COMPUSOFT, An international journal of advanced computer technology, 3 (2), February-2014 (Volume-III, Issue-II)

ISSN:2320-0790

An International Journal of Advanced Computer Technology

Beacon Frame Manipulation to Mitigate Rogue Access Points: Case of Android Smartphone Rogue Access Points

[1]Thambo Nyathi, [2]Siqabukile Ndlovu
Computer Science Department
National University of Science and Technology
P. O. Box AC 939
Ascot, Bulawayo, Zimbabwe

Abstract: The use of wireless devices to access corporate network resources is now part of the norm within corporate environments. When wireless users need to connect to a network they hardly question the source of their connectivity. Mobile phones, particularly smartphones allow users to access network resources. These harmless looking wireless devices can be a source of major threats if configured to be so. The Internet is awash with mobile apps capable of performing packet sniffing. These applications, coupled with the capability of the smartphone to be configured as an access point, can present a Smartphone Rogue Access Point. Access Points advertise their availability using what is called a beacon frame. This research paper proposes a solution which restructures this beacon frame to include an Authentic Access Point Value which can be used to defend against Rogue Access Points.

Keywords - *Smartphone, Wireless, Rogue, Access Point, Beacon and Rogue Access Point*

I. INTRODUCTION

The Android security model allows the user to be in total control of the Android device [1]. Capable users can configure and use their Android smartphones as access points [2]. With smartphones configured this way, attackers are able to perform man-in-the-middle attacks or any other form of attacks on wireless networks [3]. Wireless networks have become popular because of their improved speed and accessibility [4]. The advancement in Transistor-Transistor Logic (TTL) and battery technology has led to the proliferation of mobile phones. 17% of mobile phone owners find it convenient to use their devices rather than a personal computer or laptop to access a network [5]. Mobile phones, particularly smartphones, are equipped with web browsers also known as micro browsers to allow users to access network resources. Network access, including Internet access for mobile and wireless devices, is facilitated by an access controller which is usually an access point or a wireless router. An access point (AP) is a base station in a wireless local area network (WLAN). If more than one access point is used, users can roam with their mobile devices and be handed off from one cell to another [6] like in the Global System Mobile (GSM) network. It is no longer a strange sight for an employee within an organization to use a mobile device for example smartphone or iPad to access the organization’s network resources.

A. *Access Point Technology*

An 802.11 Wireless Fidelity (Wi-Fi) access point provides a connection between the electrical data path formed by an Ethernet cable and the radio frequency (RF) signal data path formed by the Wi-Fi radios. Access points provide a number of additional features and capabilities related to access management, network traffic encryption, fault tolerance and network management. An AP is a translational bridge that converts the TCP/IP data packets from their wireless technology frame encapsulation format in the air to the Ethernet frame format on the wired Ethernet network.

As per the 802.11 specification the process of network connection between a client and an AP takes the following steps. The AP continuously sends out Beacon Frames which are detected by WLAN clients within range. The client can also broadcast its own probe-request-frame on every channel. Access points within range of the client respond with a probe-response-frame and the client selects the best access point for connection and sends an authentication request to that AP. The AP will send an authentication reply and upon successful authentication the client will send an association request to the AP to which the AP will reply with an association response. The client has been authenticated and can now pass traffic to the access point. [7]

B. *Rogue Access Points*

A rogue access point (RAP) can be defined as an access point which exists and is connected to but not authorized for operation in a network. It is unmanaged but attached to an enterprise’s network [8]. A Smartphone Rogue Access Point (SRAP) is an access point that is configured using a smartphone which can be deployed for malicious purposes. A RAP can appear in a network due to unknowing users plugging in their wireless adapters to improve access to network resources, misconfigured access points or can be intentionally introduced by a malicious user. These produce an AP that does not conform to WLAN security policies of an organization and this opens an insecure interface to the organization's network from outside [9]. The RAP allows just about anyone with a wireless device to access the network, and this puts them very close to critical resources [10]. There is no effective way to prevent rogue access points from appearing in a network [8][9]. It is however optimistic to have a RAP detection mechanism in a network but this is very difficult as it may be impossible to figure out which of the detected access points are actually rogue [10]. RAP detection should be implemented as part of the security policy of an organization. When it comes to efficient wireless security, putting policies in place is just as important as applying physical protection [8][10]. It is therefore important to implement RAP detection as part of the security policy of an organization so that network administrators are able to identify illegitimate access points before the users connect to them. This paper, proposes a solution that defends against rogue access points including smartphone rogue access points in 802.11 networks. This paper outlines related work, that is, current research on RAP detection, approach to the stated problem and results of the outcome of the experiment.

II. PROBLEM

A. Problem Statement

Due to their hotspot functionality, smartphones can be configured into access points. This can be achieved using the inbuilt device settings where the device is configured as a portable Wi-Fi hotspot. The SSID of the hotspot (access point) can be set to any preferred name while other properties of the hotspot such as the security options can be set optionally. Users can have elevated user priviledges that will allow them to load custom software; increase device performance and other activities that may change the software and configurations of the device. On smartphones running on the Android operating system, this process is called rooting. Rooting refers to obtaining super-user rights and permissions on an Android smartphone's software [11]. This is similar to the root user in Linux based systems. Rooting applications are widely available on the internet, for example ODIN [12]. Once rooted, packet sniffing applications can then be installed on the smartphone [16].

B. Threat Model

Enabling the hotspot functionality, which is available on most smartphones, allows a smartphone to be configured as an access point. Malicious users can take advantage of this functionality, install packet sniffing applications and then deploy a Smartphone Rogue Access Point (SRAP) on a network [14]. This malicious individual can visit an organisation; sit at the reception pretending to be waiting for someone while playing with their smartphone in the mean time capturing packets from unsuspecting employees. The intruder can then sniff the Service Set Identifier (SSID) and connection password of the organisation's AP and then deploy her SRAP with the same. The unsuspecting employees will then connect to the network using the SRAP allowing the malicious user to perform further attacks on that network. Figure 1 illustrates a possible threat scenario.

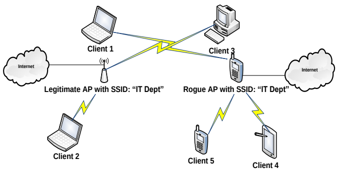
Client 1
Client 3
Internet
Internet
Legitimate AP with SSID: "IT Dept"
Rogue AP with SSID: "IT Dept"
Client 2
Client 5
Client 4

Figure 1: SRAP Threat

In Figure 1, clients 2 and 3 are connected via the legitimate AP while 1, 4 and 5 are connected via the SRAP.

III. RELATED WORK

A. RAP Detection using Network Traffic Analysis

In their paper titled "Rogue Access Point Detection by Analyzing Network Traffic Characteristics", Shetty, Song and Ma proposed an approach for RAP detection based on analysis of traffic at the edge of a network for a network comprising of both wired and wireless devices [13]. The solution first determines the origin of the packets. For packets originating from a wireless link, it checks whether the host is authorised to use that network. This is important because wireless access sources can be RAPs and it is done based on the frequency of access to a particular port and the increase in cross-port communication. If a host shows a remarkable increase in the above categories then that host is connected to a RAP. For packets originating from wired sources, the solution does not carry out any further analysis.

The main idea of implementing this approach is to help distinguish between authorised WLAN hosts from unauthorised WLAN hosts connected to rogue access points. Simulation is employed for testing and the results of the test verify the effectiveness of the approach in detecting rogue access points in a heterogeneous network comprised of wireless and wired subnets.

In essence, using this solution, RAPs are detected from the anomalies of frequent trials to connect which are marked by an increase in request packets.

B. RAP Protection for Commodity Wi-Fi Networks

Ma et al [14] developed a hybrid framework for protecting Wi-Fi networks from RAPs. This framework is designed to monitor network activities, prevent events that could lead to the generation of RAPs, discover existing RAPs, and block unauthorised network access through RAPs. RAPs are automatically detected and located through the combination of distributed wireless media surveillance and a centralised wired end socket level traffic "fingerprinting" solution. The framework includes two major components: a distribution detection module (DDM) and a centralized detection module (CDM). The former can be used on access points as small plug-ins, while the latter is located at the gateway router of a local network. The framework works in conjunction with security protocols such as Wired Equivalency Protocol and Wireless Fidelity Protected Access, and it does not require any specialised hardware. Also, it can protect the wireless network from attackers using customised equipment and/or violating the IEEE802.11 standard and works consistently under various network configurations. The Distributed Detection Module consists of a passive wireless frame collector, a RAP pre-emption engine, and a RAP detection engine. The frame collector is responsible for gathering wireless traffic. The collected data is then passed to the pre-emption engine, where checks are performed in order to block various attacks. Finally, the data is analysed by the detection engine.

C. RAP Detection using Traffic Round Trip Time

Watkins et al [15] propose to use the round trip time (RTT) of network traffic to distinguish between wired and wireless nodes. The RTT coupled with a standard wireless AP authorisation policy allows the differentiation between wired nodes, authorised APs, and RAPs. The lower capacity and the higher variability in a wireless network can be used to effectively distinguish between wired and wireless nodes. This detection is not dependent on the wireless technology and is scalable. It stays valid as the capacity of wired and wireless links increase, and is not affected by the signal range of the RAPs.

RAP Detection Using Network Traffic Analysis [13] is likely to require a significant amount of processing speed and power. It is also anomaly-based which implies that there is a high risk of vulnerability at the inception of the system while it learns the patterns of anomalies. From the description of *RAP Protection for Commodity Wi-Fi Networks* [14] according to Ma et al, their proposed solution implies a certain level of complexity because of the different components and the protocols it uses. *RAP Detection using Traffic Round Trip Time* [15] introduces factors dependent on time and the physical topology of the network is also likely to be a factor. Beacon Frame Manipulation on the other hand is simple and fast. It is capable of protecting clients against mobile rogue access points which can be deployed in the form of smartphones, that is, SRAPs and is also suitable for detection of generic RAPs.

IV. THE BEACON FRAME

A. Structure of the Beacon Frame

A beacon frames is a type of management frame used by an access point to advertise its presence [17]. It contains all the information about the network and is constructed at the Media Access Control (MAC) sub layer of the data-link layer of the Open System Interconnection (OSI) model. It is also used to identify the access point's name and other features. The access point of a service set periodically transmits the beacon frame to establish and maintain the network [17][18]. Figure 2 shows the general structure of a wireless frame. The Frame Control, Duration, Destination Address, Source Address, Basic Service Set Identifier and Sequential Control fields make up the frame header. All frames, including the beacon frame have a frame header, frame body and the frequency check sum. The components of the frame body differ per type and function of the frame.

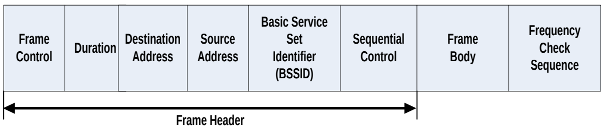

Figure 2: Beacon Frame

A typical beacon frame is about fifty (50) bytes long. The common frame header and cyclic redundancy checking (CRC) fields usually take up twenty-five (25) bytes of the frame. Just like other Open System Interconnection (OSI) layer 2 frames; the header includes source and destination MAC addresses as well as other information regarding the communications process [18]. The destination MAC address is always set to all ones, which is the broadcast MAC address. This forces all the other stations on that channel to receive and process each beacon frame. The CRC field within the beacon provides error detection functionality. Parameter sets are portions of the frame body which include information about specific signalling methods and setting values in these is optional.

The frame body of the beacon frame consists of two sections: mandatory fields and optional fields. The mandatory fields include the timestamp field which shows when the beacon was transmitted, the beacon interval field which handles when beacon frames are sent relative to each other, capability info field and the SSID field which is of a variable size. The timestamp field also helps with synchronisation with the client [18]. To synchronise, the receiving stations change their local clocks to match with the AP's clock. Figure 3 is an illustration of the mandatory fields of the beacon frame. These fields are always present in every beacon frame.

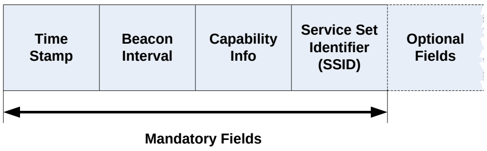

Figure 3: Mandatory fields of the Beacon Frame

The optional fields of the beacon frame body include the information elements namely the FH (Frequency Hopping spread spectrum) parameter set, DS (Direct Sequence spread spectrum) parameter set, CF (Contention-Free) parameter set IBSS (Independent Basic Service Set) parameter set and the TIM (Traffic Indication Map) field. These are illustrated in Figure 4. Each information element (IE) is made up of three fields: the Element ID (eid) field, Length field, and the Information field. Each IE is assigned a unique ID which fits in 1 octet. The Element ID uniquely identifies an information element [19]. The Length field specifies the length of the element-specific Information field and the Information field contains specific information for each element. There are 22 information elements which can be part of the beacon frame.

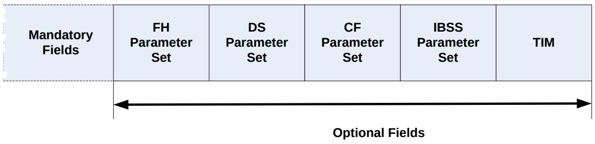

Figure 4: Optional fields of the Beacon Frame

B. Time Stamping

In computing, a timestamp is time assigned to a message or transaction. This time indicates the actual time the event occurred [19]. It usually gives this time in the form of date and time of day and is calibrated to fractions of a second for precision [20]. This precision makes it possible for networked devices to communicate effectively. Timestamps are usually used for logging events. When a station receives a beacon frame it uses the value in the timestamp field to update its local clock.

This process enables synchronization among all stations that are associated with the same access point [21]. Each beacon sent by an AP contains a timestamp. This timestamp is used by client stations to keep their clocks synchronized with the AP.

C. Beacon Frame Manipulation

In this paper we present a solution that employs time to generate a value that clients will use to authenticate APs. Time is extracted from the timestamp field of the beacon frame. The timestamp value is used by the client to update its local clock for synchronization [18]. It marks the time the beacon was sent. To perform beacon frame manipulation, an access point is set up using hostapd. Hostapd is a daemon which allows creation of software wireless access points. It works by generating the beacon frame template and then sending that for configuration to an actual beacon in the kernel. In the experiment, the timestamp value is passed through an exponential function to produce a single value referred to as the Authentic Access Point Value (AAPV) which is then embedded in the unused bits of the length field of the CF parameter set information element of the beacon frame. The CF Parameter Set information element, which is part of the optional fields of the beacon frame, shown in Figure 4 is included in access points that support contention-free operation to keep all stations ready and aware of contention-free operations [22, 23]. It is transmitted in beacons by access points that support contention-free operation. Contention-free operation supports applications that require near real-time service. The access point then broadcasts this special beacon and all clients belonging to the network seek to connect only to that AP. The idea of embedding information within a beacon frame is not new [22].

V. EXPERIMENT SETUP

A. The Access Point

Developing an access point offers the flexibility and customizability required for this experiment as compared to editing firmware of an already existing AP. The AP is created and configured using the hostapd.conf file which is a configuration file for hostapd. Table 1 shows the settings of the configured access point.

Property	**Setting**
Interface	wlan0
SSID	"stoprogue"
Mode	G
Channel	1

Table 1: Table of settings

Once the access point is setup, the beacon frame is manipulated using the beacon.c file found in hostapd. This file implements the beacon frame generation process [24]. The beacon.c file is modified and used to embed a value "y" in the unused bits of the length field of the CF parameter set information element. The value is derived using equation 2:

$$x = (hh)^3 + (mm)^2 + ss \quad (1)$$

$$y = \frac{1}{1+e^{-x}} \quad (2)$$

In equation 1, the variables used to calculate the value x are extracted from the timestamp field of the beacon frame where hh is the hours, mm minutes and ss seconds. The calculated value is then passed onto equation 2 to calculate y which is the AAPV. y is derived just before the beacon is sent and is stored in the CF parameter set information element.

```
#define WLAN_EID_CF_PARAMS 4
```

Code Listing 1:

The definition of the CF Parameter set information element is shown in code listing 1. The CF parameter set constitutes 8 bytes with the element ID taking up 1 byte, the length field taking up 1 byte and the information field taking up 6 bytes. As stated earlier, the length field has 5 unused bits which are used to place the dynamic AAPV.

```
static u8 * hostapd_eid_cf_params(struct hostapd_data
*hapd, u8 *eid, u8 *pos, u8 *end, int max_len)
{
        if(hapd->iface->current_mode == NULL ||
             hapd->iface->current_mode->mode !=
             HOSTAPD_MODE_IEEE80211G)
                  return eid;

        u8 *pos = eid;
        u8 *end = eid + max_len;
        int[]  I = y;

        os_memcpy(pos, eid, 3);

        pos += 8;

        pos++ = I;

        return eid;
}
```

Code Listing 2

Using the hostapd_ied_cf_params() function, the beacon frame is manipulated by inserting an AAPV (y) into the length field of the CF parameter set. Code Listing 2 is an illustration of this process. The interface mode is set to "g" (which is the mode for the wireless adapter being used for the experiment) and the element id (eid) of the CF parameter set is returned. The position pointer (pos) points to the element id of the CF parameter set and the size of the element is determined by the length of the eid and max-len. The integer array I stores the AAPV (y) in the length field of the information element. The eid is then copied into the pos variable with a length of 3 bits with the remaining 5 bits being skipped. The first 3 bits of the length field are already used and so they too are skipped such that pos is incremented by 8. y is placed in the next position.

B. The client

The client uses its clock time to calculate its own value of y (y') which is compared to the embedded y value in the CF parameter set of the received beacon frame. Code Listing 3 is seamlessly integrated to the polling drivers associated with the WLAN. If the embedded value (y) is the same as the one calculated by the client (y'), the function WLAN_CONNECT() is called. This function passes control to the WLAN drivers. If the values do not match, the client continues to poll for the appropriate beacon frame as illustrated in the POLL() function.

```
    int main(int argc, char *argv[])
    {

            /* checks for device */
            if(argc < 2){
                    printf("enter interface :\n");
                    return(0);
            }


            /* opens device */
            handle = pcap_open_live(dev, 1024, 0, 1024,
errbuf);
            if(handle == NULL){
                    printf("Couldn't open device %s\n",
errbuf);
                    return(1);
            }

            /* checking for AAPV */
    void got_packet(u_char *args, const struct pcap_pktlen
    *lenght, const int*packet){
    /* drop everything that does not contain the value "y" in
the length field */
    identified = FALSE;
            for (i = 0; i < 300; i++){
                    if(identified == FALSE && packet[9] ==
y){
                            identified = TRUE;
                    }
            }
            if(identified = TRUE){
                    WLAN_CONNECT();
            }
            else
                    POLL();
    }
```

Code Listing 3

The AAPV architecture is shown in Figure 5. Control to the WLAN drivers is passed via the AAPV polling application.

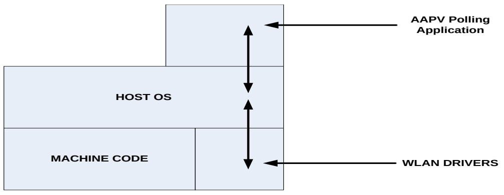

Figure 5: AAPV Architecture

VI. TESTING AND RESULTS

A wireless network consisting of an AAPV access point, a SRAP and fifteen (15) client machines is setup. Of the fifteen (15) clients, ten (10) have the AAPV application installed and the other five (5) use the default WLAN drivers. The AAPV access point and SRAP are both setup with the same SSID and connection password. The SRAP is Android-based and has PacketShark installed. Twenty (20) connection attempts were made per client to the transmitted beacons and the results were as shown in Table 2.

CLIENT	**SUCCESS (AAPV)**	**SUCCESS (SRAP)**	**BEACON FRAME DETECTION (AAPV)**	**BEACON FRAME DETECTION (SRAP)**
AAPV (10)	95%	0.05%	100%	100%
NON-AAPV (5)	0%	75%	100%	100%

Table 2: Connection attempts results

The 0.05% failure rate of the AAPV clients managing to connect to the SRAP was due to misconfiguration of the AAPV client program on the particular clients. Of the total connection attempts the AAPV enabled clients successfully connected 95% of the time to the AAPV access point and detected both the AAPV and the SRAP. The non-AAPV enabled clients failed completely to connect to the AAPV access point and managed to connect to the SRAP at 75% success rate.

VII. LIMITATIONS AND RECOMMENDATIONS

The proposed system uses time to produce a dynamic value. Generally, the values y and y' are likely to differ due to latency of the beacon frame. This difference is likely to increase if repeaters are used between the AP and the clients. Scalability is another factor likely to suffer as a result of latency in which case it is recommended that an error margin be incorporated in calculating y'. The use of Wi-Fi Protected Access (WPA) [25] is encouraged.

VIII. DISCUSSION AND CONCLUSION

The results of the research indicate that manipulating the beacon frame can stop clients from connecting to unauthorised access points including smartphone rogue access points. The beacon frame can be manipulated by using any free bits of any information element. The beacon frame still remains standard after manipulation and there is no need to alter the hardware. A deep understanding of how clients connect to access points is vital in this research and there is need to understand the frame exchange processes involved in establishing a connection.

REFERENCES

[1] William Enck and Patrick McDaniel, October 2008. Understanding Android's Security Framework. Systems and Internet Infrastructure Securities

[2] Burns, Jessie., Mobile Application Security on Andriod. 2009. Black Hat, USA

[3] Matthew Gast, 2002, 802.11 Wireless Networks: The Definitive Guide, O'Reilly, 2002

[4] Min-kyu Choi, Rosslin John Robles, Chang-hwa Hong and Tai-hoon Kim, Wireless Network Security: Vulnerabilities, Threats and Countermeasures, 2008. International Journal of Multimedia and Ubiquitous Engineering. Vol. 3, No. 3, July, 2008

[5] The Pew Research Center. Internet & American Life Project Cell Internet Use. 2012. Available from: http://www.pewinternet.org/Reports/2012/Cell-Internet-Use-2012.aspx. (Accessed 23 January 2013)

[6] Stalling, W., 2006. Cryptography and Network Security. 5th edition. Prentice Hall

[7] Parthip, S., 2012. 802.11 Sniffer Capture Analysis - Management Frames and Open Auth. Available from https://supportforums.cisco.com/docs/DOC-24651. (Accessed 27 January 2013)

[8] Geier, Jim, Identifying Rogue Access Points. 06 January 2006. Available from http://www.wi-fiplanet.com. (Accessed 13 December, 2012)

[9] Wexler, Joanie, Do we really need rogue AP protection?. November 15, 2004. Available from http://www.networkworld.com/newsletters/wireless/index.html. (Accessed 3 February 2013)

[10] Cisco Systems. 2006. Five steps to securing your wireless LAN and preventing wireless threats. Whitepaper, Copyright © 2006 Cisco Systems, Inc.

[11] Cogen, David., December 1, 2011. How to root the Samsung Galaxy (all versions). Available from http://theunlockr.com/2011/12/01/how-to-root-the-samsung-galaxy--all-versions/. (Accessed 25 July, 2012)

[12] John, A. 2011, What is rooting on Android. The advantages and disadvantages.

[13] Shetty, Sachin., Song, Min., Ma, Liran., (2007), Rogue Access Point Detection by Analyzing Network Traffic Characteristics, IEEE Conference Publications

[14] Ma, Liran., Teymonan, Amin Y., Cheng, Xinzhen (2008), A Hybrid Rogue Access Point Protection Framework for Commodity Wi-Fi Networks, IEEE INFOCOM 2008

[15] Watkins, L., Beyah, R., Corbett, C., A passive approach to Rogue Access Point Detection using Traffic Round Trip Time

[16] Vanderauwera, J., Bruinsma, L., Carlier, S. And Hassanmahomed, T., Compromising Wireless Security with Android, 31 December 2009.

[17] The MathWorks Inc, IEEE 802.11 WLAN – Beacon Frame, Available from www.mathworks.com. (Accessed 24 January 2013)

[18] GEIER, Jim, (n.d.). Beacons Revealed. Available from: www.wi-fiplanet.com. (Accessed 26 January, 2013)

[19] m. Rouse, September 2005, timestamp, http://whatis.techtarget.com/definition/timestamp

[20] Coleman, David. D and Westcott, David. A, CWNA: Certified Wireless Network Administrator Official Study Guide: Exam PW0-105, 2012, John Wiley & Sons

[21] http://www.wi-fiplanet.com/tutorials/article.php/1492071/80211-Beacons-Revealed.htm

[22] Gupta, V. and Rohil, M. K., Information Embedding in IEEE 802.11 Beacon Frame, National Conference on Communication Technologies & its impact on Next Generation Computing CTNGC 2012 Proceedings published by International Journal of Computer Applications® (IJCA)

[23] Hostapd, Available from http://en.hostapd.org/wiki/Hostapd#Jouni_Malinen.27s_hostapd

[24] hostapd, http://hostap.epitest.fi/

[25] Mitchelle Bradley, WPA - WI-Fi Protected Access, Available from: http://compnetworking.about.com/cs/wirelesssecurity/g/bldef_wpa.htm