Tuesday, 25 June 2019

PENTESTER LAB NOTES:

Chapter 1: Linux and scripting

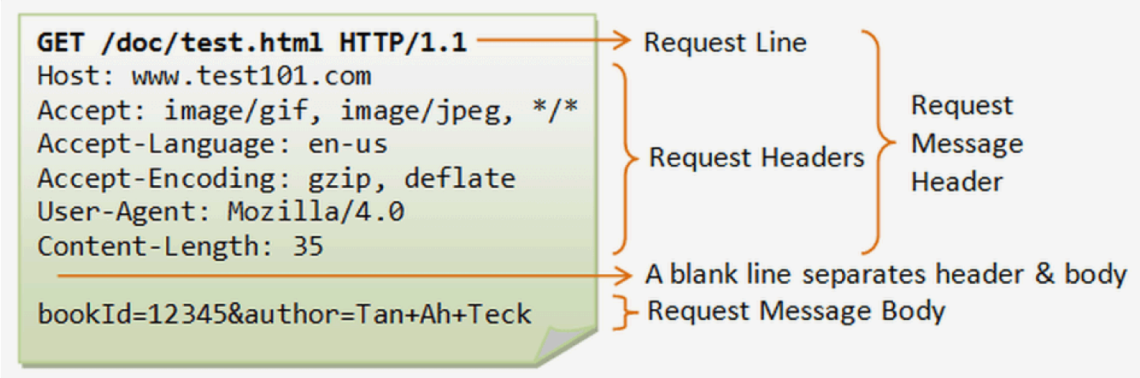
How HTTP Works

HTTP is an application layer protocol built on top of TCP that uses a client-server communication model. HTTP clients and servers communicate through request-and-response messages. The three main HTTP message types are GET, POST, and HEAD.

- HTTP GET messages sent to a server contain only a URL. Zero or more optional data parameters may be appended to the end of the URL. The server processes the optional data portion of the URL, if present, and returns the result (a web page or element of a web page) to the browser.

- HTTP POST messages place any optional data parameters in the body of the request message rather than adding them to the end of the URL.

- HTTP HEAD requests work the same as GET requests. Instead of replying with the full contents of the URL, the server sends back only the header information (contained inside the HTML section).



HTTP is what's called a stateless system. What this means is that unlike other file transfer protocols such as FTP, the HTTP connection is dropped after the request has been completed.

NOTE: 80 IS THE DEFAULT PORT FOR HTTP AND 443 IS THE DEFAULT FOR HTTPS.

Chapter 2 : HTTP

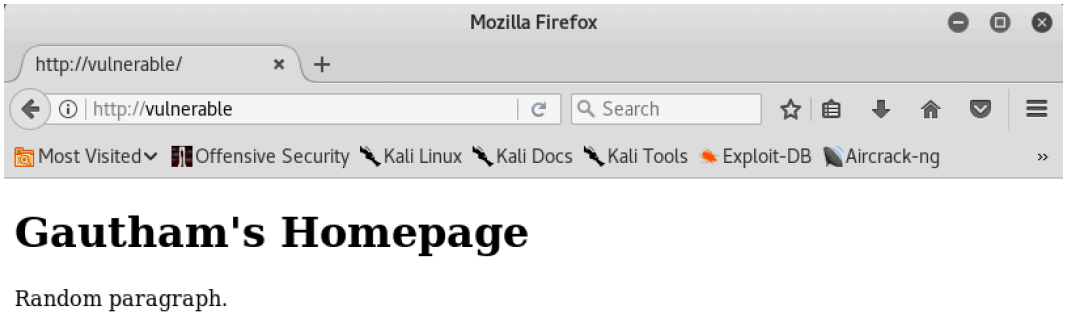
TCP/IP:

- TCP/IP specifies how data is exchanged over the internet by providing end-to-end communications that identify how it should be broken into packets, addressed, transmitted, routed and received at the destination.
- Collectively, the TCP/IP suite of protocols is classified as stateless, which means each client request is considered new because it is unrelated to previous requests. Being stateless frees up network paths so they can be used continuously.
- As indicated in the name, there are two layers to TCP/IP. The top layer, TCP, is responsible for taking large amounts of data, compiling it into packets and sending them on their way to be received by a fellow TCP layer, which turns the packets into useful information/data.
- The bottom layer, IP, is the locational aspect of the pair allowing the packets of information to be sent and received to the correct location.

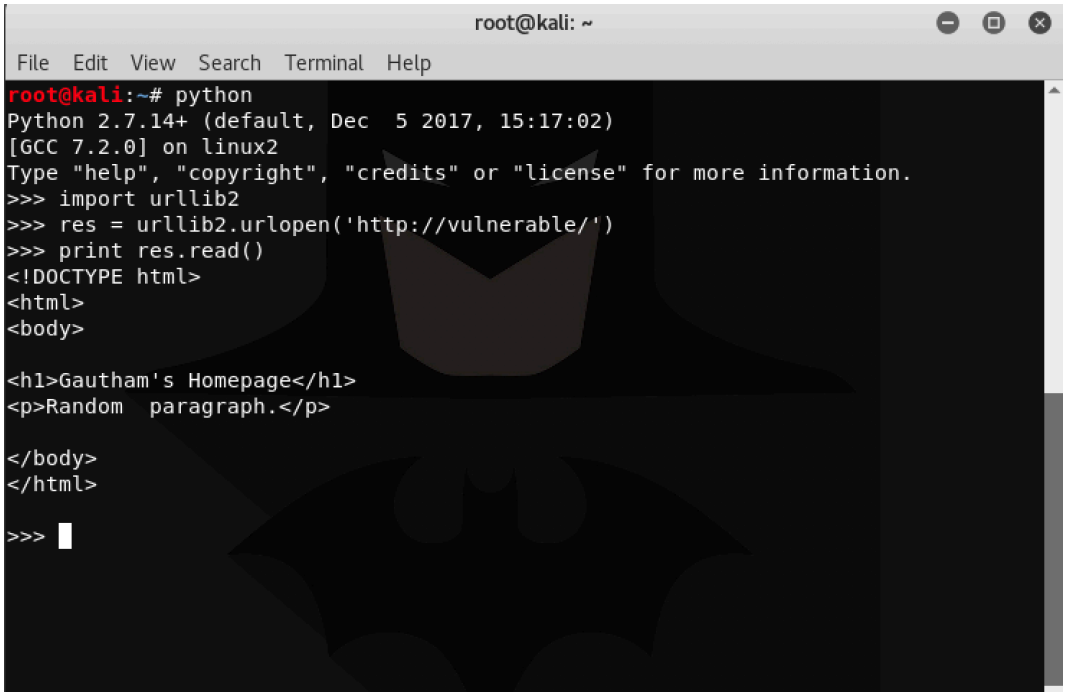
TRANSPORT LAYER SECURITY (TLS):

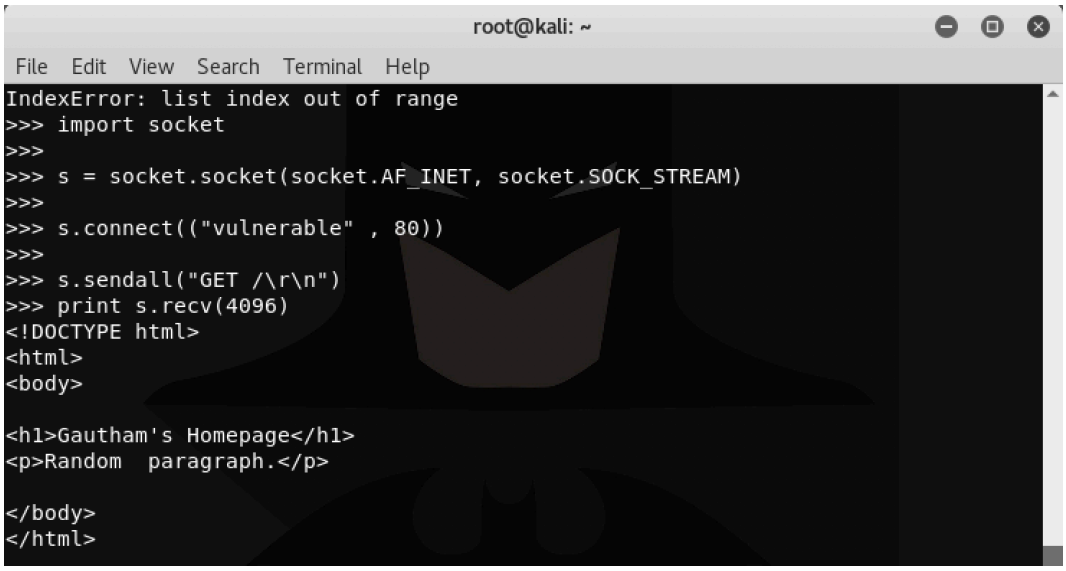
- TLS is a cryptographic protocol that provides end-to-end security of data sent between applications over the Internet. It is mostly familiar to users through its use in secure web browsing, and in particular the padlock icon that appears in web browsers when a secure session is established.
- TLS evolved from Secure Socket Layers (SSL) which was originally developed by Netscape Communications Corporation in 1994 to secure web sessions.
- With symmetric cryptography, data is encrypted and decrypted with a secret key known to both sender and recipient; typically 128 but preferably 256 bits in length (anything less than 80 bits is now considered insecure). Symmetric cryptography is efficient in terms of computation, but having a common secret key means it needs to be shared in a secure manner.
- Asymmetric cryptography uses key pairs – a public key, and a private key. The public key is mathematically related to the private key, but given sufficient key length, it is computationally impractical to derive the private key from the public key. This allows the public key of the recipient to be used by the sender to encrypt the data they wish to send to them, but that data can only be decrypted with the private key of the recipient. This is used by TLS for generation of session keys for every session.
- A Certificate Authority (CA) is an entity that issues digital certificates conforming to the ITU-T's X.509 standard for Public Key Infrastructures (PKIs). Digital certificates certify the public key of the owner of the certificate (known as the subject), and that the owner controls the domain being secured by the certificate.

Hands on : Find the initial homepage : /var/www/html/index.html and change the original html page.
Add the line 127.0.0.1 vulnerable in the /etc/hosts file.

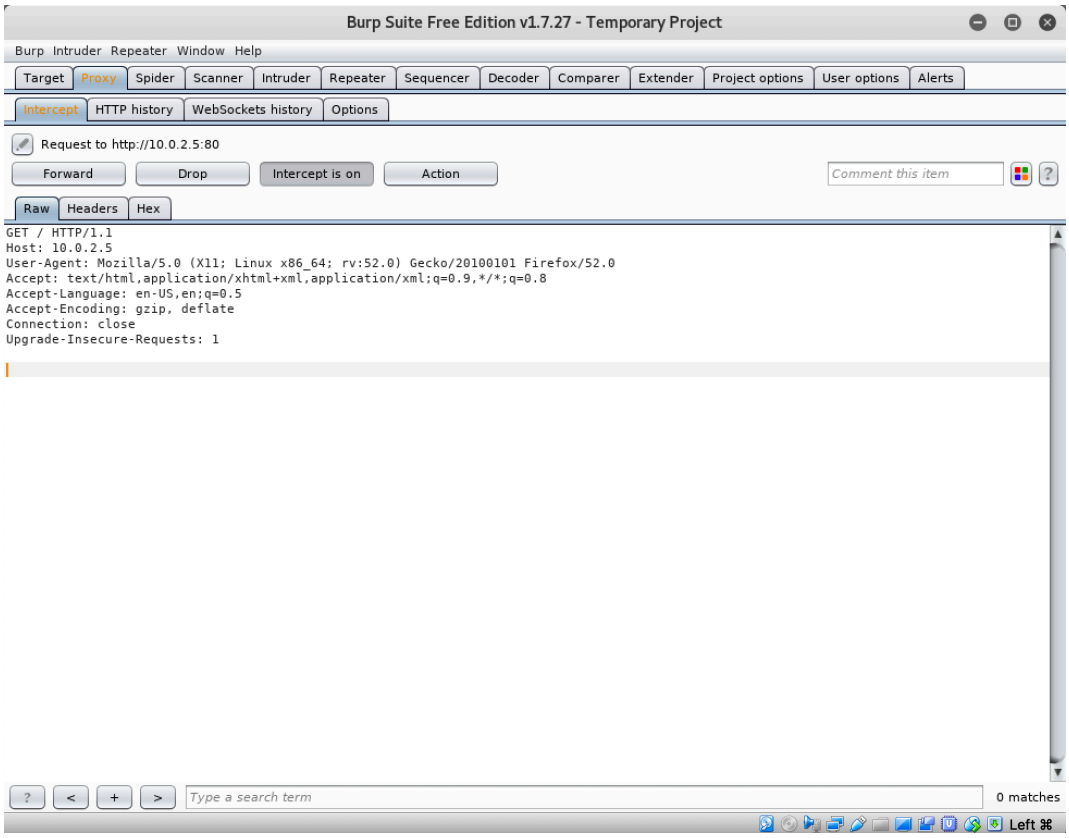


Hands on : Write a python script using the urllib2 and the socket library to retrieve html data from the url.





Hands on : Use burp suite



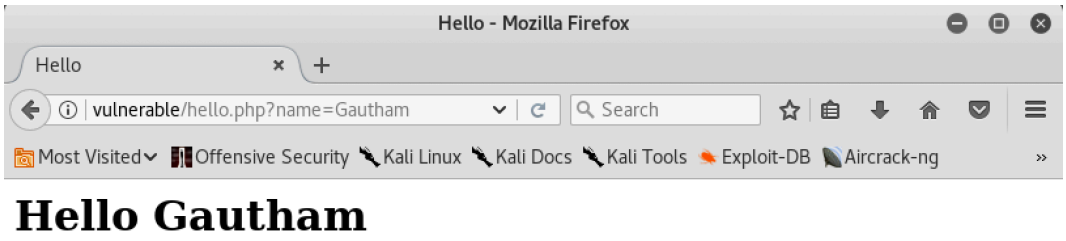
Chapter 3 : PHP and DNS

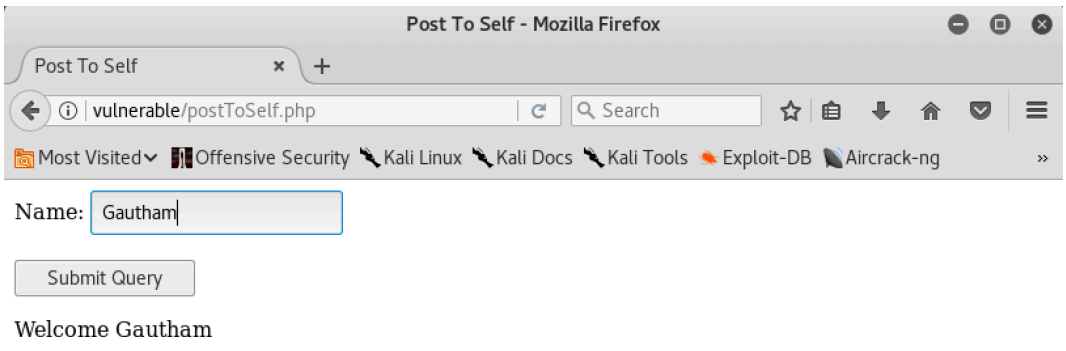
Virtual hosting: Virtual hosting is a technology that allows to have multiple domain names on a single physical server (and of course treat them differently).This allows one server to share its resources, such as memory and processor cycles, without requiring all services provided to use the same host name.

There are two principal kinds of virtual hosting:

- Name based: Different domain names point to the same IP. When the client makes a requests it must use the Host header, so the web server acts accordingly. One of the biggest problem of using Name Based virtual hosting is the use of SSL. The problem exists because the web server must know which certificate present to the client before data are sent, but the only thing that discriminate two virtual hosted site is the Host header which is part of the data!
- IP Based: Different names point to different IPs, but they all belong to the physical server (es multiple network interfaces)

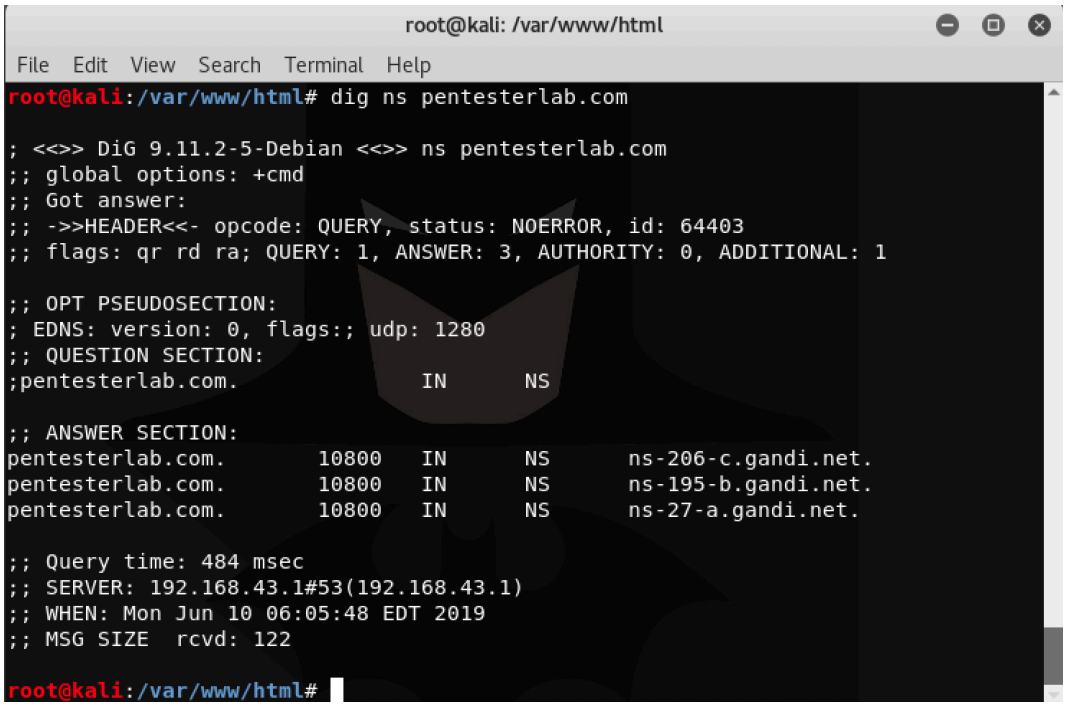
Hands on : make a php script to display the argument given in the url.

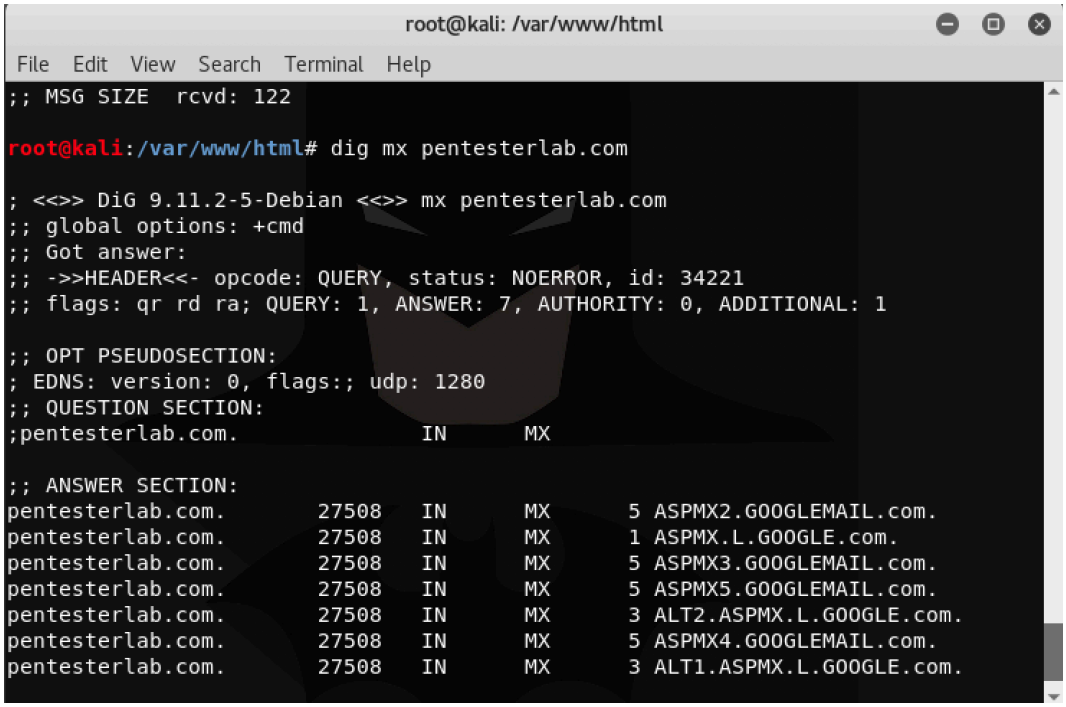




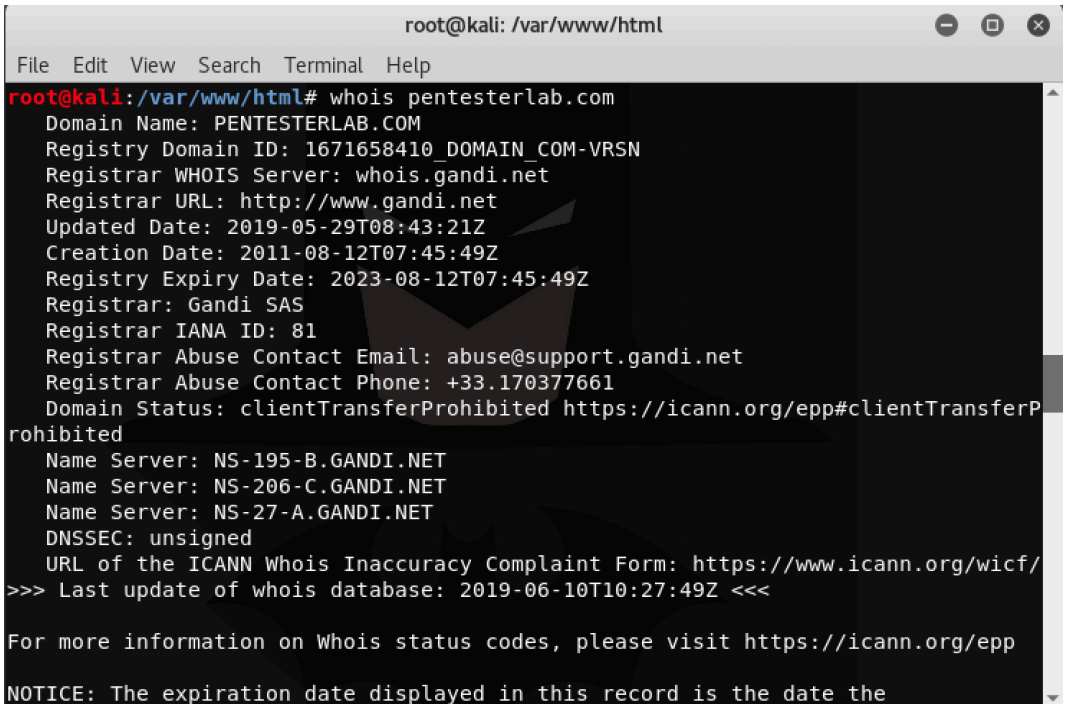
DNS:
The Domain Name Systems (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DN S translates domain names to IP addresses so browsers can load Internet resources.

Hands on : Dig ns for mail servers and dig mx for name servers.



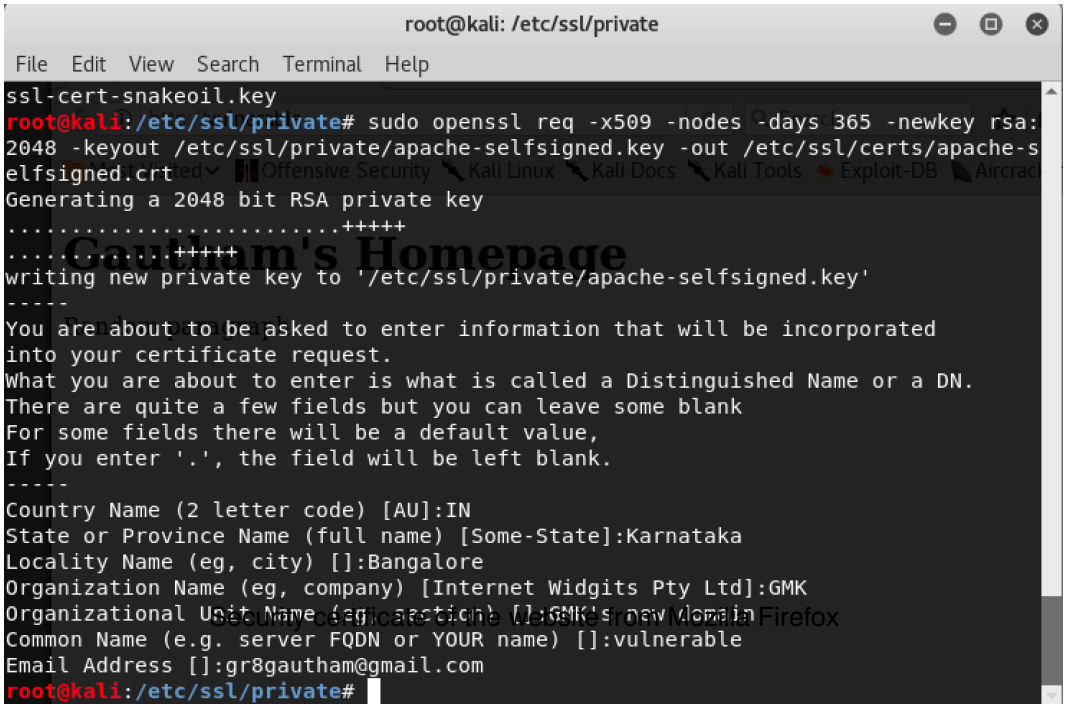


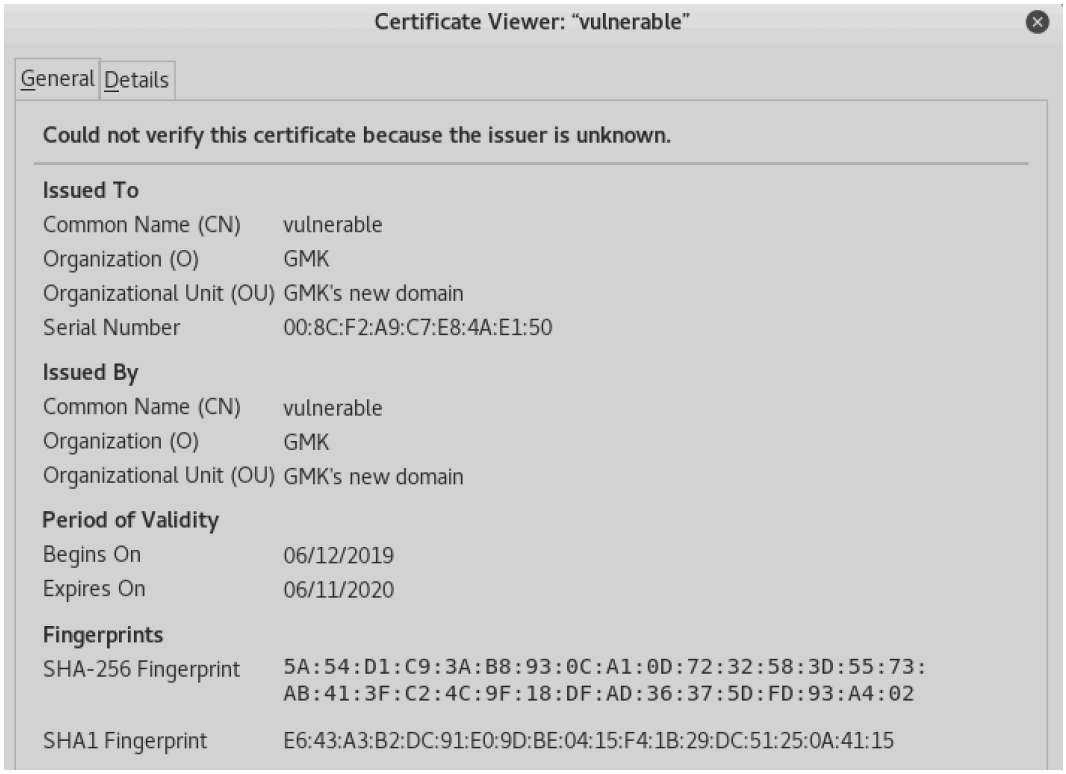
Hands on : Find information about a website using the whois DNS lookup.



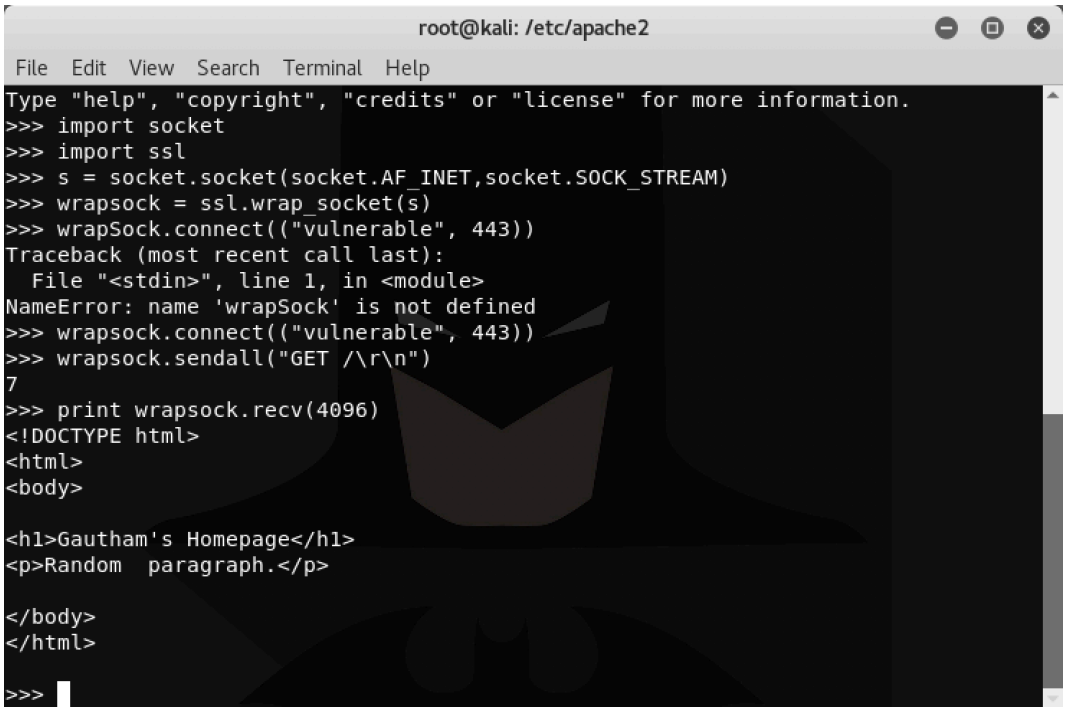
Chapter 4 : SSL and TLS

Hands on : Make a self signed SSL certificate to convert the ‘vulnerable’ domain from http to https. Use this tutorial - https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-16-04





Hands on : write a ssl client using a http library and a socket to retrieve html from the created webpage.



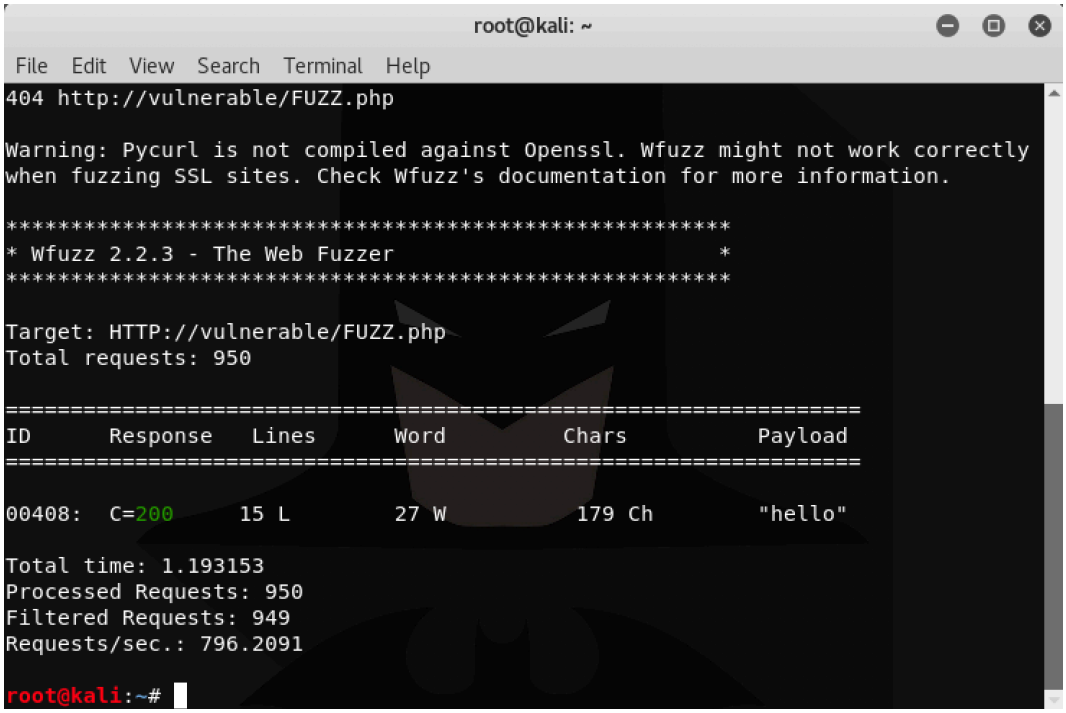

```python
#!/usr/bin/python
# HTTPS Client with Header Info

import httplib

# connect to HTTPS server, send GET request, receive response
h = httplib.HTTPSConnection('vulnerable')
h.request('GET', '')
r = h.getresponse()

# get and print HTTP header response
rh = r.getheaders()
print 'Header:'
for i in rh:
    print i[0], ':', i[1]
print '\n'

# get and print content of response
rr = r.read()
print 'Content:'
print rr
print '\n'

# close HTTP connection to server
h.close()
```

Chapter 5 : SQLi and Local file include

Multipurpose Internet Mail Extensions (MIME) is an Internet standard that extends the format of email to support:

- Text in character sets other than ASCII
- Non-text attachments: audio, video, images, application programs etc.
- Message bodies with multiple parts
- Header information in non-ASCII character sets

Virtually all human-written Internet email and a fairly large proportion of automated email is transmitted via SMTP in MIME format.



Website fingerprinting -
A lot of information can be retrieved by connecting to the web application using netcat or telnet:
$ telnet vulnerable 80
Where:

- vulnerable is the hostname or the IP address of the server;
- 80 is is the TCP port used by the web application (80 is the default value for HTTP).

By sending the following HTTP request:
GET / HTTP/1.1 - request for the root page of the server i.e index.html
Host: vulnerable

It is possible to retrieve information on the version of PHP and the web server used just by observing the HTTP headers sent back by the server:
HTTP/1.1 200 OK
Date: Thu, 24 Nov 2011 04:40:51 GMT
Server: Apache/2.2.16 (Debian)

X-Powered-By: PHP/5.3.3-7+squeeze3
Vary: Accept-Encoding
Content-Length: 1335
Content-Type: text/html

Here the application is only available via HTTP (nothing is running on the port 443). If the application was only available via HTTPs, telnet or netcat would not be able to communicate with the server, the tool openssl can be used:
$ openssl s_client -connect vulnerable:443
Where:

- vulnerable is the hostname or the IP address of the server;
- 443 is is the TCP port used by the web application (443 is the default value for HTTPs).

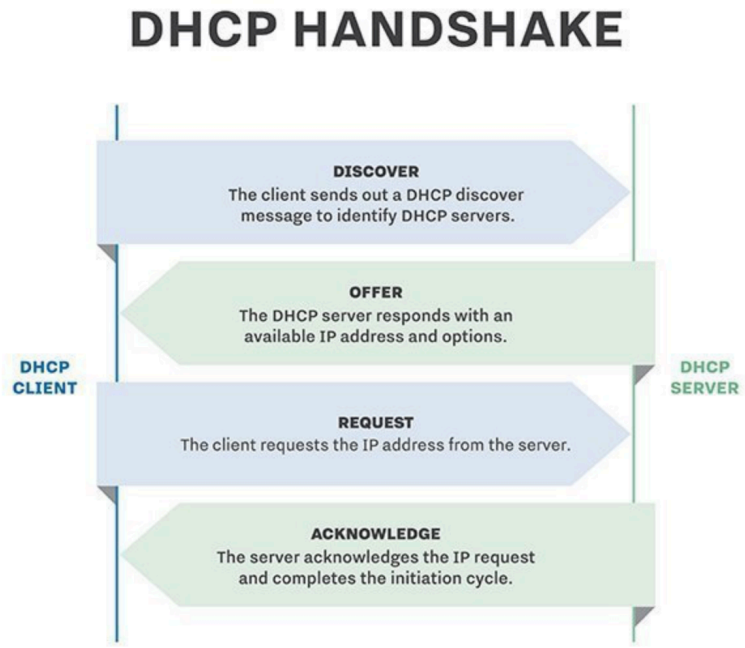
Using an application such as Burp Suite (http://portswigger.net/) set up as a proxy makes it easy to retrieve the same information.

Refer - https://www.owasp.org/index.php/Fingerprint_Web_Server_(OTG-INFO-002)
https://pentesterlab.com/exercises/from_sqli_to_shell/course
https://pentesterlab.com/exercises/php_include_and_post_exploitation/course

Chapter 6 : More SQL injections

Antisec movement-
The Anti Security Movement (also written as antisec and anti-sec) is a movement opposed to the computer security industry. Antisec is against full disclosure of information relating to
software vulnerabilities, exploits, exploitation techniques, hacking tools, attacking public outlets and distribution points of that information. The general thought behind this is that the computer security industry uses full disclosure to profit and develop scare-tactics to convince people into buying their firewalls, anti-virus software and auditing services.

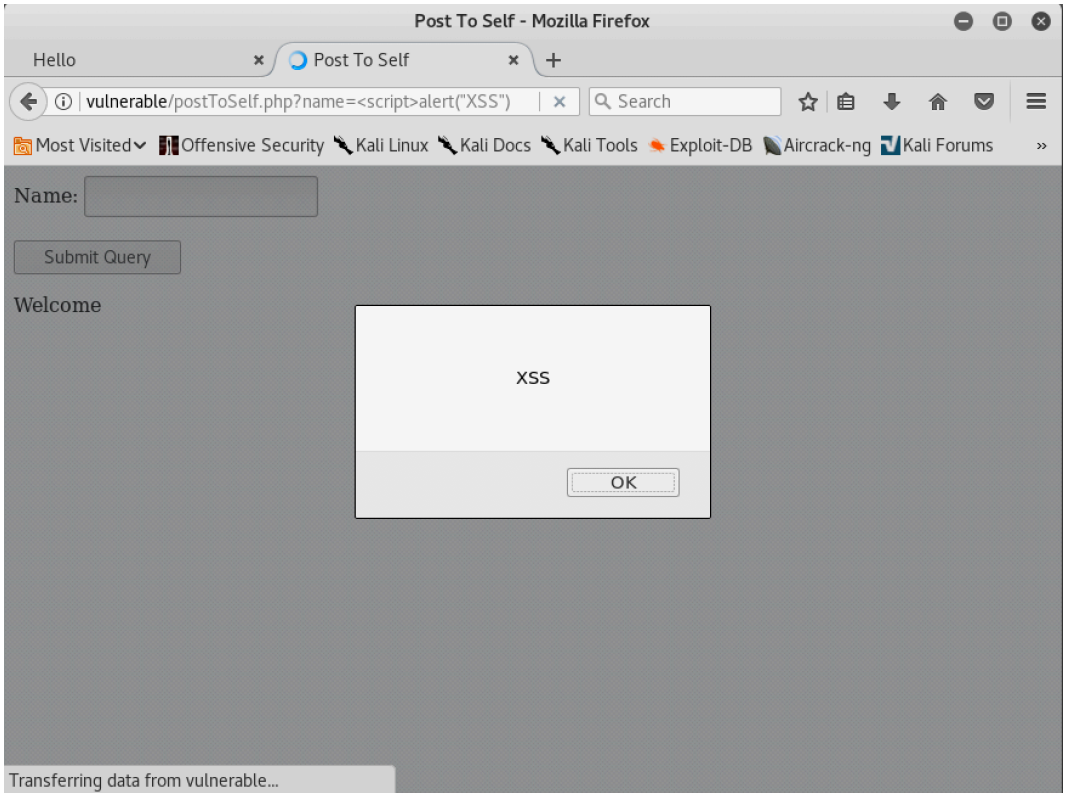
DHCP -
DHCP (Dynamic Host Configuration Protocol) is a network management protocol used to dynamically assign an Internet Protocol (IP) address to any device, or node, on a network so they can communicate using IP. DHCP automates and centrally manages these configurations rather than requiring network administrators to manually assign IP addresses to all network devices. DHCP can be implemented on small local networks as well as large enterprise networks.



RFC -
A Request for Comments (RFC) is a formal document drafted by the Internet Engineering Task Force (IETF) that describes the specifications for a particular technology. When an RFC is ratified, it becomes a formal standards document. RFCs were first used during the creation of the ARPANET protocols that came to establish what became today's Internet.

FTP - (1971)
FTP is a client-server protocol that relies on two communications channels between client and server: a command channel for controlling the conversation and a data channel for transmitting file content. Clients initiate conversations with servers by requesting to download a file. Using FTP, a client can upload, download, delete, rename, move and copy files on a server. A user typically needs to log on to the FTP server, although some servers make some or all of their content available without login, also known as anonymous FTP.

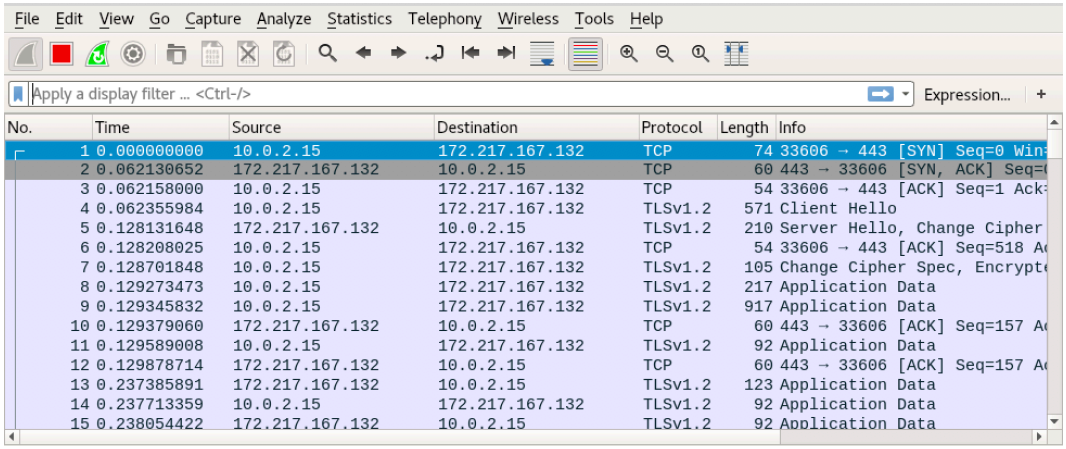
Hands on : testing vulnerabilities on the created website



Chapter 7 : FTP and Traffic analysis

Phrack-
Phrack is an ezine written by and for hackers, first published November 17, 1985.[1] Described by Fyodor as "the best, and by far the longest running hacker zine,"[2] the magazine is open for contributions by anyone who desires to publish remarkable works or express original ideas on the topics of interest. It has a wide circulation which includes both hackers and computer security professionals
Refer : http://phrack.org

Hands on : follow the tcp stream using wireshark and analyse the packets sent/received.



Hands on: Install FTP server and write an FTP client.

```
>>> from ftplib import FTP
>>> ftp = FTP('ftp.debian.org')     # connect to host, default port
>>> ftp.login()                     # user anonymous, passwd anonymous@
'230 Login successful.'
>>> ftp.cwd('debian')               # change into "debian" directory
>>> ftp.retrlines('LIST')           # list directory contents
-rw-rw-r--    1 1176     1176         1063 Jun 15 10:18 README
...
drwxr-sr-x    5 1176     1176         4096 Dec 19  2000 pool
drwxr-sr-x    4 1176     1176         4096 Nov 17  2008 project
drwxr-xr-x    3 1176     1176         4096 Oct 10  2012 tools
'226 Directory send OK.'
>>> ftp.retrbinary('RETR README', open('README', 'wb').write)
'226 Transfer complete.'
>>> ftp.quit()
```

Chapter 8 : Linux review and code execution

Iptables -
iptables is a user-space utility program that allows a system administrator to configure the tables provided by the Linux kernel firewall (implemented as different Netfilter modules) and the chains and rules it stores. Different kernel modules and programs are currently used for different protocols; iptables applies to IPv4, ip6tables to IPv6, arptables to ARP, and ebtables to Ethernet frames.

Internet Control Message Protocol -
Since IP does not have an inbuilt mechanism for sending error and control messages. It depends on Internet Control Message Protocol(ICMP) to provide an error control. It is used for reporting errors and management queries. It is a supporting protocol and used by networks devices like routers for sending the error messages and operations information.

Refer : https://pentesterlab.com/exercises/linux_host_review/course
https://pentesterlab.com/exercises/cve-2012-1823/course

Chapter 9 : HTTP server and firewall

Nmap-
Nmap (Network Mapper) is a free and open-source network scanner created by Gordon Lyon (also known by his pseudonym Fyodor Vaskovich).[3] Nmap is used to discover hosts and services on a computer network by sending packets and analyzing the responses.
Nmap provides a number of features for probing computer networks, including host discovery and service and operating system detection. These features are extensible by scripts that provide more advanced service detection,[4] vulnerability detection,[4] and other features. Nmap can adapt to network conditions including latency and congestion during a scan.

Setuid-
setuid and setgid (short for "set user ID" and "set group ID") are Unix access rights flags that allow users to run an executable with the permissions of the executable's owner or group respectively and to change behaviour in directories. They are often used to allow users on a computer system to run programs with temporarily elevated privileges in order to perform a specific task. While the assumed user id or group id privileges provided are not always elevated, at a minimum they are specific.

Hands on : Write a http server
Refer : https://www.afternerd.com/blog/python-http-server/

```
import http.server
import socketserver

PORT = 8080
Handler = http.server.SimpleHTTPRequestHandler

with socketserver.TCPServer(("", PORT), Handler) as httpd:
    print("serving at port", PORT)
    httpd.serve_forever()
```

<should be saved in the same directory as index.html>

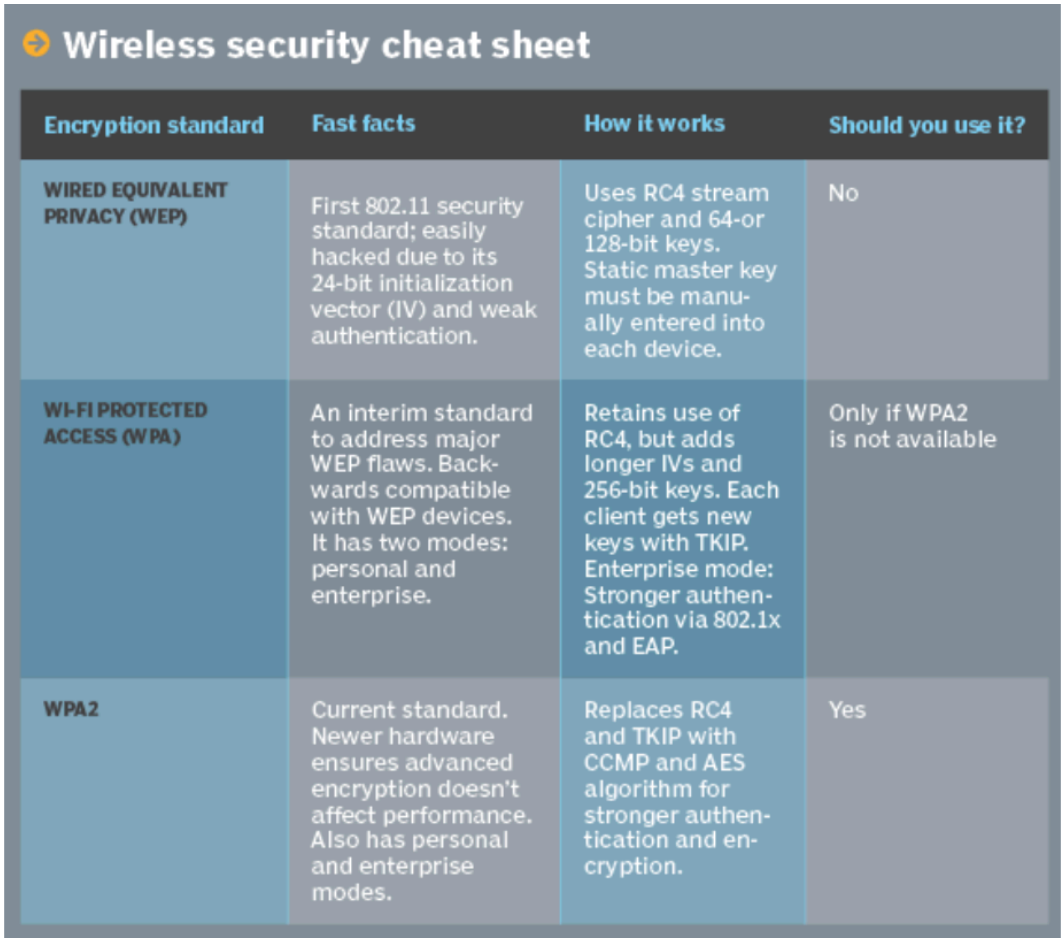
Chapter 10 : Nmap and crypto attacks

Wireless Fidelity (WiFi)
WiFi is a technology that uses radio waves to provide network connectivity. A WiFi connection is established using a wireless adapter to create hotspots - areas in the vicinity of a wireless router that are connected to the network and allow users to access internet services. Once configured, WiFi provides wireless connectivity to your devices by emitting frequencies between 2.4GHz - 5GHz, based on the amount of data on the network.

Wired Equivalency Privacy (WEP)
Developed in the late 1990s as the first encryption algorithm for the 802.11 standard, WEP was designed with one main goal in mind: to prevent hackers from snooping on wireless data as it was transmitted between clients and access points (APs). From the start, however, WEP lacked the strength necessary to accomplish this.
WEP uses the RC4 stream cipher for authentication and encryption. The standard originally specified a 40-bit, preshared encryption key -- a 104-bit key was later made available after a set of restrictions from the U.S. government was lifted. The key must be manually entered and updated by an administrator.
The key is combined with a 24-bit initialization vector (IV) in an effort to strengthen the encryption. However, the small size of the IV increases the likelihood that keys will be reused, which, in turn, makes them easier to crack. This characteristic, along with several other vulnerabilities -- including problematic authentication mechanisms -- makes WEP a risky choice for wireless security.

Wi-Fi Protected Access (WPA)
The numerous flaws in WEP revealed the urgent need for an alternative, but the deliberately slow and careful processes required to write a new security specification posed a conflict. In response, in 2003, the Wi-Fi Alliance released WPA as an interim standard, while the Institute of Electrical and Electronics Engineers (IEEE) worked to develop a more advanced, long-term replacement for WEP.
WPA has discrete modes for enterprise users and for personal use. The enterprise mode, WPA-EAP, uses more stringent 802.1x authentication with the Extensible Authentication Protocol, or EAP. The personal mode, WPA-PSK, uses preshared keys for simpler implementation and management among consumers and small offices. Enterprise mode requires the use of an authentication server.
Although WPA is also based on the RC4 cipher, it introduced several enhancements to encryption -- namely, the use of the Temporal Key Integrity Protocol (TKIP). The protocol contains a set of functions to improve wireless LAN security: the use of 256-bit keys, per-packet key mixing -- the generation of a unique key for each packet -- automatic broadcast of updated keys, a message integrity check, a larger IV size (48 bits) and mechanisms to reduce IV reuse.
WPA was designed to be backward-compatible with WEP to encourage quick, easy adoption. Network security professionals were able to support the new standard on many WEP-based devices with a simple firmware update. This framework, however, also meant the security it provided was not as robust as it could be.
Wi-Fi Protected Access 2 (WPA2)
As the successor to WPA, the WPA2 standard was ratified by the IEEE in 2004 as 802.11i. Like its predecessor, WPA2 also offers enterprise and personal modes.

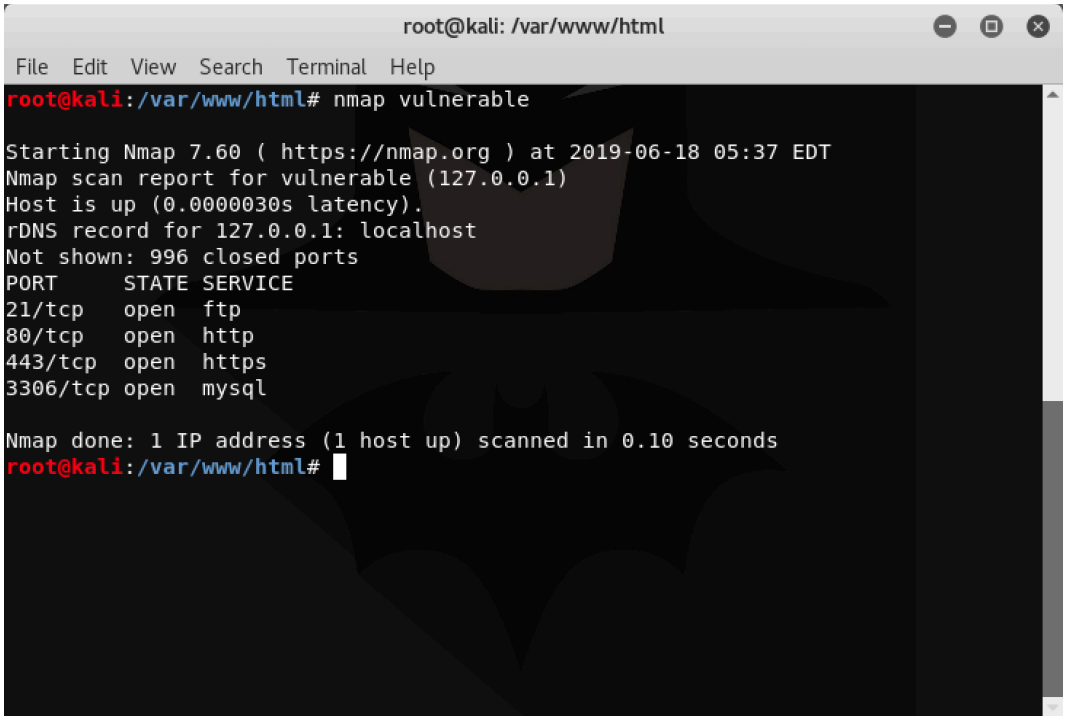
Although WPA2 still has vulnerabilities, it is considered the most secure wireless security standard available.
WPA2 replaces the RC4 cipher and TKIP with two stronger encryption and authentication mechanisms: the Advanced Encryption Standard (AES) and Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP), respectively. Also meant to be backward-compatible, WPA2 supports TKIP as a fallback if a device cannot support CCMP.
Developed by the U.S. government to protect classified data, AES is composed of three symmetric block ciphers. Each encrypts and decrypts data in blocks of 128 bits using 128-, 192- and 256-bit keys. Although the use of AES requires more computing power from APs and clients, ongoing improvements in computer and network hardware have mitigated performance concerns.
CCMP
protects data confidentiality by allowing only authorized network users to receive data, and it uses cipher block chaining message authentication code to ensure message integrity.
WPA2 also introduced more seamless roaming, allowing clients to move from one AP to another on the same network without having to reauthenticate, through the use of Pairwise Master Key caching or preauthentication.



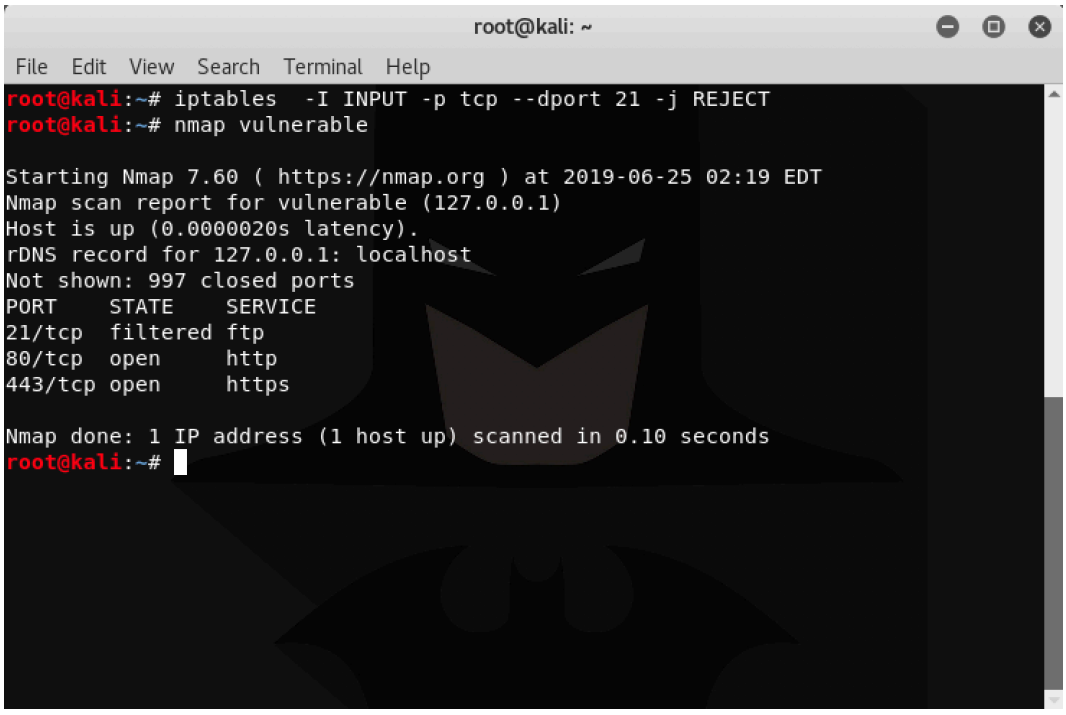
Wireless security cheat sheet

Encryption standard	Fast facts	How it works	Should you use it?
WIRED EQUIVALENT PRIVACY (WEP)	First 802.11 security standard; easily hacked due to its 24-bit initialization vector (IV) and weak authentication.	Uses RC4 stream cipher and 64-or 128-bit keys. Static master key must be manually entered into each device.	No
WI-FI PROTECTED ACCESS (WPA)	An interim standard to address major WEP flaws. Backwards compatible with WEP devices. It has two modes: personal and enterprise.	Retains use of RC4, but adds longer IVs and 256-bit keys. Each client gets new keys with TKIP. Enterprise mode: Stronger authentication via 802.1x and EAP.	Only if WPA2 is not available
WPA2	Current standard. Newer hardware ensures advanced encryption doesn't affect performance. Also has personal and enterprise modes.	Replaces RC4 and TKIP with CCMP and AES algorithm for stronger authentication and encryption.	Yes

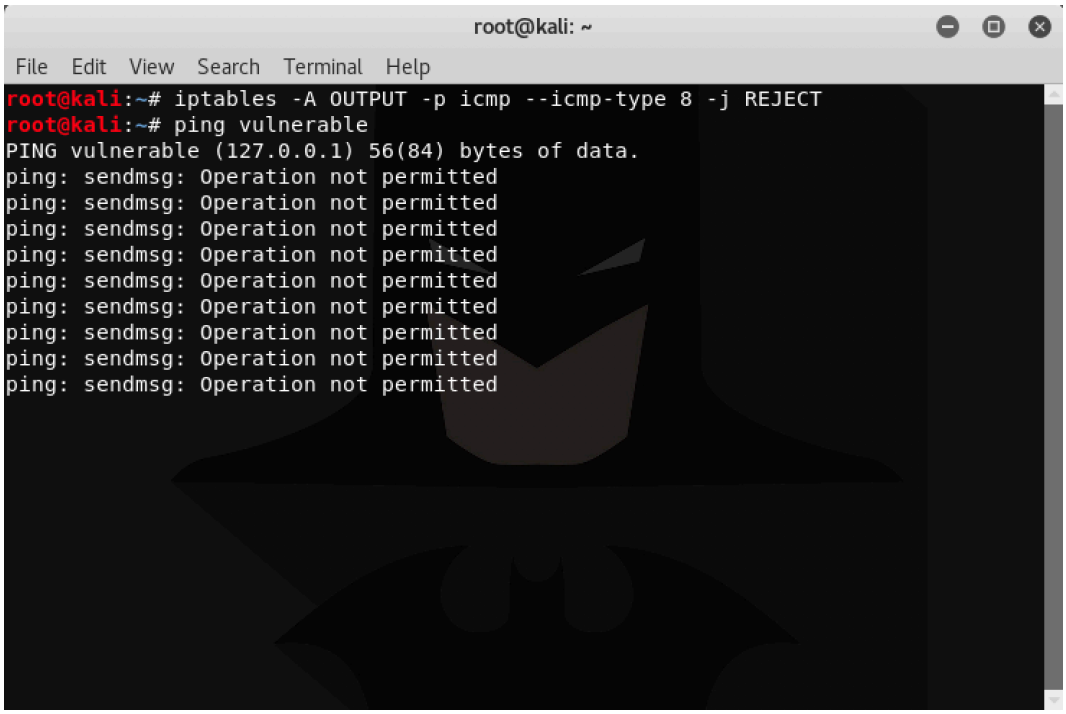
Hands on : use Nmap to find open ports on the local system



Hands on : use iptables to close open ports. Use the command:
`iptables –I INPUT –p tcp ––dport <port number> –j ACCEPT`



Refer : https://vitux.com/how-to-block-allow-ping-using-iptables-in-ubuntu/


root@kali: ~
File Edit View Search Terminal Help
root@kali:~# iptables -A OUTPUT -p icmp --icmp-type 8 -j REJECT
root@kali:~# ping vulnerable
PING vulnerable (127.0.0.1) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted

Chapter 11 : WiFi

Environment variables:

An environment variable is a named object that contains data used by one or more applications. In simple terms, it is a variable with a name and a value. The value of an environmental variable can for example be the location of all executable files in the file system, the default editor that should be used, or the system locale settings. Users new to Linux may often find this way of managing settings a bit unmanageable. However, environment variables provide a simple way to share configuration settings between multiple applications and processes in Linux.

Network time protocol:

The Network Time Protocol (NTP) is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks. In operation since before 1985, NTP is one of the oldest Internet protocols in current use. NTP was designed by David L. Mills of the University of Delaware.
NTP is intended to synchronize all participating computers to within a few milliseconds of Coordinated Universal Time (UTC). It uses the intersection algorithm, a modified version of Marzullo's algorithm, to select accurate time servers and is designed to mitigate the effects of variable network latency. NTP can usually maintain time to within tens of milliseconds over the public Internet, and can achieve better than one millisecond accuracy in local area networks under ideal conditions. Asymmetric routes and network congestion can cause errors of 100 ms or more.

Server message block:
In computer networking, Server Message Block (SMB), one version of which was also known as Common Internet File System (CIFS), operates as an application-layer or presentation-layer network protocol mainly used for providing shared access to files, printers, and serial ports and miscellaneous communications between nodes on a network. It also provides an authenticated inter-process communication mechanism. Most usage of SMB involves computers running Microsoft Windows, where it was known as "Microsoft Windows Network" before the introduction of Active Directory. Corresponding Windows services are LAN Manager Server (for the server component) and LAN Manager Workstation (for the client component).

Refer : https://pentesterlab.com/exercises/rack_cookies_and_commands_injection/

Hands on : set up a WEP network and crack the key
Refer : https://www.aircrack-ng.org/doku.php?id=simple_wep_crack
Refer : https://null-byte.wonderhowto.com/how-to/hack-wi-fi-cracking-wep-passwords-with-aircrack-ng-0147340/

Chapter 12 : Linux exploitation

Memory management:

Memory management is a form of resource management applied to computer memory. The essential requirement of memory management is to provide ways to dynamically allocate portions of memory to programs at their request, and free it for reuse when no longer needed. This is critical to any advanced computer system where more than a single process might be underway at any time.
Several methods have been devised that increase the effectiveness of memory management. Virtual memory systems separate the memory addresses used by a process from actual physical addresses, allowing separation of processes and increasing the size of the virtual address space beyond the available amount of RAM using paging or swapping to secondary storage. The quality of the virtual memory manager can have an extensive effect on overall system performance.

Stack:

In computer science, a call stack is a stack data structure that stores information about the active subroutines of a computer program. This kind of stack is also known as an execution stack, program stack, control stack, run-time stack, or machine stack, and is often shortened to just "the stack". Although maintenance of the call stack is important for the proper functioning of most software, the details are normally hidden and automatic in high-level programming languages. Many computer instruction sets provide special instructions for manipulating stacks.

Buffer overflow protection:

Buffer overflow protection is any of various techniques used during software development to enhance the security of executable programs by detecting buffer overflows on stack-allocated variables, and preventing them from causing program misbehaviour or from becoming serious security vulnerabilities. A stack buffer overflow occurs when a program writes to a memory address on the program's call stack outside of the intended data structure, which is usually a fixed-length buffer. Stack buffer overflow bugs are caused when a program writes more data to a buffer located on the stack than what is actually allocated for that buffer. This almost always results in corruption of adjacent data on the stack, which could lead to program crashes, incorrect operation, or security issues.

Hands on : solve nebula exercises from 0 to 4.
Refer : https://github.com/antoinet/nebula

Chapter 13 : SSL pinning and linux exploitation

Refer : https://www.imperialviolet.org/2011/05/04/pinning.html for public key pinning.

Refer : https://moxie.org/blog/authenticity-is-broken-in-ssl-but-your-app-ha/ for protection against false ssl ceritifiacates etc.

Hands on : Solve nebula levels 5-9.
Refer : https://github.com/antoinet/nebula

Chapter 14 : Web for Pentester
Refer : https://pentesterlab.com/exercises/web_for_pentester/course

Hands on : solve nebula levels 10 - 14
Refer : https://github.com/antoinet/nebula

Tuesday, 25 June 2019

Chapter 15 : Web for Pentester II
Refer : https://pentesterlab.com/exercises/web_for_pentester_ii/course

Hands on : solve nebula levels 15-19
Refer : https://github.com/antoinet/nebula