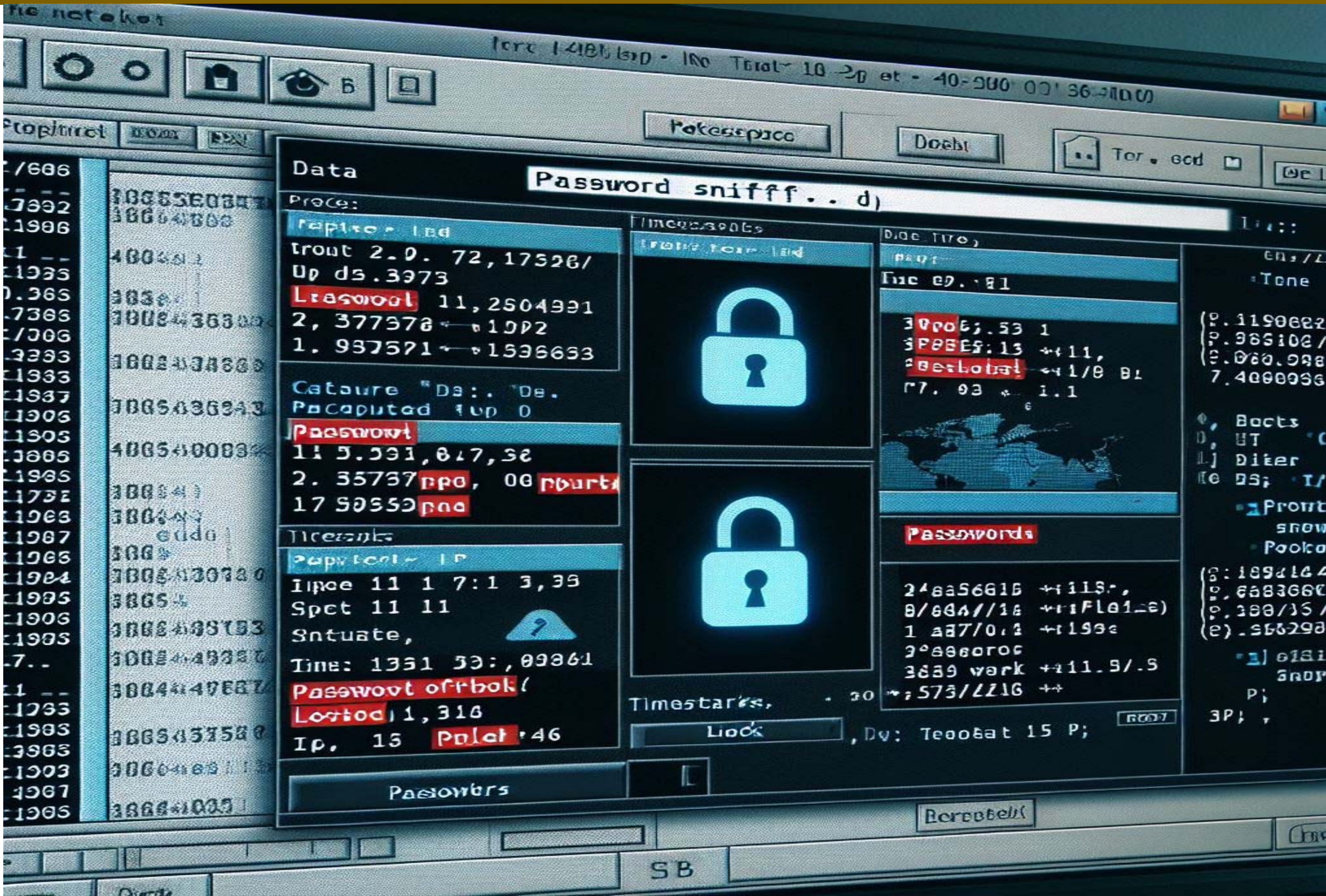
Network Packet Forensics

Ali Ali

Digitally signed by Ali Ali
DN: c=LB, st=Beirut, l=AA, o=MISC, ou=ISC, cn=Ali Ali
Date: 2024.10.24 06:27:49 +03'00'

Ali Ali



Network Packet Forensics
Wireshark
Password Sniffing

❑ *Password Sniffing with Wireshark*

- *Overview*

Password sniffing involves capturing and analyzing network traffic to extract sensitive information, such as passwords. This is particularly feasible with clear text protocols that do not encrypt communication, making all data, including passwords, visible to anyone who can intercept the traffic

- *Capturing HTTP Passwords*

HTTP (Hypertext Transfer Protocol) typically operates on port 80/TCP and transmits data in plain text. This lack of encryption means that anyone intercepting the traffic can see all the data being exchanged, including login credentials

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing

- *Example of Capturing HTTP Passwords*

During penetration testing or network analysis, you might encounter HTTP traffic. Here's how you can capture login credentials using Wireshark:

1. Start Wireshark: Open Wireshark and start capturing traffic on the relevant network interface

2. Filter HTTP Traffic: Use the filter http to display only HTTP traffic

3. Locate POST Requests: Look for HTTP POST requests, which often contain login credentials

4. Examine the Data: Click on a POST request and examine the packet details. You might see something like this:

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing

```
POST /login HTTP/1.1
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 29

username=admin&password=12345
```

The username and password are transmitted in plain text

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing

- *Monitoring HTTPS Packets over SSL/TLS*

HTTPS (HTTP Secure) encrypts data using SSL/TLS, making it much more secure than HTTP. However, with the right tools and access to the server's private key, it is possible to decrypt HTTPS traffic

- *Dissecting HTTPS Packet Captures*

To analyze HTTPS traffic in Wireshark, follow these steps:

1. Open the Capture File: Open the provided HTTPS/TLS.pcapng file in Wireshark

2. Observe the Handshake: Look for the SSL/TLS handshake, which includes messages like Client Hello, Server Hello, and key exchange information

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing

3. Decrypt the Data: To decrypt the encrypted application data, you need the server's private key

- *Steps to Decrypt TLS/SSL Data*

1. Navigate to Preferences: Go to Edit > Preferences > Protocols > TLS

2. Add Decryption Keys: Add the necessary values:

- *IP Address: x.x.x.x*
- *Port: 443*
- *Key File: Path to the server's private key file*

Once configured, Wireshark will decrypt the TLS/SSL data, allowing you to view the previously encrypted information

Network Packet Forensics
Wireshark
Password Sniffing

- *Telnet Password Capture*

Telnet is a protocol that operates on port TCP/23 and is often used for administrative purposes. However, it is known for its lack of security because it transmits data, including passwords, in plain text. This makes it vulnerable to interception by attackers

- *Example of Telnet Password Capture*

1. Start Wireshark: Open Wireshark and start capturing traffic on the relevant network interface

2. Filter Telnet Traffic: Use the filter telnet to display only Telnet traffic

3. Locate Login Attempts: Look for Telnet login attempts in the captured packets

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing

4. Examine the Data: Click on a packet and examine the details to find the username and password transmitted in plain text

- *FTP Password Capture*

FTP (File Transfer Protocol) typically uses ports TCP/20 and TCP/21. Like Telnet, FTP transmits data in plain text, making it easy for attackers to capture login credentials

  - *Example of FTP Password Capture*

1. Start Wireshark: Open Wireshark and start capturing traffic on the relevant network interface

2. Filter FTP Traffic: Use the filter ftp to display only FTP traffic

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing

3. Locate Login Attempts: Look for FTP login attempts in the captured packets

4. Examine the Data: Click on a packet and examine the details to find the username and password transmitted in plain text

- *SMTP Password Capture*

SMTP (Simple Mail Transfer Protocol) uses port TCP/25 for sending emails. Although secure versions exist (e.g., using STARTTLS on port TCP/465), many servers still support plain text authentication

  - *Example of SMTP Password Capture*

1. Start Wireshark: Open Wireshark and start capturing traffic on the relevant network interface

Network Packet Forensics
Wireshark
Password Sniffing

2. Filter SMTP Traffic: Use the filter smtp to display only SMTP traffic

3. Locate Authentication Attempts: Look for SMTP authentication attempts in the captured packets

4. Decode Base64: SMTP often uses Base64 encoding for usernames and passwords. Decode these values using a Base64 decoder

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing

- *Decoding Base64 Example*

1. Captured Base64 encoded credentials might look like this:

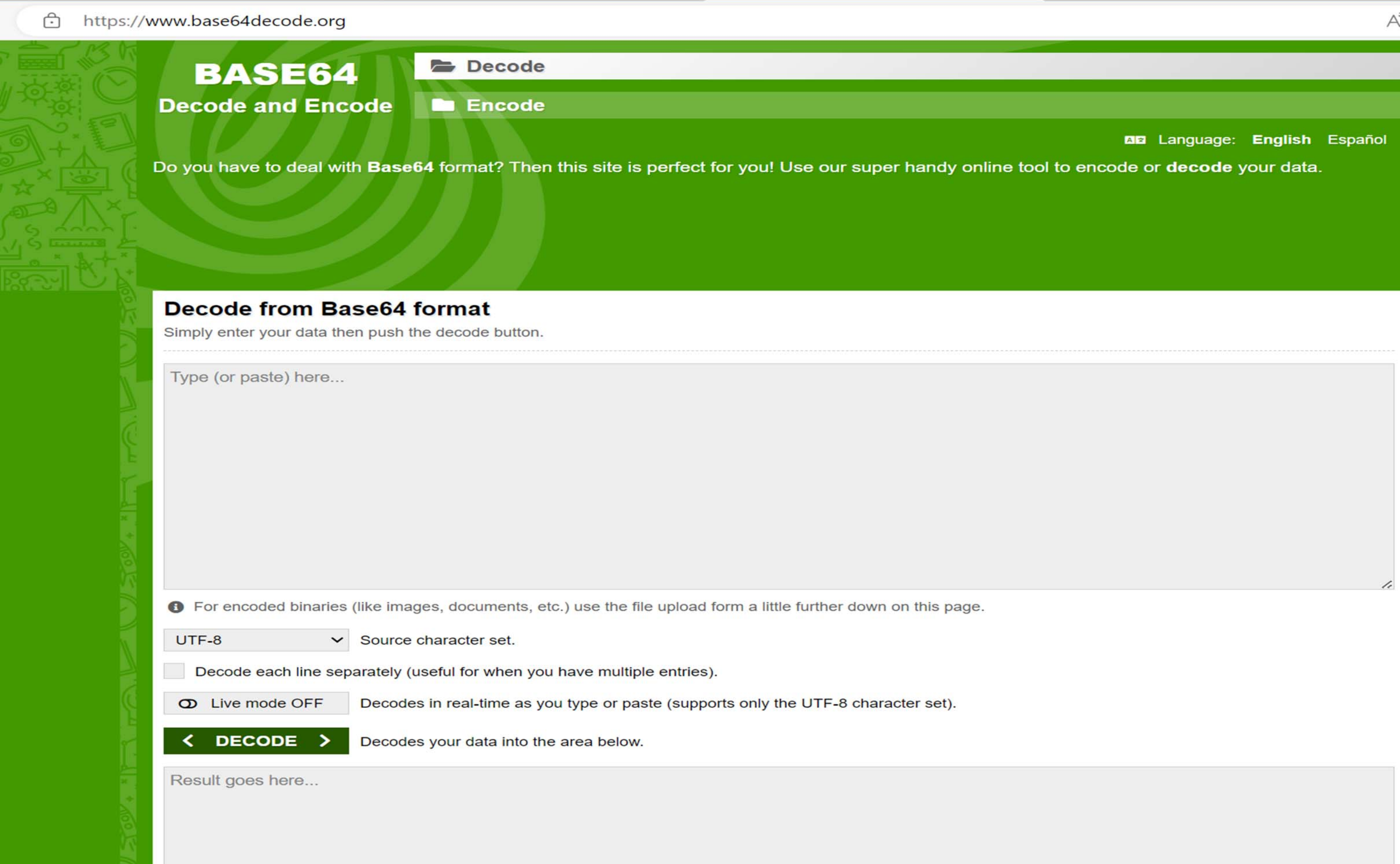
- *Username: Z3VycGFydGFwQHBhdHJpb3RzLmlu*
- *Password: cHVuamFiQDEyMw==*

2. Use an online Base64 decoder to decode these strings:

- *Decoded Username: ....*
- *Decoded Password: ....*

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing



https://www.base64decode.org

Do you have to deal with **Base64** format? Then this site is perfect for you! Use our super handy online tool to encode or **decode** your data.

Decode from Base64 format

Simply enter your data then push the decode button.

Type (or paste) here...

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 — Source character set.

□ Decode each line separately (useful for when you have multiple entries).

Live mode OFF — Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > — Decodes your data into the area below.

Result goes here...

Network Packet Forensics
Wireshark
Password Sniffing

- *Analyzing SNMP Community String*

The Simple Network Management Protocol (SNMP) typically runs on port UDP/161. Its main objective is to manage and monitor network devices and their functions. SNMP has three versions, with the first two (v1 and v2c) being in plain text. SNMP uses something akin to authentication called a community string. Therefore, capturing the SNMP community string is almost like capturing credentials

- *Example of Capturing SNMP Community String*

1. Start Wireshark: Open Wireshark and start capturing traffic on the relevant network interface

2. Filter SNMP Traffic: Use the filter snmp to display only SNMP traffic

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing

3. Locate the Community String: Look for the community string in the captured packets

4. Examine the Data: Click on a packet and examine the details to find the community string transmitted in plain text

An attacker could now use the community string to collect detailed system information. This could enable the attacker to learn sensitive details about the system and make further attempts. Note that the community string sometimes also allows you to modify your remote system configuration (read/write access)

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing

- *Capturing MSSQL Passwords*

The Microsoft SQL server usually runs on port TCP/1433. This is another service we can use with Wireshark to capture the password. If the server is not configured using the ForceEncryption option, it is possible to record plain text authentication directly or via a downgrade attack

- *Example of Capturing MSSQL Passwords*

1. Start Wireshark: Open Wireshark and start capturing traffic on the relevant network interface

2. Filter MSSQL Traffic: Use the filter mssql to display only MSSQL traffic

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing

3. Locate Login Attempts: Look for login attempts in the captured packets

4. Examine the Data: Click on a packet and examine the details to find the username and password transmitted in plain text

Now, we have a privileged account on the MSSQL server. This would have a critical impact, allowing the attacker to take complete control over the database server or potentially lead to remote command execution (RCE)

- *Capturing PostgreSQL Passwords*

PostgreSQL is another widely used SQL database server. It runs on port TCP/5432 and accepts a variety of authentication methods. It is usually set to disallow clear-text authentication, but it can also be set to allow it. In such cases, a well-positioned attacker could intercept network traffic and obtain the username and password

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing

- *Example of Capturing PostgreSQL Passwords*

1. Start Wireshark: Open Wireshark and start capturing traffic on the relevant network interface

2. Filter PostgreSQL Traffic: Use the filter pgsql to display only PostgreSQL traffic

3. Locate Authentication Attempts: Look for authentication attempts in the captured packets

4. Examine the Data: Click on a packet and examine the details to find the username and password transmitted in plain text

It should be noted that PostgreSQL authentication occurs in multiple packets. The username and database name come first, followed by the password in the subsequent packet

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing

- *Creating Firewall Rules with Wireshark*

While Wireshark itself cannot block network traffic, it is a powerful tool for analyzing traffic and helping to create firewall rules. By examining the traffic in Wireshark, you can identify patterns and specific packets that you want to block

- *Steps to Create Firewall Rules with Wireshark*

1. *Capture Traffic: Start Wireshark and capture the network traffic on the relevant interface*

2. Analyze Traffic: Identify the packets you want to block. This could be based on IP addresses, ports, protocols, or other criteria

3. Select Packet: Click on the packet you want to block to highlight it

Network Packet Forensics
Wireshark
Password Sniffing

4. Generate Firewall Rule: Navigate through the menu to generate a firewall rule based on the selected packet. This is typically done through the "Analyze" or "Tools" menu, depending on your version of Wireshark

- *Example Menu Navigation*

  - *Windows Firewall (netsh): Generate rules for Windows Firewall*
  - *IP Filter (ipfw): Generate rules for IP Filter*
  - *NetFilter (iptables): Generate rules for iptables*
  - *Packet Filter (pf): Generate rules for Packet Filter*

Ali Ali

Network Packet Forensics
Wireshark
Password Sniffing

- *Copy and Paste Rules*

Once you have generated the rules, you can copy them and paste them directly into your firewall configuration. Here's an example of how the syntax might look for different firewalls:

1. Windows Firewall (netsh):

```
netsh advfirewall firewall add rule name="Block Traffic" protocol=TCP dir=in localport=80 action=block
```

2. IP Filter (ipfw):

```
ipfw add deny tcp from any to any 80
```

Network Packet Forensics
Wireshark
Password Sniffing

3. NetFilter (iptables):

```
iptables -A INPUT -p tcp --dport 80 -j DROP
```

4. Packet Filter (pf):

```
block in on em0 proto tcp from any to any port 80
```

THE
DUNNIING-KRUGER
EFFECT

Questions

Ali Ali