First, the user will receive a link to the Iranian nuclear site and a packet capture file of the communication in the Iranian reactor. In addition, the user will receive intelligence indicating that the checksum function used to verify the integrity of the packets was modified by the technical team in the reactor, and now they are using a custom function that you have access to. The Iranian nuclear site will be seemingly secure and will only allow access from another unknown Iranian nuclear site. This site is supposed to be highly secure and inaccessible, so the user will need to modify the HTTP packet sent to the server so that its referrer field originates from that site.

To determine which site this is, the user will need to analyze the packet capture file and find the DNS query that received a positive response. This query contains a domain name that allows access to the reactor site.

Furthermore, the reactor site does not permit access to users who do not speak Persian, so you will also need to modify the language field in the HTTP packet to be in Persian.

After doing all this, the site will present a password field. Only the correct password will allow progression to the next stage. To figure out the password, we need to analyze the documentation file and filter out only packets with a valid checksum to distinguish the original communication from the reactor workers from decoy communications. To filter these, we need to write a Scapy-based script that calculates the checksum for each packet in the file and prints only the content of the packets with a valid checksum, meaning they are part of the communication between the reactor workers.

After this sorting, we will receive a string encrypted with a Vigenère cipher. Once the puzzle solver realizes which cipher is being used, they can use online tools to decrypt the cipher. The decrypted string will be a line from the movie \*The Dictator\*, providing a strong hint about the password.

The line discusses the structure of the missile, and the joke in the movie is that the dictator believes a missile should be "pointy." This is the password and the answer to the question below: "What is the ideal structure for a nuclear missile?"

After entering the correct password, the solver will receive a funny image from \*The Dictator\*. Hidden in the image are two executable files. One contains an SMTP email server running on localhost, and the other contains a password validation mechanism. The email server will provide the flag only to someone who sends an email to a unique Iranian government address. To discover this email address, the solver must crack the password validation mechanism.

The password validation file is an executable that verifies the password and prints the email if the password is correct. To retrieve the password, the solver will need to perform non-trivial reverse engineering. The file contains several anti-debugging mechanisms, and the email is not explicitly encoded, nor is the key used to decrypt the encrypted email address.

Once the user manages to extract the email from the file, they must write a Python script that sends an email using the SMTP protocol. If the email is sent correctly, they will receive the flag.**סיכום:**

**Starting Point: PCAC file, custom checksum description, and website domain**

**Components**

**1. Documentation File**

* Created using the Scapy package in Python
* Contains network packets with DNS requests and responses
* Includes encrypted messages using Vigenère cipher

**2. Website**

* Built using Flask
* Hosted on PythonAnywhere
* Requires specific referrer and language headers
* Password protected

**3. Image File**

* Contains two hidden executable files (EXE)

**4. Executable Files**

**First Executable (Python-compiled EXE)**

* Implements a simple SMTP email server
* Expects an email from a specific address
* Sends back a message containing the flag

**Second Executable (C-compiled EXE)**

* Password validation file
* Uses a custom hashing algorithm
* Employs anti-debugging techniques
* Provides an encrypted email address when correct password is entered

**CTF Solution Flow**

1. Analyze the documentation file to find the correct referrer for the website
2. Access the website using the correct referrer and language headers
3. Decrypt the Vigenère cipher messages to find the website password
4. Download the image file from the website
5. Extract the two executable files from the image
6. Reverse engineer or patch the second executable to obtain the email address
7. Use the first executable (SMTP server) to send an email to the obtained address
8. Receive the flag in the response email

: **Topics**

1.UDP protocol (custom check sum)

2.HTTP protocol- using it to change the Referer filed and the langue field of the packet.

3.DNS – need to analyze DNS transportation in order to get the right domain of the referrer

4. scapy – have to use scapy or tools like scapy in order to analyze the custom check sum (wireshark just can’t filter them out)

5. cipher – decrypting the Vigner cypher text, and understanding how xor encryption works in order to understand how to get the password while revere engineering it.

6. reverse engineering- the only way to get the password to find out the email address.

7. SMTP protocol – have to send the mail to the server in the right SMTP format.

8. python sockets- implementing SMTP protocol client using python socket.