

Vulnerability Risk Randomization Wireless Networks

Aditya K Sood , Mlabs SecNiche Security

2008© All Rights Reserved.

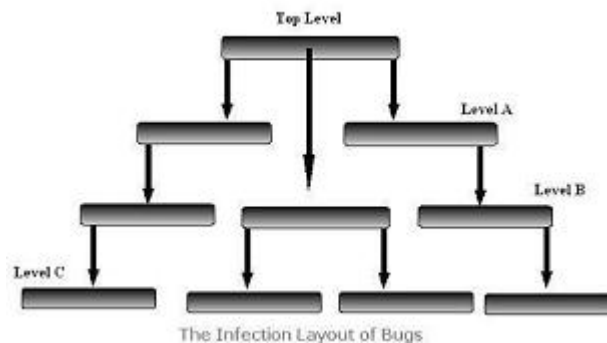
This document is a copyright work. SecNiche makes no representation or warranties, either express or implied by or with respect to anything in this document, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose or for any indirect special or consequential damages. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, photocopying, recording or otherwise, without prior written consent of SecNiche. While every precaution has been taken in the preparation of this publication, this publication and features described herein are subject to change without notice.

[1] Abstract:

This paper provides a reflection on the vulnerability scenario with reference to wireless system errors and various security vectors. The vulnerability risk randomization entirely depends on handling and control of security vectors. Despite of number of vulnerability assessment methodologies and deployment techniques, the bugs still continue to flourish. The inferences from various cases do still not suffice enough to thwart the bugs originating from the system. The vulnerability is always disseminated by post influential measures. The risk of vulnerability randomization is high from security perspective.. The security realm is based on bug existence and vulnerability patching. The induction of randomization factor in vulnerability finding has made the task onerous. The effects are directly proportional to vulnerability assessment which is becoming hard enough though. The randomization effect is due to versatility in security vectors and error handling. The error trapping mechanisms are inadequate to hinder the bug generation. The wireless vulnerabilities follow this pattern too.

[2] Vulnerability Risk Randomization in Wireless Networks:

Security vulnerabilities emanate from generation of errors. The errors play a critical role in security realm because they leverage a lot of information. Dissection of errors can contribute to circumventing the attacks. The tracing and logging of errors are very handy techniques for undertaking the working functionality of the applications. The purpose is to understand the vulnerability risk randomization concept. The applications or system services are prone to errors. This is because the applications are designed through code snippets generated by humans which are prone to errors. The coding discrepancies become the major source of exploitation of applications. Our security strategy do cover the post influential security steps but still loophole persists. The errors evolve at the top and goes on traversing through the application to reach the bottom. Hierarchical model of error generation is presented below:



This in depth means there are a number of cross reference elements and coupled modules that are being called simultaneously to process the same kind of request. So if any of the modules is error prone then you can imagine the error infection in the application. That's why it is always advised to undertake stress testing of applications prior to launch. But I think most of the cases this seldom happens. The application checks are vital for the stream oriented working of system. The enumeration is undertaken to have a look at the error infection mechanism. A very simplified outlined structure has been provided determining the infection layout of errors from top to bottom and how the error flows seamlessly from the parent element. The generation of errors affects the vulnerability vectors by making them randomized. For Example spoofing a MAC address procedure can be used as a launch pad for different attacks. The error occurred is at base level but it some what gets divergent there by raising the attack surface.

[3] Security Vectors:

The security vectors are more structural in their working approach. This layout explains security in terms of very definite vectors which will be applied mathematically to the realm of security to carve the new persistent standards. The security itself delves into a new paradigm incessantly as new technology is evolving along with the parameters. In order to understand this parametric layout, a detailed analysis is indispensable because core knowledge strengthens the hold on security parameters. The exploiting parameters assist in grappling with the technology more efficiently as these parameters define the infirmities persisting in the security domain and how it traverses out into the main stream thereby jeopardizing the whole specific domain.

There are myriads of security models, some of them become the benchmarks but some are adapted to meet the present yardsticks of security. Ignoring this will instigate the exploitation parameters to wreck the networks. The human learning is indispensable in the practical implementation of security to grapple with the transformations in the technology. Both the factors must be diffused in a definitive manner to dethrone the insecurities and instabilities inherent in the security domain. This works in both the ways as pros and cons are intrinsic to the technology making the technology realm very subtle. The unbridled pace of development has made the protection of security domain more critical. We will look at the security vector in detail, how to implement it in the mathematical way to understand the security domain more clearly.

The stress is on the security parameters that define the domain of security upon which the security vectors are defined. We are going to analyze the every single entity that has implications on the security domain. The Security Domain is defined as the set of parameters that affect the security. We define security domain as:

Sd = {Authentication, Authorization, Confidentiality, Integrity, Access Control}

The S (t) is called to be as the Security vector that relates to every single security parameter that is defined in the Security Domain (Sd).

S(t) => The Security Vector Function.
S => The Security Vector
t => The Stress Limit Of Security Parameters.

Integrating all of the parameters based on the domain parameters results in:

S(t) = S1(t)Confidentiality + S2(t)Integrity + S3(t)Authentication + S4(t)Authorization + S5(t)Access Control

So if we analyze this mathematical layout we can simply conclude that all the security parameters must have a stress limit. Let's look at the continuity of security vectors in the context of bug wars.

The continuity plays a crucial role in the security implications because the security should be continuous and impregnable. Let's look at the continuity of security vector. The basic considerations are:

=> S(t) parameter check should be defined.
=> If S(t) is clearly applicable for any type of stress limit t.

Let's undertake security entity (a) with stress limit time t1
Let's undertake security entity (b) with stress limit time t2

The security vector is continuous if

S(a) at t1 = S(b) at t2

This defines the security entity in equilibrium because after the time phases out the security vector are constant. This means the security parameters become hard enough to dethrone the encountered flaws. We will look at the CIA Model called as Confidentiality, Integrity and Availability Model.

The attack space is termed as the space in the security domain where the attacks are possible, overriding the present security constraints. Thus the attack space marginalizes the security domain. Let's look into it:

S(t) Security Vector function defined.

S1(t) at t=t1 with stress limit t1 after which attack occur in S1
S2(t) at t=t2 with stress limit t2 after which attack occur in S2
S3(t) at t=t3 with stress limit t3 after which attack occur in S3

So with full domain parametric:

S(t) = S1(t) at t=t1(i) + S2(t) at t=t2(j) + S3(t) at t=t3(k)

You can look very clearly i, j, k space directions are defined [1]. We set our base on the CIA model for attack space analysis:

i => Confidentiality , j => Integrity , k => Availability

The Attack space [1] is negligible provided the vector function is continuous at every time phase. This does not allow the attack parameters to have their space in the security domain and cause the attack. So S(t) is considered to be as continuous if and only if:

S1(t) at t=t1 , S1(t) at t=t2 , S1(t) at t=t3 are continuous.

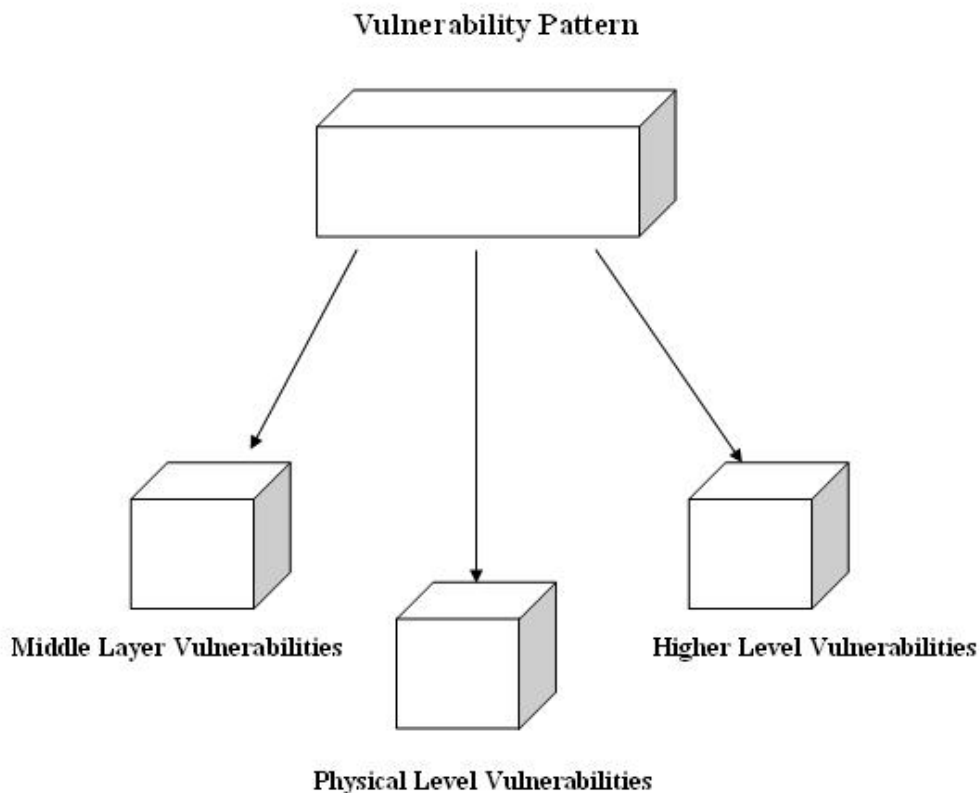
This holds that for whole security vector to be continuous, and then the individual security vectors must be continuous too. If this is true, the attack space tends to zero and the attack parameters are dethroned. The exploitation theory states that the exceptions generate the loopholes that infiltrate the system matrix.

Ingenious handling of exploitation vectors is indispensable to curb randomization risk . The errors must be filtered at the top and should not be allowed to traverse deeper. The software engineering model should be dynamic in order to combat these entities. The assessment methodologies should be sophisticated enough to ensure that the security is not compromised. At last, randomization effect is an outcome of unhandled errors. It can be avoided to some extent but not fully prevented.

[4] Wireless Vulnerabilities Pattern:

Every single technology is exposed to vulnerabilities. The wireless technologies inherit some weaknesses as well as vulnerability problems that lead to improper management of wireless networks. As a result of it wireless networks are prone to attacks. These vulnerabilities are not only software based but hardware based too. It has been noticed many times that a wireless hardware show inconsistencies. The driver used to operate those hardware's is malfunctioned that raises attack surface. It is always believe that wireless networks are prone to vulnerabilities when data is sent across the air that allows intruder or an attacker to capture data and use it in an illicit way. This is not even true always. The wireless network security is differentiated over this pattern with wired local area networks. If you see in wireless security the data is encrypted in lower layers prior to passing it to the higher layers.

Data can be captured at higher layers. If wired networks are seen then data can be captured at lower layers very easily. According to IEEE 802.11 the data is transmitted through infrared, Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DHSS). The physical layer functions around these data transmission mechanisms. The Wired Equivalence Privacy (WEP) does not provide End to End privacy but only provides security element between stations. It means it's a security implementation mechanism. For Example: The WEP only decrypts the body data there by leaving headers. The Extended Security Set Identifier (ESSID) is used for authentication among parties. Most of the time headers are not encrypted so an attacker can easily gather information of the parties under going communication. Let's look into vulnerability pattern diagram:

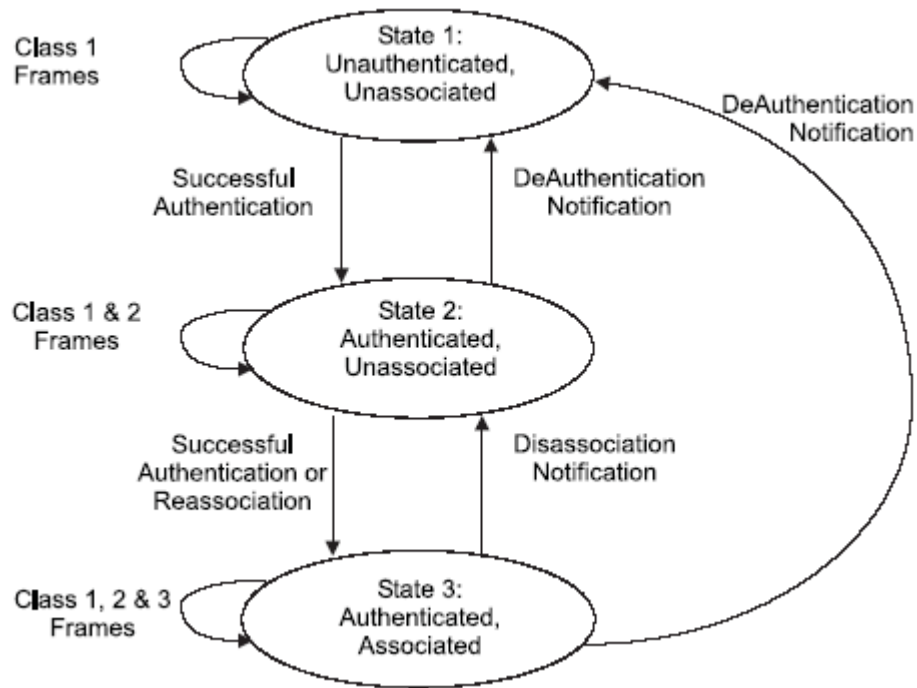


The classification is done on this pattern. The vulnerabilities are divided into three major parts. The physical layer vulnerabilities which are having interface with system hardware. The higher layer vulnerabilities are having interface with user. The middle layer flaws occur during data transmission between resultant parties. So the behavior defines the overall control of the wireless system from security point of view. We will discuss these patterns with certain examples to clarify the concept.

[4.1] Denial of Service

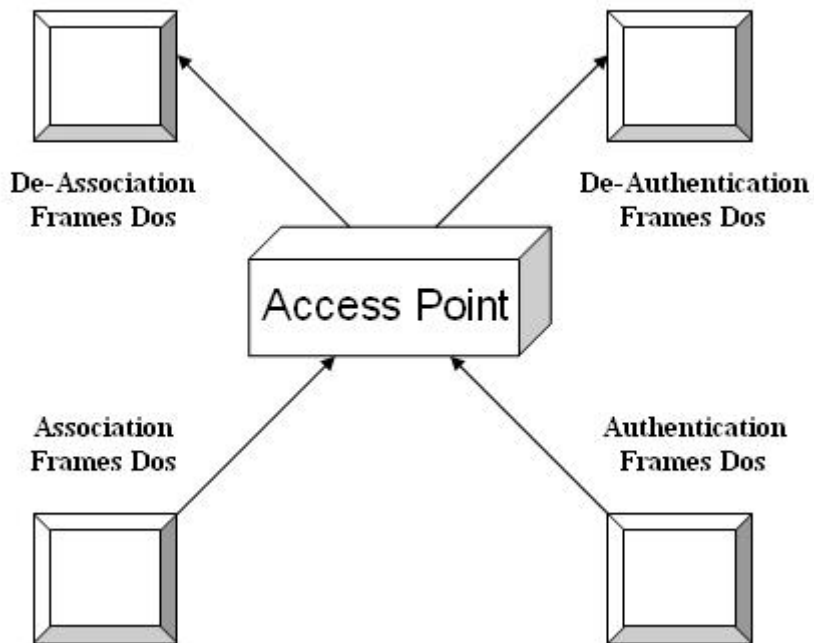
The very basic attack that kicks every single network is Denial of Service. These types of attack usually occur as a result exploitation of features of technology. The wireless networks are considered to be as open playground when beacons are sent continuously in ad-hoc [IBSS] and infrastructural [IBSS] mode.

The vulnerability vector comes to play when an attacker simply extracts information and try to manipulate the normal functioning of wireless network by injecting rogue traffic. These type of attacks are hard to combat but can be lowered to some extent by early detection mechanism. Again it can be analyzed the vulnerability vector gets randomized after some time which results in chain reaction of different type of attacks. Let's see the association and authentication mechanism.



The various ways that attacker adopt are:

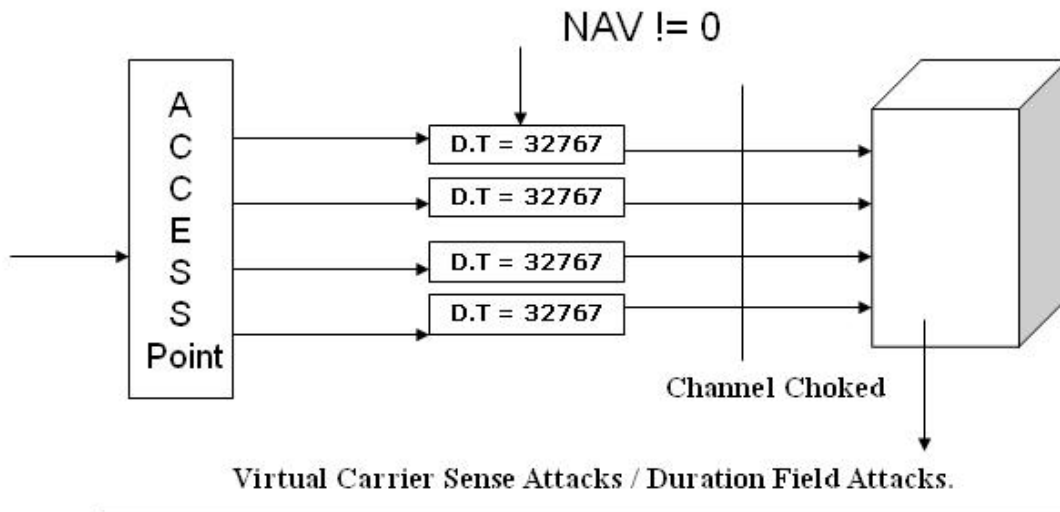
1. The frame mechanism is used to connect and disconnect a station from the access point. The frames are called as management frames. These management frames provide no authentication. So it becomes easy for an attacker to extract BSSID of an access point and flood the whole wireless network with de-authentication messages by sending messages to broadcast address i.e. [FF:FF:FF:FF:FF:FF]
2. The attacker can flood the network by sending number of disassociation frames to the access point which in turn legitimately brings the wireless network down. The disassociation and de-authentication DoS vulnerability follows the same pattern and vice versa for authentication and association frames. Let's see the structure:



3. The Extensible Authentication Protocol itself does not specify any authentication mechanism. It is used mainly for providing a general frame work for a connection to be authenticated. It has been noticed that access point allocates some resources when a connection is to be made by sending EAPoL start message for authentication. An attacker can easily exploit this functionality to launch denial of service attacks. The same point is applicable with EAPoL logoff messages.

[4.2] Network Allocation Vector Checks:

The wireless medium can be reserved by the access point by specifying a time value in frames that are going to be transmitted. The duration time sets the network allocation vector. This actually sets a limit on the device that is transmitting signals. Till the network allocation vector is not zero the device will not be able to send the packets respectively. This vulnerable factor is exploited by attackers a lot. The attackers regularly send frames with high duration value to block all channels. The largest value that can be specified is 32767. This will result in extensible denial of service and cause harm to network. Let's see:-



That's how NAV vulnerability is exploited.

[4.3] Physical Devices Malfunctioning:

The devices that comprise of wireless infrastructure show a vulnerable behavior in number of conditions. The vulnerability behavior varies from device to device. A lot of vulnerabilities have been published presenting this issue. The point to discuss is that even physical device can generate vulnerability pattern which in turn results in attack surface. So again a vulnerability risk is randomized from security point of view. The randomization vector can affect the security vector from any direction. It depends on the triggering of vulnerability.

[Router] [4.3.1]

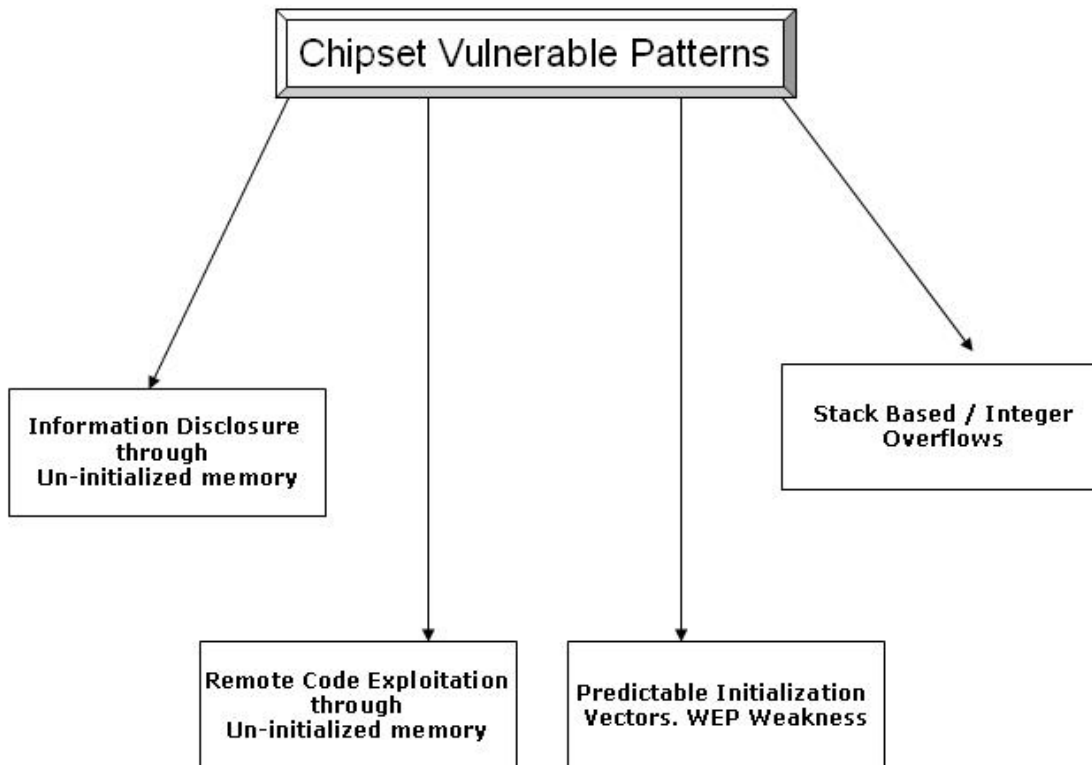
Number of routers used for wireless routing shows a problematic behavior. For Example Belkin wireless router is exposed to authentication vulnerability in which the authentication gets disabled when an administrator logs into web console. This results in information disclosure composed of access point information, router configuration etc. Further exploitation results in configuration of router from any client in the network.

[Access Point] [4.3.2]

It has been noticed that access points running IOS are vulnerable to memory exhaustion vulnerability. The physical device can be easily exploited by the attacker. In this a spoofed ARP messages are sent to management interface which consumes the IOS ARP table. For Example Cisco Aironet APs are of this kind. These types of attacks stop the traffic passage.

[Chipset Drivers] [4.3.3]

The wireless cards are functional through device driver programs. The drivers enable the physical device to have an interface with the system for reliable functioning. It has been taken into account vulnerable drivers leads to in-core vulnerabilities that can hijack overall system. It reflects the randomization affect of those vulnerabilities. Let's see:



a) In Linux systems there is a specific limit on the frame size to be sent. Some of drivers of the chipsets injects frame that is lower in size than the requisite frame. The left part is filled with uninitialized memory contents. As the chipset deals directly at kernel level, the memory contents of the kernel can be read through those frames. This issue has been seen in Ornicco and Prism 54 drivers. The attacker can easily exploit this context by sniffing traffic.

b) The chipset drivers are also vulnerable in handling malformed frames that can corrupt the internal memory which favors the remote command execution. The vulnerability vector is initiated by an attacker by sending a corrupt frame to the target thereby exploiting memory. On successful exploitation of the driver, commands can be executed in the context of system. The Intel Centrino Pro wireless chipsets drivers are prone to this type of issue. It is applicable to certain chipsets not all. But the vulnerability pattern persists. Even the MAC wireless drivers are vulnerable to this issue like Atheors adapter drivers.

c) The chipset drivers, on handling malformed frames can escalate the privileges of user. It means a normal user can gain the local kernel privileges through malformed frames. It means system undergo compromise by a vulnerability in chipset drivers. For Example: Intel PRO/Wireless 2100 chipset.

d) Some of the chipsets generate the initialization vectors that can be easily predictable. For Example for every single encryption of frame the IV is incremented by one. This process is repeated several times. An attacker on capturing number of signals can predict the working context. The IV's are generated by chipsets used in wireless networking. The Lucent Ornicco cards hold this issue.

e) The buffer overflow issue is another vulnerability pattern. It has been found that some of the chipset drivers running in operating systems are vulnerable to buffer overflows when a crafted beacon is sent. The prime problem is that stack is corrupted there by leading kernel level compromise. Typically called as stack overflows. It can be a result of integer overflows. As we know SSID parameter is to be passed in beacon. Some of the operating system shows vulnerability in generating probe response. The response is malformed as it is incorrectly processed. The Free BSD adheres to this issue. The Broadcom BCMWL5.SYS wireless device driver is vulnerable to a stack-based buffer overflow too.

[4.4] WEP Vulnerability Patterns:

The Wired Equivalent Privacy is one of the technique used in providing privacy to the user. Previously it was used heavily in networks. With the advent of time number of weaknesses found in it that leads to its downfall. Still it is used to some extent but with its existence, vulnerability definitely exists. There are number of issues encompassing the exploitation vectors which are provided below as:

a) The WEP introduces Integrity Check Value. It uses CRC32 mechanism to ensure integrity of data. When the message is encrypted, a CRC value is computed and appended at the end of the plain text and then encrypted with the required algorithm RC4. On the receiver end the frame payload is decrypted with same mechanism. The CRC 32 of the plain text is computed and is matched with the decrypted ICV. If a match occurs the data is in the right shape otherwise the data is tampered. The attacker can flip the arbitrary bits by encrypting the cipher text which in turn changes the decrypting text. The ICV,s are modified respectively. The messages are validated after this. It is considered to be as big weakness in WEP.

b) The WEP shows vulnerability layout in Open Authentication method. It is considered to be as a flaw of badly structured drivers for affected wireless cards. In this the attacker can easily trick the user agent if he is controlling the access point. Another issue is the presence of static WEP keys which are used again and again by the users. Numbers of users save the static WEP key as such which is a big security loop hole. The privacy bit matters a lot. If it is set to zero then once can have plaintext communication. The users will not be aware of this.

c) WEP is also vulnerable to injection attacks. It is easy to generate PRGA i.e. Pseudo Random Generation Algorithm. An attacker can easily compute PRGA by performing XOR operation on cipher packet with known plain text. It produces 8 bytes of data and with little knowledge of WEP keys arbitrary data can be injected into network by the attacker.

d) The collision between two Initialization Vectors is also a vulnerable vector in WEP. It simply reduces the integrity of message. Remember the IV is transmitted in plain text and is never re used. The IV is finite and some stations do not change the shared secret key till the IV is totally zero. Due to this factor if number of nodes increases in network a collision can occur simultaneously. The attacker can determine the plaintext of second frame if he has knowledge of plain text of first frame. It's again a big vulnerability factor.

e) Another possible vulnerability in WEP leads to Inverse Inductive Attack. In this an attacker can extract the plain text out of encrypted packet by using access point as decoder. This is due to weak replay mechanism.

f) The WEP keys are vulnerable to brute force and dictionary attacks too.

So that how exactly the vulnerability pattern exists.

Conclusion:

The vulnerability vector possesses randomization affect. The random nature shows the affected behavior on the system when that specific vulnerability is triggered. The error mechanism shows the top to bottom approach of error infection. The study clearly presents a hierarchical way starting from error generation and its resultant affect. It also possesses the structure of ingrained security vectors. The vulnerability patterns present the way vulnerabilities occur and its stringent nature. Looking at every single issue the randomization factor is there. It means the response of vulnerability can trigger flaw in any part of the working object. So a risk of randomization is always there. The vulnerability affects the security vectors. It is advisable to derive and develop highly efficient security techniques to stop the occurrence of vulnerability patterns.

References:

- [1]. <http://standards.ieee.org/getieee802/download/802.11-1999.pdf>
- [2]. http://sysnet.ucsd.edu/~bellardo/pubs/usenix-sec03_80211doshtml/aio.html
- [3]. <http://secunia.com/advisories/17601>
- [4]. <http://www.cisco.com/warp/public/707/cisco-sa-20060112-wireless.shtml>
- [5]. <http://www.securityfocus.com/bid/16217>
- [6]. <http://www.securityfocus.com/bid/15085/>
- [7]. <http://marc.theaimsgroup.com/?l=linux-netdev&m=113121660909437&w=2>
- [8]. <http://support.intel.com/support/wireless/wlan/sb/cs-010623.htm>
- [9]. <http://blackhat.com/html/bh-usa-06/bh-usa-06-speakers.html#ElIch>
- [10]. <ftp://ftp.freebsd.org/pub/FreeBSD/CERT/advisories/FreeBSD-SA-06:05.80211.asc>
- [11]. <http://www.wirelessve.org/entries/show/WVE-2006-0004>
- [12]. ftp://ftp.freebsd.org/pub/FreeBSD/CERT/patches/SA-06:05/80211_patch
- [13]. <http://www.wirelessve.org/entries/show/WVE-2006-0071>
- [14]. <http://projects.info-pull.com/mokb/MOKB-11-11-2006.html>
- [15]. <http://www.securitystartshere.net/page-vulns-wccd.htm>
- [16]. <http://www.toorcon.org/2005/slides/abittau/paper.pdf>
- [17]. <http://www.netstumbler.org/showthread.php?t=12489>