming from networks

in the same channel. It is important to reduce as much as possible the amount of traffic that must be captured by the WMON module.

The current HoneySpot design only considers the deployment of a single WAP (access point) per HoneySpot. There could be scenarios where multiple WAP's are needed to cover a broad area, such as the whole floor of a building. In this case, it is crucial to evaluate the channel distribution for the different WAP's and the relationship with the WMON's. If the wireless capabilities of the WAP's and WMON's are very similar, one WMON is required per each WAP. If multiple WAP's are deployed, it is crucial to evaluate the requirements from a traffic capture perspective (WMON) to monitor clients roaming between access points and correlate its traffic [13].

If the HoneySpot research is not specifically focused on evaluating the chance of the access point itself being compromised, it is mandatory to perform a detailed hardening of the WAP prior to its deployment. Amongst other aspects, it is required to:

- Limit the services (TCP and UDP) exposed to the wireless network. It is recommended to manage the AP from the wired network.
- Use a robust device administration username and password to avoid password guessing attacks.
- Update the AP firmware to the latest version to avoid the exploitation of well-known vulnerabilities.

Additionally, it could be desirable to automate a procedure to reset the WAP and reinstate it to a known state periodically to ensure that its configuration has not been modified.

### **Wireless Client(s) (WC) Module**

Apart from providing a wireless infrastructure in the form of a wireless network, the HoneySpot requires to simulate the presence of wireless clients that interact with the wireless network. This is the purpose of the WC module.

For a device to act as a wireless client, the card must be configured in managed mode, with the appropriate settings required to connect to the WAP. The WC should be located from a standard distance of the WAP. To simplify the deployment it could be between 1-10 meters far from the WAP.

There are multiples options to implement the simulation of client traffic in an automated way <sup>4</sup>. The solution must be able to establish multiple individual connections against the wireless network, or even remote Internet hosts, simulating the presence of multiple clients.

The design of this module must consider the following aspects:

- Number of clients to simulate. Each client should use its own unique MAC address.
- Type of simulated traffic. Each client could generate traffic associated to one or multiple protocols (802.1x, ARP, ICMP, TCP, UDP, IPSec, etc) and applications, such as (secure) web browsing, FTP, SSH, VPN traffic, e-mail access, etc.
- The amount of traffic to be generated per client. Each simulated client should be configured independently.

---

<sup>4</sup> Although no public tool has been released for this specific purpose, it is feasible to generate the traffic from customized scripts, traffic replay tools like tcpreplay, or traffic generators.

Client traffic must be generated in such a way that a casual observer of the wireless network cannot easily determine that the traffic has been automated. Consequently, the different traffic profiles simulated need to be generated in a random fashion and with varying information data exchanges. If an attacker can easily determine that a client is not a “real” client, for example, by observing that its web browsing pattern is reduced to access to the same site over time, there will be no incentive to compromise it.

The complexity of the WC module can be overwhelming based on the level of detail and objective tests. For example, it is possible to design dormant clients, that establish a connection, exchange some traffic, and go to sleep for a few minutes or hours, with the goal of evaluating active and passive session hijacking through MAC/IP address spoofing in hotspot-like environments. Similarly, it is possible to pre-design a Preferred Network List (PNL) for the simulated wireless clients to evaluate the occurrence of PNL attacks through access point impersonation. The list goes on and on based on the specific client attacks the HoneySpot must focus on.

### **Wireless Monitor (WMON) Module**

The Wireless Monitor (WMON) module implements the Data Capture requirement [15], focused on “capturing all the HoneySpot activities and all the information that enters and leaves the HoneySpot, without attackers knowing they are being watched.” As the main focus is in wireless layer 2 attacks, the WMON module requires a wireless card in monitor (RFMON) mode to passively sniff all the wireless traffic.

The main goal of the WMON is to collect all the wireless traffic in PCAP format <sup>5</sup> for later offline analysis. However, it is highly desirable to additionally provide real-time wireless IDS capabilities to this module, so that it can alert the analyst about suspicious events almost in real-time. Unfortunately, the open-source options for wireless IDS are pretty much limited to Snort-Wireless [16], an outdated Snort enhanced version that provides specific layer-2 wireless attack detection capabilities.

There are multiple technical challenges associated to wireless traffic capture [13]. The more relevant ones are:

- The WMON technology, such as 802.11g, must be set up in accordance with the technology used for the WAP.
- There are significant performance and storage requirements in the capturing device in order to accommodate the amount of traffic that can be generated on a wireless network over time.

As a minimum prerequisite, the WMON must at least listen on the same wireless channel the WAP is configured in order to capture all the activities from and to the WAP. Additionally, it is strongly recommended to have an additional radio in the WMON device constantly scanning the activities taking place in the other channels. This additional information is very valuable to analyze the attacker activities, such as wireless discovery scans. If the additional radio is available, no matter the location (country) where the HoneySpot is deployed, the WMON should monitor all the 802.11 channels defined, independently of the regulation. For example, for 802.11b and 802.11g networks, this means channels 1 to 14 (US: 1-11; Europe: 1-13; Japan: 1-14). Please check the law that applies to the

---

<sup>5</sup> There are multiple standard network capture tools, such as tcpdump, or Wireshark/tshark, that allow to capture the network traffic in PCAP format,

location where the HoneySpot is being deployed, but typically the reception on any channel does not break any law (transmitting on a channel out of the regulation does).

In the best-case scenario the WMON could be based on a wireless monitor device capable of collecting traffic from all the 14 802.11b/g channels simultaneously, such as those used for wireless forensics cases [13].

In the ideal scenario, the WMON wireless coverage should be equal to the area covered by the WAP signal, so that only the attacks in the WAP scope are monitored. When dealing with radio frequency (RF) technologies, this level of accuracy is very difficult to reach. The proposed solution requires that the WMON coverage must be equal or greater than the area covered by the WAP signal, trying not to miss any event affecting the HoneySpot wireless network. This requirement is accomplished by selecting and adjusting the hardware of the WMON in relation with the WAP hardware. The two main factors to be considered are the receive sensitivity and the antennas of both devices. Similarly to the WAP, the WMON needs to use omnidirectional antennas.

A crucial element in the WMON design is that it must be able to discern between all the simulated traffic generated by the WC module and any other real traffic. Both types of traffic should be correlated to find stimulus and response scenarios, where the attacker activities are driven by the WC generated traffic. Therefore, the WC and WMON modules required to communicate to synchronize their activities in real-time, or make the pattern programmed in the WC known beforehand to the WMON so it can be filtered out.

### **Wireless Data Analysis (WDA) module**

The Wireless Data Analysis (WDA) module allows the researcher to process and analyze all the information collected by the WMON module. A direct communication channel with the WMON module is required in order to retrieve all the network traffic captured in PCAP format.

The WDA module implements the Data Analysis requirement, focused on providing information intelligence and a simple and clear overview of the activities taking place within the HoneySpot. The implementation should be based on PCAP-aware network traffic analysis tools (preferable graphical tools) that process the captured files and generate overall statistics, as well as provide in-depth details of all the events captured by the WMON module. The granularity (or level of detail) of the data offered by this module is a crucial aspect to be able to gather all the information of specific wireless-based attacks and support further wireless incident handling and forensic investigations. For this purpose, full raw data packets are needed, including the signaling information (RSSI) collected by the wireless drivers running on the WMON module.

Besides the offline capabilities of the WDA module, having real-time detection and analysis capabilities in the WDA module, such as those provided by a wireless IDS management console, help to process and get the most relevant details from the alerts generated by the real-time component (wireless IDS) within the WMON. The WDA module should centralize, correlate and display all the events generated by the WMON module.

The WDA module has similar or bigger storage requirements than the WMON module. This module retrieves the data collected by the WMON module for offline analysis and statistics generation, and it is highly recommended to save at least several weeks of collected data.

### **Wired Infrastructure (WI) Module**

The goal of the Wired Infrastructure (WI) module is to provide the minimum components required by the wireless infrastructure, such as a captive portal for hotspot/public networks, or a RADIUS server for 802.1x private networks. In accordance to the research and design objectives, the wired infrastructure could or could not include other IP-based target resources that can be attacked once the attacker gets IP connectivity on the wireless network.

Additionally, for Public HoneySpots the WI should provide Internet connectivity in order to simulate an interactive and realistic public hotspot. This is one of the main differentiators between a Public and a Private HoneySpot. The Internet portion can be a real Internet connection through an ISP, or be simulated. Internet can be simulated through honeynet software, such as honeyd (<http://www.honeyd.org/concepts.php>), providing multiple “virtual” networks, routers and hosts. Alternatively, a real Internet connection can be used for the WI module. In this case, it is very important to ensure that the Data Control requirement is met [4]. If a HoneySpot is compromised, we must be able to contain any malicious activity and ensure the HoneySpot is not used to harm other Internet hosts. Based on the Honeynet Project definition, “there must be some means of controlling how traffic can flow in and out of the HoneySpot, without attackers detecting control activities. Data Control always takes priority over Data Capture.” At least, the control layer must limit outbound traffic from the HoneySpot to mitigate any liability issues derived from the HoneySpot being compromised and used illegally by the attackers. The wired control layer can be implemented using the Honeywall CDROM (<http://www.honeynet.org/tools/cdrom/>) control capabilities, although most probably the standard Honeywall version, designed for wired honeynets, would need to be accommodated to meet the specific HoneySpot and wireless traffic requirements.

Apart from the Internet access, for a Public HoneySpot it is required to provide all the common hotspot components, such as a default gateway, the controller (firewall plus web-based captive portal), and the authentication (RADIUS, LDAP, etc) and network services (DHCP, DNS, etc.).

A Private HoneySpot might be deployed with or without a WI module based on its main goals. As defined in this paper, no external wired connectivity is required and, thus, no WI module. Deployments which wish to analyze attacks to the wired infrastructure will want to include a WI module to simulate the “corporate” network and some common services and devices in this network, such as servers, printers, workstations, etc.

In the case of using a WPA/2 enterprise network, where a RADIUS server is required to perform the 802.1x/EAP authentication, the server can be implemented inside the WAP. The usage of an embedded device running OpenWRT allows running both the wireless AP and RADIUS server on the same device (the WAP module). Alternatively, the RADIUS can be located in a small WI module.

## HONEYSPOT IMPLEMENTATION OVERVIEW

The previous sections have detailed the main requirements and limitations imposed by each of the HoneySpot modules. Although they include some implementation guidance, the specific implementation details of a HoneySpot, including the main issues and concerns found during our first HoneySpot implementation, are left for a future paper.

The major considerations that need to be evaluated in order to implement a HoneySpot are:

- **Cost:** One of the critical aspects for the deployment of multiple HoneySpots in real-world environments is cost. The goal is to reduce the cost for the acquisition of the different hardware components that shape the solution. There are two major alternatives for the implementation of the different modules:
  - o The expensive, but more flexible option based on standard laptops.
  - o The cost-effective, but limited option based on embedded devices, such as the Linksys WRT54GL.
- **Security:** As a security solution, a HoneySpot cannot be compromised, because the overall results and research would be useless. It is important to provide some kind of isolation capabilities between the most vulnerable and exposed components, such as the WAP and WC, and the private and critical components, such as the WMON. The preferred protection mechanism is physical isolation. However, even when physical separation is possible, it is required to have private management communications between the WC and WMON, and the WMON and WDA. These connections must be analyzed in depth to avoid, for example, the risk of the WMON being compromised through a WC compromise.
- **Physical Security:** Unlike other types of honeypots, which might be deployed in secure locations, HoneySpots might be installed in public places where it might be difficult to ensure that physical access to some of the components is not possible. Physical attacks can range from the theft of the device itself to a configuration change through its manipulation.
- **Real-world simulation:** The most complex mandatory HoneySpot module is the WC module. It requires to accurately simulating the traffic generated by wireless clients in a wireless network.

A laptop running Linux, with the appropriate wireless card (or cards), can be used to implement any of the HoneySpot modules. It provides all the required storage space, wireless capabilities to host multiple cards (PCMCIA, Express Cards or USB), and support for other external peripherals, such as the Wi-Spy spectrum analyzer [18]. These three advantages mainly summarize the constraints of the embedded device option. Although embedded devices can run OpenWRT [20] (a Linux-based OS), the lack of storage space, the fact that they typically have a built-in single wireless radio, and the lack of external interfaces, present serious limitations to add new wireless cards or other external devices. Both hardware options can run Linux, the selected OS for all the HoneySpot software tools.

From an implementation perspective all the modules can be combined into a single box or separated to individual systems. Each alternative presents its pros and cons:

- A single box is a highly portable solution, cheaper and self-contained.
- Multiple boxes provide isolation capabilities (very important from a security perspective), and more storage and computation resources, although obviously, are more expensive.

Trying to meet all the previous concerns, and focusing on the cost and security aspects from a realistic perspective, the initially proposed solution for a Private HoneySpot is based on using the following hardware, presented on Figure 4:

- Laptop running Linux with a single wireless card (in RFMON mode) for the WMON and WDA modules, where signal strength information, extensive storage, and computation capabilities are available to collect and analyze the wireless traffic (see sidebar below). The laptop requires a wired connection for the HoneySpot management communications (e.g. with the WC module).
- Linksys WRT54GL for the WC module.
- Linksys WRT54GL (or any standard wireless AP) for the WAP module.
- No WI module is required for the initial Private HoneySpot prototype.
- In this first prototype, there are two options for the WMON and WDA modules:
  - o Both can be integrated on the same laptop device, although that requires to remotely access the laptop to analyze the data collected.
  - o The WDA can be implemented remotely, that is, the captured PCAP files are transferred (manually or automatically) to a remote system for later analysis. This is the preferred option.

The specific location where the different HoneySpot modules are deployed must be carefully evaluated to ensure reasonable physical security controls. HoneySpot components cannot be easily accessible, avoiding the risk of unauthorized physical access, or even disappear. This requirement requires to carefully evaluating the signal range prerequisites.

### Using Embedded Devices for HoneySpots

The following are major limitations to use embedded devices for some of the HoneySpot modules:

The closed and proprietary nature of the default wireless Broadcom chipset and drivers for the Linksys WRT54G (and associated family of devices: G, GL, etc.) makes impossible to retrieve signal information (RSSI data) from this device [12]. Other embedded devices, like the Asus WL-500G Premium, equipped with a miniPCI slot that can accommodate open wireless chipsets like Atheros, could be used for the WMON module.

The limited storage space available on embedded devices can be accommodated through hardware hacks, like the addition of a SD card to a WRT54GL device [12], or through embedded devices that provide a built-in USB port, like the Asus WL-500G Premium or the Linksys WRTSL54GS. This USB requirement is also needed for the connection of external peripherals like the Wi-Spy USB dongle [12].

One of the most relevant implementation-specific aspects is the design of the HoneySpot remote management capabilities or architecture. For some of the tasks already described, it is required to interconnect the different HoneySpot modules between them. For this purpose it is desirable to implement an isolated management network, that allows accessing the management interface of all modules in a secure way, basically through SSH or HTTPS. Using this network it is possible to manage and reconfigure all the modules remotely, as well as implement the communication capabilities required between:

- The WC and WMON, in order to synchronize the information about the simulated wireless traffic.

- The WMON and WDA, in order to share the captured PCAP traffic files.

The access to this network must be physically protected, and if it provides remote access capabilities, an external firewall needs to implement the filtering policy to restrict access to all the management interfaces. Additionally, it is recommended to run some kind of traffic filtering mechanism in the management interface of all the HoneySpot modules. The same management network can be used for other tasks, such as accurately synchronized the clock of all the modules using the Network Time Protocol (NTP). An overview of the initial management architecture used for the first HoneySpot prototype is presented in Figure 4.

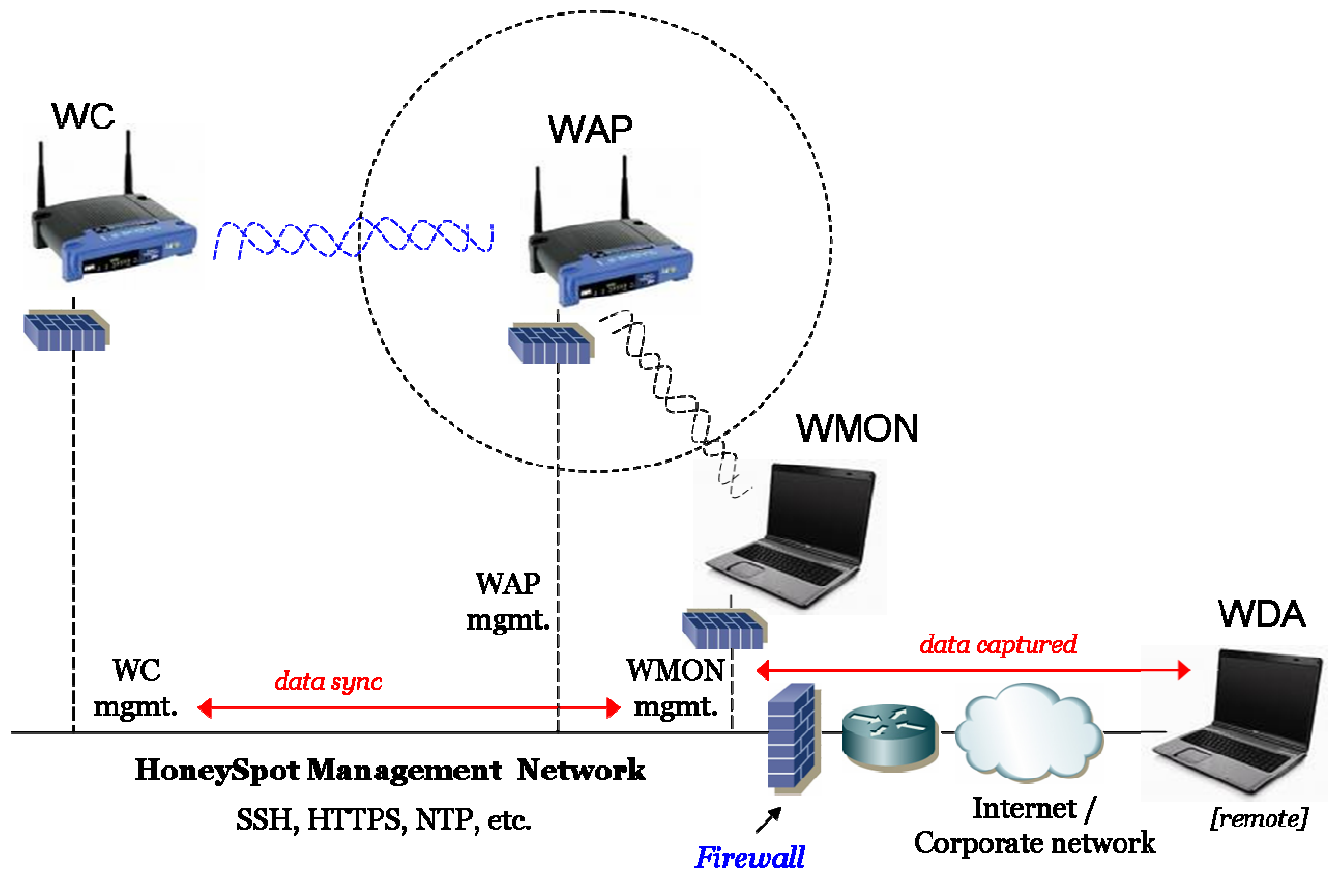


Figure 4. First Private HoneySpot Prototype & Management Network

One of the first steps to accomplish after a HoneySpot deployment is a wireless signal leakage analysis, documenting the range from where the HoneySpot signal can be received in the area surrounding the HoneySpot location. It is recommended to document in detail the results obtained and the wireless hardware used, emphasizing the use of standard hardware and avoiding high-gain antennas (unless specifically desired). The “HoneySpot Deployment Cheat-Sheet” available on Appendix A provides room to reflect this information.

## HONEYSPOOT EVOLUTION

The following are some desirable advanced capabilities for future versions of 802.11 (WiFi) HoneySpots.

One of the future goals of HoneySpots would be to have advanced capabilities to locate where the attacker is placed in relation with the WAP and WMON modules. This will help to estimate the details about the wireless gear used by attackers (antennas and wireless card specifications), and even identify the intruder through the inspection of the estimated physical location. For this advanced analysis very detailed signal analysis tools are required, plus extra capabilities to triangulate the attacker position based on the signal levels. This means at least three WMON modules are required and might even need specialized antennas. The conclusions of future research about the attacker location would help to determine if wireless attacks are commonly performed from a nearby physical location to the target network, or from farther distances using directional antennas.

The initial research on this paper is focused on the most widely deployed 802.11 standards today, 802.11b and 802.11g. The purpose of the WAP module is to work in b/g compatibility mode to accommodate both types of clients. In a future version, it would be very interesting to analyze the attacks against 802.11a networks, as the current number of 802.11a wireless networks is clearly reduced. However, due to the wide availability of multi-band wireless cards supporting the three technologies, 802.11 a/b/g, it is well worth to investigate if attackers are scanning and targeting 802.11a networks too.

Additionally, with the current publication of the 802.11n draft by the WiFi Alliance on the summer of 2007 [17], a future HoneySpot evolution requires to also analyze this multi-radio 802.11 standard. The 802.11n technology introduces a new threat, 802.11n rogue AP's operating in GreenField mode. GreenField is an operation mode to maximize the bandwidth of 802.11n networks, but it effectively renders these networks invisible to existing 802.11a/b/g wireless cards and monitoring systems.

One of the major weaknesses of any wireless technology is its physical medium, based on radio frequency (RF). Because wireless transmissions must travel through the air in the form of RF waves, they are prone to Denial of Service (DoS) attacks. An attacker simply needs to generate enough RF noise in the wireless network frequency to saturate the medium and made impossible the establishment of any communication. HoneySpots can be easily enhanced to monitor the physical medium through the addition of a spectrum analyzer, such as the Wi-Spy USB dongle [18]. This device can be added to the WMON monitor to simultaneously scan at the physical layer (layer-1) and at the link layer (layer-2). From a cost perspective, if an embedded device like the Linksys WRTGS54SL is used as the WMON module, it can be also added to it [12]. Additionally, the latest Kismet version includes a plug-in to monitor the RF using Wi-Spy [19]. This fact, plus the standard Kismet monitor capabilities, makes possible to correlate the activities at layer-1 and layer-2, and identify the 802.11 frames associated to peaks in the frequency spectrum.

Since this design is focused on the detection and analysis of layer-2 attacks, based on the assumption that higher level attacks are similar to other honeynet deployments, it might be interesting to contrast if this assumption is true by deploying HoneySpots with a honeynet "wired" side populated with simulated services.

Future related research will be focused on building similar wireless honeypot solutions for Bluetooth technologies. The term coined to refer to a Bluetooth wireless honeypots is Bluepot.

## SUMMARY

Wireless technologies are the fastest growing segments of today's telecommunications and computing industry. The ubiquity of wireless networks, both in enterprises and at home, makes extremely important to evaluate and accommodate the security mechanisms currently available to the real threats and attacks. This paper defines HoneySpots, its objectives and taxonomy, and provides an overview of the main considerations and requirements from a design and architectural perspective.

HoneySpots are targeted to research the real security threats associated with the continuous growth and consolidation of wireless networks in the enterprise and home worlds. Few organizations deploy specific wireless IDS systems, so most of the wireless attacks go unnoticed. One of the goals of this research is to create awareness about the current wireless threats and promote the need to monitor the wireless networks and the RF spectrum. The conclusions will help improve the defenses used to protect wireless networks.

The current HoneySpot design is focused on researching about attacks on wireless networks and technologies, learning about the attacker's tools, tactics and motives. Future versions could expand the current proposal to also research about attacks using the wireless network as the attacks launch point. The associated research would be very similar to the one performed through honeynets on wired infrastructures, mainly based on IP-based attacks.

## ACKNOWLEDGEMENTS

The author would like to thank the following people, members of the Spanish Honeynet Project (<http://www.honeynet.org.es>), for their review and contribution:

- Diego González Gómez
- Javier Fernandez-Sanguino Peña



## REFERENCES

- [1] “How Credit-Card Data Went Out Wireless Door. Biggest Known Theft Came from Retailer With Old, Weak Security”. Joseph Pereira. The Wall Street Journal. May 4, 2007.  
[http://online.wsj.com/article\\_email/SB117824446226991797-lMyQjAxMDE3NzA4NDIwNDQoWj.html](http://online.wsj.com/article_email/SB117824446226991797-lMyQjAxMDE3NzA4NDIwNDQoWj.html)
- [2] “Wi-Fi Honeypots a New Hacker Trap”. Kevin Poulsen. SecurityFocus. 2002-07-29.  
<http://www.securityfocus.com/news/552>
- [3] “Tenebris Wireless Honeypot Project: Assessing the threat against wireless access points”(Version 1.0). Eric Jacksch. November 19, 2002.  
<http://www.tenebris.ca/docs/TWHP20021119.pdf>
- [4] “Commuters hack wireless networks”. BBC News. 26 March, 2003.  
<http://news.bbc.co.uk/2/hi/technology/2885339.stm>
- [5] “Implementing network defense using deception in a wireless honeypot”. Suen Yek. School of Computer and Information Science, Edith Cowan University. 2004.  
<http://scissec.scis.ecu.edu.au/publications/forensics04/Yek.pdf>
- [6] “An Investigation of Unauthorized Use of Wireless Networks in Adelaide, South Australia”. Phillip Pudney and Jill. SlaySpringer Berlin / Heidelberg. July 11, 2005.  
<http://www.springerlink.com/content/9ccg622leouqaqwk/>
- [7] “Wireless Honeypot Countermeasures”. Laurent Oudot. SecurityFocus Infocus. 2004-02-13.  
<http://www.securityfocus.com/infocus/1761>
- [8] “The MAP Project”. Dartmouth College. 2006.  
<http://www.cs.dartmouth.edu/~map/>
- [9] “Wireless honeypots”. Randall Brooks. Raytheon. 2007.  
[http://www.raytheon.com/technology\\_today/current/feature\\_5.html](http://www.raytheon.com/technology_today/current/feature_5.html)  
<http://www.cise.ufl.edu/class/thehive/>
- [10] Fake wireless access point simulation and creation tools:  
- FakeAP (Black Alchemy Enterprises): <http://www.blackalchemy.to/project/fakeap/>  
- Hotspotter (Max Moser): [http://www.remote-exploit.org/codes\\_hotspotter.html](http://www.remote-exploit.org/codes_hotspotter.html)  
- Karma (Dino A. Dai Zovi): <http://www.theta44.org/karma/>
- [11] “WPA™ Deployment Guidelines for Public Access Wi-Fi® Networks”. Wi-Fi Alliance. October 28, 2004.  
[http://www.wi-fi.org/files/wp\\_6\\_WPA%20Deployment%20for%20Public%20Access\\_10-28-04.pdf](http://www.wi-fi.org/files/wp_6_WPA%20Deployment%20for%20Public%20Access_10-28-04.pdf)
- [12] “Linksys WRT54G Ultimate Hacking”. Paul Asadoorian and Larry Pesce. Raul Siles (Technical Editor). Syngress. May, 2007. ISBN: 1597491667.

- [13] “Wireless Forensics: Tapping the Air - Part One”. Raul Siles. SecurityFocus Infocus. 2007-01-02.  
<http://www.securityfocus.com/infocus/1884>
- [14] ” Wireless Forensics: Tapping the Air - Part Two”. Raul Siles. SecurityFocus Infocus. 2007-01-08.  
<http://www.securityfocus.com/infocus/1885>
- [15] “Honeynet Definitions, Requirements, and Standards” (ver 1.6.0). The Honeynet Project. 14 October, 2004.  
<http://www.honeynet.org/alliance/requirements.html>
- [16] Snort-Wireless. Andrew Lockhart. 2005.  
<http://snort-wireless.org>
- [17] “Wi-Fi CERTIFIED™ 802.11n draft 2.0: Taking Wi-Fi® to the Next Level”. Wi-Fi Alliance. May, 2007.  
[http://www.wi-fi.org/files/kc/WFA\\_802\\_11n\\_Consumers\\_May07.pdf](http://www.wi-fi.org/files/kc/WFA_802_11n_Consumers_May07.pdf)
- [18] Wi-Spy Spectrum Analyzer. MetaGeek.  
<http://www.metageek.net/Products/Wi-Spy>
- [19] Kismet: Wi-Spy plug-in and Spectools. Mike Kershaw.  
<http://www.kismetwireless.net/spectools/>
- [20] OpenWRT. Linux distribution for embedded devices.  
<http://openwrt.org>

# Appendix A

## HONEYSPOT DEPLOYMENT CHEAT-SHEET

HONEYSPOT DEPLOYMENT CHEAT-SHEET	
<b>HoneySpot name</b>	
<b>Location</b>	
<b>HoneySpot Type</b>	
<b>Type</b>	Public or Private
<b>Security level</b>	Level 0, 1 or 2
<b>802.11 technology</b>	E.g. 802.11b, 11g, 11b/g, etc.
<b>Basic Features</b>	
<b>SSID</b>	E.g. linksys
<b>Channel</b>	1-14
<b>WiFi IP range</b>	Eg. 192.168.1.0/24
<b>DHCP server</b>	On/off <i>DHCP IP address range</i>
<b>AP IP address/mask</b>	
<b>AP management</b>	Protocols allowed: Eg. HTTP, HTTPS, Telnet, SSH, etc.
- Wired side	
- Wireless side	
<b>Admin username/pass</b>	
<b>Security Features</b>	
<b>Authentication method</b>	Open, Open/Shared (WEP), PSK or Enterprise (WPA/2)
- 802.1X/EAP type	<i>For DWEP and WPA/2 only (PEAP, EAP/TLS, TTLS, etc.)</i>
<b>Encryption method</b>	No encryption (Open), WEP, TKIP or CCMP
- Key	<i>For WEP or WPA/2-PSK only</i>
- Key length	<i>For WEP or WPA/2-PSK only</i>
- Re-keying interval	<i>For DWEP only</i>
<b>MAC address filtering</b>	Yes/no <i>List of MAC addresses filtered</i>
<b>SSID broadcast</b>	Yes/no
<b>PSPF</b>	Yes/no
<b>Captive portal</b>	Yes/no
- Details:	<i>Details of the captive portal software and settings</i>

<b>Wireless Signal Leakage Analysis</b>	
---	--

<b>Details</b>	
----------------	--

# Appendix B

## HONEYSPOT MODULES HW/SW SPECIFICATIONS CHECKLIST

HONEYSPOT MODULES HW/SW SPECIFICATIONS CHECKLIST
<b>Wireless Access Point (WAP) module</b>
<ul style="list-style-type: none"> <li>- <b>Standard WiFi access point</b>, or standard computer (laptop or desktop) running Linux, with wireless and wired networking capabilities.</li> <li>- The access point must provide support for all the functionality (802.11b/g, captive portal, RADIUS server, etc – see WI also –) and security features (WEP, WPA/2, MAC filtering, etc) required for the deployment.</li> <li>- Wireless radio and antennas “equal/similar to” the WMON module.</li> <li>- The wireless card must run in master (AP) mode.</li> <li>- No special software is required on this module.</li> <li>- A wired management connection between the WMON and the WAP is required.</li> </ul>
<b>Wireless Client(s) (WC) module</b>
<ul style="list-style-type: none"> <li>- <b>Standard computer (laptop or desktop) running Linux, or Linux-based embedded device</b>, with wireless and wired networking capabilities.</li> <li>- The client system must provide support for all the functionality (captive portal, RADIUS server, etc) and security features (WEP, WPA/2, MAC filtering, etc) required by the WAP module.</li> <li>- The wireless card must run in managed (client) mode.</li> <li>- Customized software (tools or scripts) is required to simulate the client traffic.</li> <li>- A wired management connection between the WMON and the WC is required.</li> </ul>
<b>Wireless Monitor (WMON) module</b>
<ul style="list-style-type: none"> <li>- <b>Standard computer (laptop or desktop) running Linux</b>, or Linux-based embedded device, with wireless and wired networking capabilities.</li> <li>- The monitor system requires a significant amount of storage space (internal or external – USB) for the captured traffic.</li> <li>- Wireless radio and antennas “equal/similar to” the WAP module.</li> <li>- The wireless card must run in monitor (RFMON) mode.</li> <li>- The wireless card must support the capture of signal (RSSI) information (Prism header, radio tap header, etc).</li> <li>- Standard software is required to capture the wireless traffic (tcpdump, Wireshark, etc).</li> <li>- Standard/customized software is required to analyze the wireless traffic in real-time (Snort-Wireless).</li> <li>- An extra wireless card is recommended to capture traffic from all channels (channel hopping).</li> <li>- Wired management connections between the WMON and the WC and WAP are required.</li> <li>- A wired remote management connection between the WMON and the WDA is required.</li> </ul>
<b>Wireless Data Analysis (WDA) module</b>
<ul style="list-style-type: none"> <li>- <b>Standard computer (laptop or desktop) running Linux</b> with wired networking capabilities.</li> <li>- The analysis system requires a significant amount of storage space for the captured traffic.</li> <li>- Standard/customized software (tools or scripts) is required to analyze and generate statistics from the wireless traffic.</li> <li>- A wired remote management connection between the WMON and WDA is required.</li> </ul>

**Wireless Infrastructure (WI) module (*optional*)**

- **Standard computer (laptop or desktop) running Linux** with wired networking capabilities.
- The infrastructure system requires all the resources needed (CPU, memory, storage space, etc) for the creation of the simulated or real wired infrastructure.
- The infrastructure must provide support for all the functionality (captive portal, RADIUS server, etc –see WAP module –) required for the deployment.
- Standard/customized software (tools or scripts) is required to simulate, or provide access to a real, wired infrastructure.
- A wired management connection between the WAP and WI is required.
- Internet connectivity could be required, including the appropriate controls (e.g. Honeywall) needed to limit the traffic generated to the Internet.

# Appendix C

## LIST OF WELL-KNOWN WIRELESS ATTACKS

List of well-known wireless attack types that should be detected by a HoneySpot:

- Active network scanning (E.g. Wardriving activities through Netstumbler)
- WEP-based attacks:
  - o WEP key guessing (Nessus Datacom)
  - o Dictionary attacks against the WEP key
  - o FMS attacks (and enhancements), aka WEP key cracking:
    - With or without traffic replay
  - o Inverse Inductive WEP attack (aka chop-chop)
  - o PRGA determination and packet injection:
    - Challenge/Response WEP authentication
    - IV collisions and known plaintext
    - Chop-chop
  - o WEP fragmentation attacks
- WPA/2-based attacks:
  - o Dictionary attacks against the WPA/2-PSK key
- 802.1X/EAP attacks:
  - o Dictionary attacks against LEAP authentication
  - o LEAP MitM attacks
  - o EAP username and password guessing (EAPOL Logon)
  - o EAP MitM attacks
- Wireless clients attacks:
  - o Wireless driver vulnerabilities exploitation
    - Wireless client and driver fingerprinting
  - o Wireless 802.11 protocols fuzzing
    - With and without 802.11 fragmentation
  - o PSPF attacks (direct traffic injection and eavesdropping)
  - o Rogue access points (AP's) and AP impersonation
    - Open and WEP-based spoofed access points
    - Preferred Network List (PNL) attacks
    - PEAP and TTLS configuration weaknesses
- Denial of Service (DoS) attacks:
  - o Physical medium (e.g. RF jamming)
  - o 802.11 weaknesses in management and control frames
    - Authentication, deauthentication, association, and deassociation
    - Beacons, invalid authentication type, etc.
    - Power management (TIM and PS-Poll)
    - Medium management (RTC/CTS)
  - o TKIP built-in DoS (MIC integrity)

- With or without WMM extensions
- EAP Logon/Logoff attacks
- Firmware (cards and AP's) vulnerabilities

## LIST OF WELL-KNOWN IP-LEVEL ATTACKS OVER WIRELESS NETWORKS

Apart from the pure wireless threats listed above, there are other IP-layer attacks common to wireless networks:

- Wireless clients attacks:
  - Hotspot IP traffic injection
  - Session hijacking (web-based vulnerabilities)
  - MitM attacks (e.g. SSL)
- Access point vulnerabilities and attacks:
  - Username and password guessing
  - Vulnerable services (HTTP, DHCP, SNMP, etc)
- Hotspot (public wireless infrastructure) attacks:
  - Hotspot controller vulnerabilities
  - Web-based captive portal vulnerabilities
  - Session hijacking through MAC and IP address spoofing
    - Active and passive
  - Service theft through protocol tunneling (DNS, ICMP, etc)