Cyber Security Fellowship by ByteWise powered by Secure Pwn.

Who we are?

At Secure Pwn, we're passionate about all things tech. From cloud engineering to cybersecurity, we're dedicated to helping our clients stay ahead of the curve in a rapidly evolving digital landscape. Whether you're a student looking to launch your career, an established professional seeking to enhance your skills, or a business in need of cutting-edge services, we've got you covered.

We specialize in organizing technology events that bring together the brightest minds in the industry, providing top-notch training to help you master the latest tools and techniques, and delivering customized services tailored to meet your unique needs. With our team of experienced professionals and our commitment to excellence, we're confident that we can help you achieve your goals and succeed in today's fast-paced world of tech.

So, whether you're looking to stay on top of the latest trends in cloud engineering and cybersecurity, or you're ready to take your business to the next level, trust Secure Pwn to be your partner in success. Contact us today to learn more about how we can help you thrive in the digital age.

Collaboration with ByteWise

We are glad to collaborate with the ByteWise to provide on the course content for Cyber Security. ByteWise has been working extraordinary for providing the students a great learning pattern and practical approach. I am hopeful you will have a great journey learning Cyber Security.

Follow us on LinkedIn

Linkedin: https://www.linkedin.com/company/secure-pwn

Author’s profile: https://www.linkedin.com/in/syedalizain033

Author’s Email: Syedalizain03@gmail.com

Author’s social media: **@syedalizain033**

Good luck,

Contents

What we expect you already know?

- Python (Basics)
- Internet browsing
- Understanding of IT (Basics)
- Quick learning of IT and computer science

This course is completely for beginners. For learning cyber security, there is only one rule: "**DON'T BE A SCRIPT KIDDIE**".

Introduction

What is Cyber Security?

Cybersecurity is the practice of protecting computer systems, networks, and sensitive data from unauthorized access, theft, and damage. It involves a range of technologies, processes, and practices designed to safeguard information and prevent cyber-attacks, which can take many forms, including viruses, malware, phishing, and ransomware.

To protect against cyber-attacks, cybersecurity professionals use a combination of technical tools and strategies. These can include firewalls, intrusion detection systems, anti-virus software, encryption, and secure coding practices. They may also implement security policies and procedures to govern access to sensitive data, train employees on how to identify and avoid cyber threats, and conduct regular audits and risk assessments to identify vulnerabilities and improve security.

One of the key challenges in cybersecurity is staying up-to-date with the latest threats and techniques used by hackers and cybercriminals. This requires constant monitoring and analysis of new security vulnerabilities and emerging trends in cyber-attacks. Cybersecurity professionals must also work to balance security with usability and accessibility, ensuring that systems and data remain secure while still allowing users to access and use the technology they need to do their jobs.

Overall, cybersecurity is a critical component of modern technology and plays an increasingly important role in protecting individuals, businesses, and governments from the growing threat of cyber-attacks. As technology continues to evolve and cyber threats become more sophisticated, the need for cybersecurity professionals and robust cybersecurity strategies will only continue to grow.

Why is it important to learn Information Technology before learning Cyber Security?

Learning information technology before learning cybersecurity is important because cybersecurity is a specialized field that builds on a foundation of IT knowledge and skills.

Information technology encompasses a wide range of technologies and tools used to manage and process information. This includes computer hardware and software, databases, networks, and other digital systems. IT professionals are responsible for designing, implementing, and maintaining these systems, as well as ensuring that they are secure and reliable.

Cybersecurity, on the other hand, is a subset of IT that focuses specifically on protecting computer systems, networks, and data from cyber threats. To be effective in this field, it's important to have a strong foundation in IT, including an understanding of computer networks, operating systems, and programming languages.

CIA Triad

The CIA triad is a model that defines the three main objectives of cybersecurity: Confidentiality, Integrity, and Availability.

- Confidentiality: Confidentiality is the ability to keep data secret and protected from unauthorized access. This includes the use of encryption, access controls, and other security mechanisms to ensure that only authorized users have access to sensitive data. For example, in a healthcare system, patient medical records must be kept confidential to protect their privacy.
- Integrity: Integrity refers to the accuracy and consistency of data. Maintaining data integrity ensures that the information is not tampered with, corrupted, or modified in any unauthorized way. For example, in a financial system, it is important to ensure that financial transactions are not tampered with or modified in any way.
- Availability: Availability means ensuring that data and services are available and accessible to authorized users when they need them. This includes the use of redundant systems, backup and recovery procedures, and other measures to ensure that systems are always available. For example, in an e-commerce website, it is important to ensure that the website is always available for customers to place orders and make purchases.

The CIA triad is a fundamental concept in cybersecurity and is used as a guide for developing security policies, procedures, and controls to protect against cyber threats. It is important to balance these three objectives to ensure that security measures do not compromise the functionality of the system and the user experience.

Web Security

Web security is the practice of protecting websites, web applications, and web services from threats such as unauthorized access, theft of data, denial of service, and malware attacks. Web security is a critical component of cyber security as it ensures the confidentiality, integrity, and availability of sensitive information that is transmitted over the internet.

There are various web security threats that organizations face, such as cross-site scripting (XSS), SQL injection attacks, cross-site request forgery (CSRF), phishing, and malware infections. These threats can result in significant financial losses, reputation damage, and legal penalties for businesses.

Web security is important because it protects the sensitive data of users and organizations, such as personal information, financial data, and intellectual property. A web security breach can cause significant harm to individuals and organizations, leading to loss of trust and damage to reputation. It is crucial for organizations to implement strong web security measures to ensure that their web applications and services are secure and to protect their users' data.

To ensure web security, organizations should implement various security measures, such as using secure communication protocols, using strong authentication mechanisms, implementing firewalls, and performing regular security audits and vulnerability assessments. Additionally, organizations should provide web security training to their employees to ensure that they are aware of the latest security threats and how to prevent them.

What is vulnerability?

A vulnerability refers to the security weakness that is found in the system, application, or anything. It may be a bug which has some impact on the application but with the security impact or in the CIA triad.

Who is Penetration Tester?

A penetration tester, also known as an ethical hacker, is a cybersecurity professional who is responsible for evaluating the security of computer systems, networks, and applications by simulating attacks to identify vulnerabilities and weaknesses. Penetration testers use a variety of techniques and tools to identify vulnerabilities and attempt to exploit them to gain unauthorized access or perform other malicious activities. They work to identify and report these vulnerabilities to the organization so that appropriate measures can be taken to fix them and improve overall security. Penetration testing is an essential component of any comprehensive cybersecurity program, as it helps organizations identify and remediate vulnerabilities before they can be exploited by malicious actors.

Who is web penetration tester?

A web penetration tester is a type of security professional who specializes in identifying vulnerabilities and potential attacks in web applications, websites, and other online services. Their job is to simulate attacks on web systems in order to identify weaknesses before they can be exploited by malicious attackers.

Web penetration testers use a variety of tools and techniques to identify and exploit vulnerabilities in web systems, such as cross-site scripting (XSS), SQL injection, and cross-site request forgery (CSRF). They may also test for weaknesses in authentication and authorization mechanisms, session management, and other aspects of web security.

Week 1-2: Warm up

What is networking?

A computer network is a set of computers sharing resources located on or provided by network nodes. The computers use common communication protocols over digital interconnections to communicate with each other. These interconnections are made up of telecommunication network technologies, based on physically wired, optical, and wireless radio-frequency methods that may be arranged in a variety of network topologies.

The nodes of a computer network can include personal computers, servers, networking hardware, or other specialized or general-purpose hosts. They are identified by network addresses, and may have hostnames. Hostnames serve as memorable labels for the nodes, rarely changed after initial assignment. Network addresses serve for locating and identifying the nodes by communication protocols such as the Internet Protocol.

Computer networks may be classified by many criteria, including the transmission medium used to carry signals, bandwidth, communications protocols to organize network traffic, the network size, the topology, traffic control mechanism, and organizational intent.

Computer networks support many applications and services, such as access to the World Wide Web, digital video, digital audio, shared use of application and storage servers, printers, and fax machines, and use of email and instant messaging applications.

Network packets

Most modern computer networks use protocols based on packet-mode transmission. A network packet is a formatted unit of data carried by a packet-switched network.

Packets consist of two types of data: control information and user data (payload). The control information provides data the network needs to deliver the user data, for example, source and destination network addresses, error detection codes, and sequencing information. Typically, control information is found in packet headers and trailers, with payload data in between.

With packets, the bandwidth of the transmission medium can be better shared among users than if the network were circuit switched. When one user is not sending packets, the link can be filled with packets from other users, and so the cost can be shared, with relatively little interference, provided the link isn't overused. Often the route a packet needs to take through a network is not immediately available. In that case, the packet is queued and waits until a link is free.

The physical link technologies of packet network typically limit the size of packets to a certain maximum transmission unit (MTU). A longer message may be fragmented before it is transferred and once the packets arrive, they are reassembled to construct the original message.

What is network protocol?

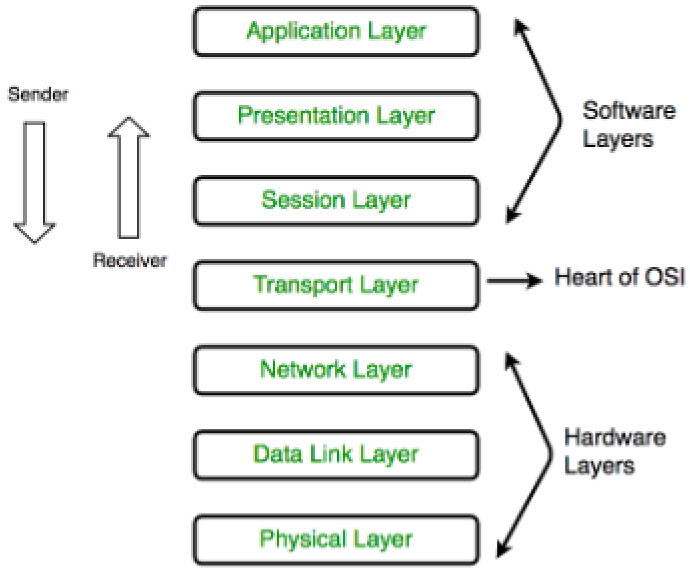
A network protocol is a set of rules and standards that enables communication between devices over a network. It defines the format and order of messages exchanged between devices, as well as the actions taken upon receipt of those messages. A protocol provides a common language for devices on a network to communicate and ensures that data is transferred accurately and efficiently. Examples of network protocols include TCP/IP, HTTP, FTP, DNS, and SSH. We shall study them in detail.

OSI Model

The OSI model is a conceptual framework used to describe the functions of a networking system. It stands for Open Systems Interconnection and was developed by the International Organization for Standardization (ISO). The model is divided into seven layers, each of which has a specific function:

- Physical layer: This layer deals with the physical aspects of data transmission, such as the cables, connectors, and hardware devices used to send and receive data.
- Data link layer: This layer deals with the formatting and organization of data into frames, which are then transmitted over the physical layer.
- Network layer: This layer is responsible for addressing and routing data packets across different networks.
- Transport layer: This layer provides reliable data transmission services between two endpoints, ensuring that all data is transmitted and received correctly.
- Session layer: This layer establishes, maintains, and terminates communication sessions between two devices.
- Presentation layer: This layer is responsible for formatting and presenting data to the application layer in a way that is understandable to the user.
- Application layer: This layer provides services to end-user applications, such as email, file transfer, and web browsing.

The OSI model is useful for understanding the various functions of a networking system and for troubleshooting issues that may arise at different layers of the system. It also serves as a standard reference model for networking protocols and allows for the interoperability of different systems and devices.



TCP/IP

The TCP/IP (Transmission Control Protocol/Internet Protocol) model is a protocol stack that is used in computer networks and the internet. It is a communication protocol that allows computers to connect and communicate with each other over a network.

The TCP/IP model is made up of four layers, each with a specific function:

- Application layer: This layer provides network services to applications. Examples of protocols at this layer include HTTP, FTP, SMTP, and Telnet.
- Transport layer: This layer is responsible for end-to-end communication between hosts. It establishes and terminates connections, and provides flow control, error correction, and congestion control. Examples of protocols at this layer include TCP and UDP.
- Internet layer: This layer is responsible for addressing, routing, and fragmentation of data packets. It ensures that data is transmitted from the source to the destination over the internet. The protocol at this layer is the Internet Protocol (IP).
- Network access layer: This layer provides physical access to the network, and defines the protocols and hardware necessary for communication. Examples of protocols at this layer include Ethernet, Wi-Fi, and DSL.

The TCP/IP model is similar to the OSI (Open Systems Interconnection) model in that it is a layered model that describes how data is transmitted over a network. However, the TCP/IP model is more widely used in practice, particularly in the context of the internet.

TCP/IP	OSI

Application	Application
Transport	Presentation
	Session
	Transport
Network/Internet	Network
Data Link	Data Link
Physical	Physical

Above table would give a detailed information about the difference. It’s important to study TCP/IP in every aspect. We shall be focusing more on Application layer and Transport layer.

DNS

DNS stands for Domain Name System, which is a hierarchical decentralized naming system for computers, services, or other resources connected to the Internet or a private network. DNS is essentially a protocol that translates human-readable domain names into IP addresses, which are the numerical addresses that computers use to identify and communicate with each other on a network.

The DNS protocol is based on a client/server architecture, where DNS clients (usually web browsers or other Internet applications) send requests to DNS servers, which are responsible for responding with the IP address associated with the requested domain name. When a user types a URL into their web browser, the browser sends a DNS query to the DNS server to look up the IP address associated with that domain name. The DNS server then responds with the appropriate IP address, which the browser uses to connect to the web server and retrieve the webpage.

DNS operates using a hierarchical system of DNS servers, where each server is responsible for a particular portion of the DNS name space. At the top of the hierarchy are the root servers, which are responsible for providing information about the top-level domains (such as .com, .org, .net, etc.). Below the root servers are the top-level domain (TLD) servers, which are responsible for providing information about the domains registered within that TLD. Finally, there are the authoritative name servers, which are responsible for providing the IP addresses associated with specific domain names.

Overall, DNS is a critical protocol for the functioning of the Internet and is used extensively in all types of network communication. It is important for security professionals to understand how DNS works, as it is often a target for attacks such as DNS spoofing or DNS cache poisoning, which can be used to redirect users to malicious websites or steal sensitive information.

Self-learning on TCP/IP (Important and MUST)

1. https://www.geeksforgeeks.org/tcp-ip-model/
2. https://www.youtube.com/watch?v=BnWn18qUYyA (Protocol basics)

3. https://www.scaler.com/topics/computer-network/application-layer-protocols/ (Application protocols)
4. https://www.youtube.com/watch?v=JkEYOt08-rU (What is a DNS and how it works)
5. https://www.youtube.com/watch?v=fQC4v07gs5k (How TCP protocol works)
6. *(TCP protocol is a connection protocol in transport layer of TCP/IP Stack Model. Don't confuse between TCP/IP Stack and TCP Protocol.)*
7. https://www.youtube.com/watch?v=rmFX1V49K8U (Diving deep into TCP/IP protocol and communication).
8. https://www.youtube.com/watch?v=aE75gHVK16A (HTTP)
9. https://www.ibm.com/docs/en/ibm-mq/9.3?topic=tls-overview-ssltls-handshake (TLS)
10. https://www.youtube.com/watch?v=C4Gtq5anlyc (TLS)

We covered from the videos: what is TCP/IP model, what is a protocol, studying about how protocols work and feels like in the networks, DNS, HTTP, and cleared concepts of communications.

Recap

So far, we have understood how HTTP works, let me explain you something about the communication a deeper in simple words which would make you different from other developer. Let's imagine everything like a story and image.

When you search something in the browser, the browser has to first establish the connection with the web server. Browser does not know whether the web server wants to communicate or not. So, the web server decides to first communicate with the web server using the connection-oriented protocol TCP. TCP has an ability to not just communicate but to verify that both are communicating and agreeing for the communication. The browser has a tab opened which opens a port of your system as the TCP port being logical from around 65000. Usually, it chooses a 4-5 numbers long port because short number ports are mostly reserved or specified for other protocols like FTP which are going to exist for longer but the tab won't exist for longer. Coming back to story, the tab opens connection via the port and performs a successful connection with the web server on its IP and port number. Here is another magic knowledge, the TCP protocol only carries port number, including the sender's port and receiver's port so that the connection could be received back on the same port, and the IP address is placed on the network layer, not the TCP layer or TCP protocol. When it establishes the connection, it sends the HTTP or HTTPs communication over this established connection. When you close the tab, it sends the finish signal to terminate the connection. Study the structure of TCP packet to understand what it contains in it.

Quiz time

- Which protocol is reliable and why? TCP or UDP
- Which stack is followed in practical approach? TCP/IP or OSI?
- Encryption of data being sent is handled on which layer?
- What is FTP protocol and on which layer it does work?
- On which layer, should DNS work? Guess which protocol DNS follows?
- Bob says, TCP protocol provides data integrity, John says UDP provides data integrity, who is right?
- Listening to above conversation, Kevin is totally blank. Can you explain him how one of them provides data integrity?

Feel free to use internet and research on each question, but remember the rule: -

"**DON'T BE A SCRIPT KIDDIE**".

Week 3-4: Web communication and interception

The HTTP structures

An HTTP (Hypertext Transfer Protocol) request consists of several parts:

- Request line: This line specifies the HTTP method (GET, POST, PUT, DELETE, etc.), the URL of the resource being requested, and the HTTP version being used.
- Headers: These are optional fields that provide additional information about the request, such as the browser being used, the preferred language of the response, and any authentication information.
- Empty line: This is an empty line that marks the end of the header section and separates the header section from the body section.
- Body: This is an optional section that contains data being sent in the request, such as form data or JSON data.

Here is an example of a simple HTTP request:

GET /index.html HTTP/1.1

Host: www.example.com

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

In this example, the request is a GET request for the index.html file on the www.example.com server using HTTP version 1.1. The "Host" field specifies the domain name of the server, the "User-Agent" field provides information about the browser being used to make the request, and the "Accept" field specifies the types of content that the client is willing to accept in the response.

HTTP request methods

GET, POST, PUT, and DELETE are the four most commonly used HTTP methods. They are used to specify the intended action to be performed for a given resource.

GET

GET is used to retrieve data or information from a specified resource. The requested data is returned in the response body of the HTTP message. A GET request should not change the state of the resource on the server. For example, if you want to retrieve a webpage from a server, you can use a GET request.

Example:

```
GET /api/products HTTP/1.1

Host: example.com
```

In this example, a GET request is sent to the server to retrieve the products from the API endpoint.

POST

POST is used to submit data or information to be processed to a specified resource. The data sent to the server is included in the body of the HTTP message. A POST request can change the state of the resource on the server. For example, when you submit a form on a website, a POST request is sent to the server.

Example:

```
POST /api/products HTTP/1.1

Host: example.com

Content-Type: application/json


{

  "name": "Product Name",

  "price": 100

}
```

In this example, a POST request is sent to the server to create a new product with the given name and price.

PUT

PUT is used to update a resource on the server. The data sent to the server is included in the body of the HTTP message. A PUT request should be idempotent, meaning that sending the same request multiple times should have the same effect as sending it once. For example, when you update a user's profile on a website, a PUT request is sent to the server.

Example:

PUT /api/products/1 HTTP/1.1

Host: example.com

Content-Type: application/json

{

"name": "New Product Name",

"price": 200

}

In this example, a PUT request is sent to the server to update the product with ID 1 with the new name and price.

DELETE

DELETE is used to delete a specified resource on the server. The data sent to the server is included in the body of the HTTP message. A DELETE request should be idempotent, meaning that sending the same request multiple times should have the same effect as sending it once. For example, when you delete a post on a website, a DELETE request is sent to the server.

Example:

DELETE /api/products/1 HTTP/1.1

Host: example.com

In this example, a DELETE request is sent to the server to delete the product with ID 1.

HTTP Headers

Read about all the HTTP readers we are going to utilize in web security.

https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Headers_Cheat_Sheet.html

Structure of website

The website has 3 major parts: domain, path or directory, parameters and its value.

Showing an example site: "**example.com/buy/products?productname=shirt**". I know you already understood which part belongs to what category. Talking about the end point, if someone says that what was the end point you performed XYZ attack on? You should know that the end point means the path or directory including the parameter and value if there is any so the other person could exactly reproduce it. The endpoint may be the “buy” in the above example. For example, you saying that the buy endpoint is vulnerable to XYZ attack so all endpoints from that endpoint are vulnerable. Means “/buy” was vulnerable to all paths after buy would be vulnerable too.

Burp Suite

Bob is the greatest hacker ever found in this universe. Bob claims that since he started studying web security, he is using one tool till now, **Burp Suite**.

Burp Suite is a popular and powerful tool used for web application security testing. It is designed to be an all-in-one solution for penetration testing and vulnerability scanning of web applications. Burp Suite provides a comprehensive range of features, such as intercepting and modifying HTTP requests and responses, scanning for security vulnerabilities, and brute force testing. Some of the key features of Burp Suite include:

- Intercepting Proxy: Burp Suite acts as a proxy server that sits between the client and the server. This allows the user to intercept and modify HTTP requests and responses.
- Spidering: Burp Suite has a built-in web spider that can crawl a website and map out its structure.
- Scanner: Burp Suite comes with a scanner that can automatically scan a website for security vulnerabilities.
- Intruder: Burp Suite's intruder tool can be used for brute force testing of web applications.
- Repeater: The repeater tool allows the user to manually modify and re-send individual HTTP requests.

Burp Suite is widely used in the field of web application security testing by security researchers, penetration testers, and ethical hackers. Its powerful features and ease of use make it a popular choice for professionals in this field.

Downloading burp

The burp suite can be downloaded from its official website.

How to install Burp Suite: https://youtube.com/watch?v=ZWKqxQF6aow

Burp now provides build-in browser facility however you can connect it with other browser if you want. Most researchers use Firefox with Burp using following link.

https://www.youtube.com/watch?v=Vn_Zst6BMGo

Learning Burp suite

You can learn burp suite form the links below. There are high chances that there is a little change in the interfaces compared to in the videos but not a major change, so understanding the concept and exploring would be enough.

https://www.youtube.com/watch?v=G3hpAeoZ4ek

https://www.youtube.com/watch?v=IWWYNDiwYOA

Lab Development

We shall be developing our lab for learning the burp suite and understanding how the web application may work in the backend and how it can be seen from a black perspective. Before that, we shall jump into the types of testing.

Types of testing

Black box testing, gray box testing, and white box testing are three different approaches to software testing.

Black box testing

In black box testing, the tester has no knowledge of the internal workings of the system under test. The tester treats the system as a "black box" and only tests the input and output of the system. Black box testing is done from the perspective of the user, with the aim of finding defects that would impact the user experience. It is typically used for functional testing and is very effective in uncovering errors related to data flow, computation errors, and user interface issues. Testers use various techniques such as boundary value analysis, equivalence partitioning, and error guessing to conduct black box testing. Penetration testing is also a type of black box testing.

Gray box testing

Gray box testing is a combination of black box and white box testing. In gray box testing, the tester has some knowledge of the internal workings of the system. The tester can see the internal code and logic but does not have full access to it. Gray box testing is typically used for integration testing, and the testers have access to some limited information such as database schemas, system designs, and other high-level information.

White box testing

In white box testing, the tester has complete knowledge of the internal workings of the system under test. The tester can see the source code, the logic, and the internal workings of the system. White box testing is typically used for unit testing, and it is very effective in uncovering coding errors such as syntax errors, data flow errors, and logic errors. Testers use techniques such as code reviews, static analysis, and code coverage analysis to conduct white box testing.

Most of the time in penetration testing, we are doing the black box testing where we do not have the source code so it makes it necessary to know how it would look like in the low level architecture like networks, protocols, and request-response scheme.

Flask

We shall be writing codes in flask. I assume that you have knowledge of flask.

https://www.youtube.com/watch?v=Z1RJmh_OqeA

Make sure to develop different request types, GET, POST, and then intercept it in the burp suite and observe how do you see the data. Make sure to research on how the request is looking like and what headers are doing what.

Hacker's approach

The hacker intercepts the http request, observes all the data going, including the parameters, authentication tokens, access control data to see what a user is not intended to do with these data provided. Then the hacker tries to manipulate each header exactly knowing what header does what action in the backend what may be the expected response from the backend and if there is any possibility in bypassing that implemented restriction. For example, if the website says if admin header is set then its admin otherwise don't allow to access the admin panel and a hacker manipulates the headers by adding the admin header allowing himself to access the admin panels and perform privileged actions.

What's next?

Thank you for reaching this end and learning with us. I hope so you have gained enough knowledge to understand the basics. Those who understood enough shall move to the next month course which will be shared through the ByteWise. For any query, feel free to reach me out. Do share the Secure Pwn with others and keep learning.

Good Luck.