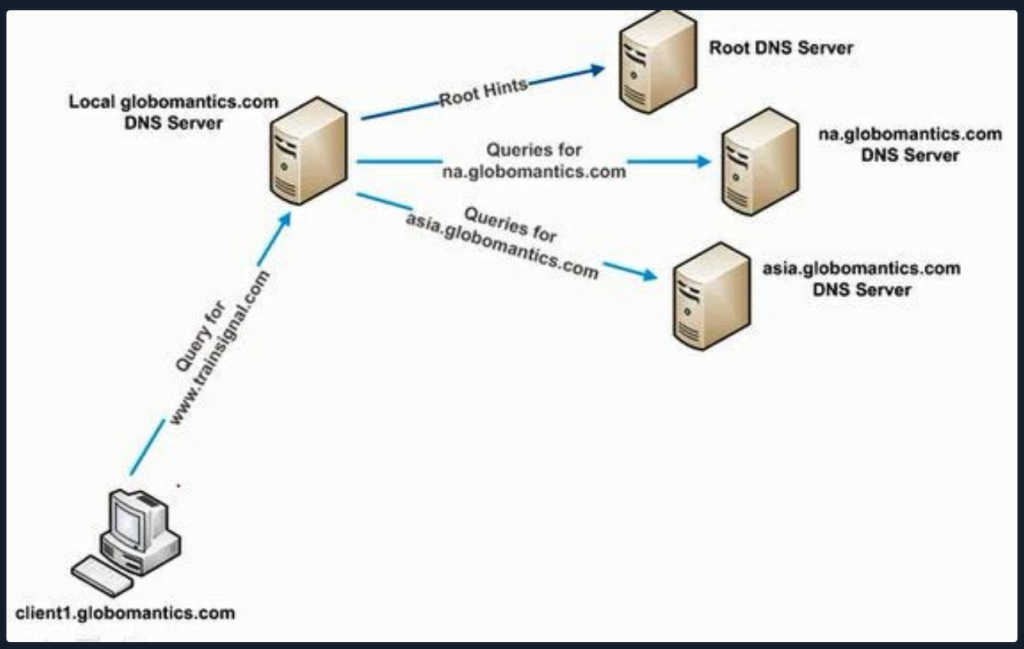
Peculiar DNS Traffic

DNS Traffic can be cumbersome to inspect, as many times our clients will generate a ton of it, and abnormalities can sometimes get buried in the mass volume of it. However, understanding DNS and some direct signs of malicious actions is important in our traffic analysis efforts.

DNS Queries

DNS queries are used when a client wants to resolve a domain name with an IP address, or the other way around. First, we can explore the most common type of query, which is forward lookups.



Generally speaking, when a client initiates a DNS forward lookup query, it does the following steps.

- Request:
  - `Where is academy.hackthebox.com?`
- Response:
  - `Well its at 192.168.10.6`

Step	Description
`1. Query Initiation`	When the user wants to visit something like academy.hackthebox.com it initiates a DNS forward query.
`2. Local Cache Check`	The client then checks its local DNS cache to see if it has already resolved the domain name to an IP address. If not it continues with the following.
`3. Recursive Query`	The client then sends its recursive query to its configured DNS server (local or remote).
`4. Root Servers`	The DNS resolver, if necessary, starts by querying the root name servers to find the authoritative name servers for the top-level domain (TLD). There are 13 root servers distributed worldwide.
`5. TLD Servers`	The root server then responds with the authoritative name servers for the TLD (aka .com or .org)
`6. Authoritative Servers`	The DNS resolver then queries the TLD's authoritative name servers for the second-level domain (aka hackthebox.com).

7. Domain Name's Authoritative Servers	Finally, the DNS resolver queries the domains authoritative name servers to obtain the IP address associated with the requested domain name (aka academy.hackthebox.com).
8. Response	The DNS resolver then receives the IP address (A or AAAA record) and sends it back to the client that initiated the query.

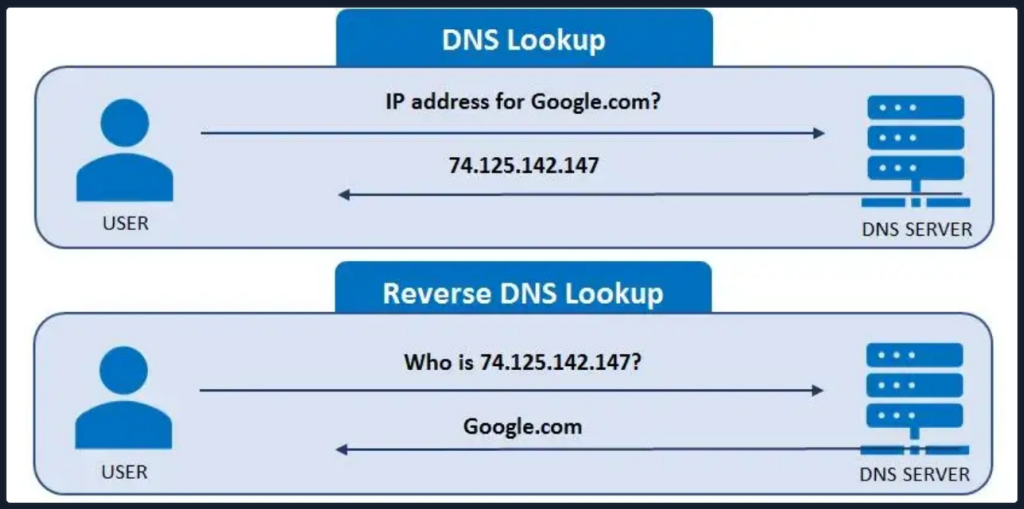
DNS Reverse Lookups/Queries

On the opposite side, we have Reverse Lookups. These occur when a client already knows the IP address and wants to find the corresponding FQDN (Fully Qualified Domain Name).

- Request:
  - `What is your name 192.168.10.6?`
- Response:
  - `Well its academy.hackthebox.com :)`

In this case the steps are a bit less complicated.

Step	Description
1. Query Initiation	The client sends a DNS reverse query to its configured DNS resolver (server) with the IP address it wants to find the domain name.
2. Reverse Lookup Zones	The DNS resolver checks if it is authoritative for the reverse lookup zone that corresponds to the IP range as determined by the received IP address. Aka 192.0.2.1, the reverse zone would be 1.2.0.192.in-addr.arpa
3. PTR Record Query	The DNS resolver then looks for a PTR record on the reverse lookup zone that corresponds to the provided IP address.
4. Response	If a matching PTR is found, the DNS server (resolver) then returns the FQDN of the IP for the client.



DNS Record Types

DNS has many different record types responsible for holding different information. We should be familiar with these, especially when monitoring DNS traffic.

Record Type	Description
A (Address)	This record maps a domain name to an IPv4 address
AAAA (Ipv6 Address)	This record maps a domain name to an IPv6 address
CNAME (Canonical Name)	This record creates an alias for the domain name. Aka hello.com = world.com
MX (Mail Exchange)	This record specifies the mail server responsible for receiving email messages on behalf of the domain.
NS (Name Server)	This specifies an authoritative name servers for a domain.
PTR (Pointer)	This is used in reverse queries to map an IP to a domain name

TXT (Text)	This is used to specify text associated with the domain
SOA (Start of Authority)	This contains administrative information about the zone

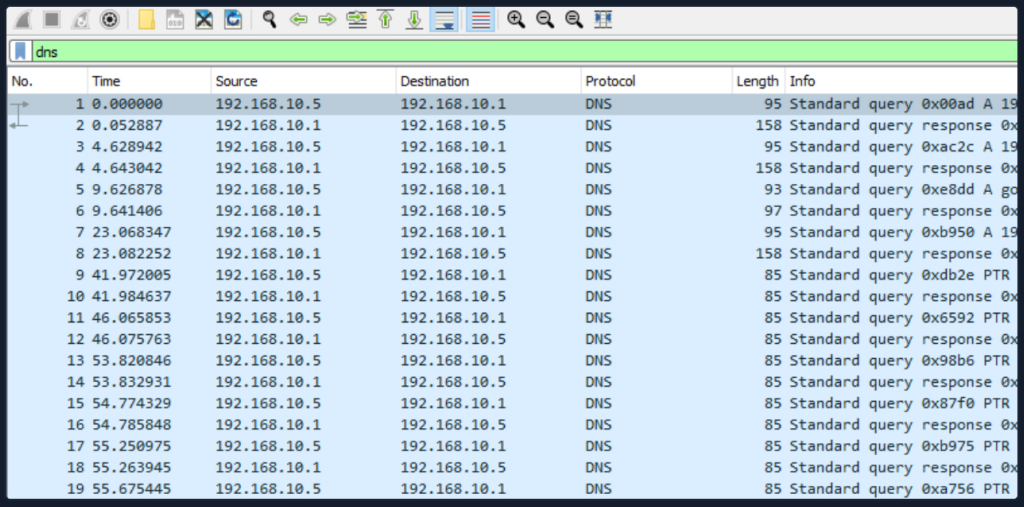
Finding DNS Enumeration Attempts

Related PCAP File(s):

- `dns_enum_detection.pcapng`

We might notice a significant amount of DNS traffic from one host when we start to look at our raw output in Wireshark.

- `dns`



dns

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.10.5	192.168.10.1	DNS	95	Standard query 0x00ad A 19
2	0.052887	192.168.10.1	192.168.10.5	DNS	158	Standard query response 0x
3	4.628942	192.168.10.5	192.168.10.1	DNS	95	Standard query 0xac2c A 19
4	4.643042	192.168.10.1	192.168.10.5	DNS	158	Standard query response 0x
5	9.626878	192.168.10.5	192.168.10.1	DNS	93	Standard query 0xe8dd A go
6	9.641406	192.168.10.1	192.168.10.5	DNS	97	Standard query response 0x
7	23.068347	192.168.10.5	192.168.10.1	DNS	95	Standard query 0xb950 A 19
8	23.082252	192.168.10.1	192.168.10.5	DNS	158	Standard query response 0x
9	41.972005	192.168.10.5	192.168.10.1	DNS	85	Standard query 0xdb2e PTR
10	41.984637	192.168.10.1	192.168.10.5	DNS	85	Standard query response 0x
11	46.065853	192.168.10.5	192.168.10.1	DNS	85	Standard query 0x6592 PTR
12	46.075763	192.168.10.1	192.168.10.5	DNS	85	Standard query response 0x
13	53.820846	192.168.10.5	192.168.10.1	DNS	85	Standard query 0x98b6 PTR
14	53.832931	192.168.10.1	192.168.10.5	DNS	85	Standard query response 0x
15	54.774329	192.168.10.5	192.168.10.1	DNS	85	Standard query 0x87f0 PTR
16	54.785848	192.168.10.1	192.168.10.5	DNS	85	Standard query response 0x
17	55.250975	192.168.10.5	192.168.10.1	DNS	85	Standard query 0xb975 PTR
18	55.263945	192.168.10.1	192.168.10.5	DNS	85	Standard query response 0x
19	55.675445	192.168.10.5	192.168.10.1	DNS	85	Standard query 0xa756 PTR

We might even notice this traffic concluded with something like `ANY`:

32	93.113760	192.168.10.5	192.168.10.1	DNS	121 Standard query 0x2e28 ANY 192.168.10.1 OPT
38	103.122114	192.168.10.5	192.168.10.1	DNS	121 Standard query 0x90c9 ANY 192.168.10.1 OPT
45	113.128223	192.168.10.5	192.168.10.1	DNS	121 Standard query 0x9c2a ANY 192.168.10.1 OPT
50	113.209771	192.168.10.1	192.168.10.5	DNS	97 Standard query response 0x2e28 Refused ANY 192.168.10.1

This would be a clear indication of DNS enumeration and possibly even subdomain enumeration from an attacker.

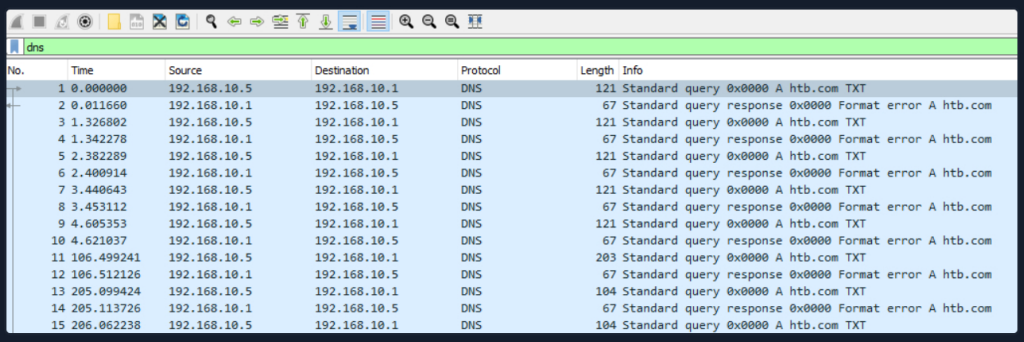
Finding DNS Tunneling

Related PCAP File(s):

- `dns_tunneling.pcapng`

On the other hand, we might notice a good amount of text records from one host. This could indicate DNS tunneling. Like ICMP tunneling, attackers can and have utilized DNS forward and reverse lookup queries to perform data exfiltration. They do so by appending the data they would like to exfiltrate as a part of the TXT field.

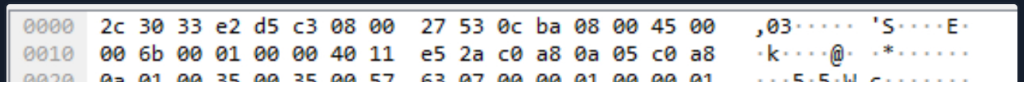
If this was happening it might look like the following.



dns

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.10.5	192.168.10.1	DNS	121	Standard query 0x0000 A htb.com TXT
2	0.011660	192.168.10.1	192.168.10.5	DNS	67	Standard query response 0x0000 Format error A htb.com
3	1.326802	192.168.10.5	192.168.10.1	DNS	121	Standard query 0x0000 A htb.com TXT
4	1.342278	192.168.10.1	192.168.10.5	DNS	67	Standard query response 0x0000 Format error A htb.com
5	2.382289	192.168.10.5	192.168.10.1	DNS	121	Standard query 0x0000 A htb.com TXT
6	2.400914	192.168.10.1	192.168.10.5	DNS	67	Standard query response 0x0000 Format error A htb.com
7	3.440643	192.168.10.5	192.168.10.1	DNS	121	Standard query 0x0000 A htb.com TXT
8	3.453112	192.168.10.1	192.168.10.5	DNS	67	Standard query response 0x0000 Format error A htb.com
9	4.605353	192.168.10.5	192.168.10.1	DNS	121	Standard query 0x0000 A htb.com TXT
10	4.621037	192.168.10.1	192.168.10.5	DNS	67	Standard query response 0x0000 Format error A htb.com
11	106.499241	192.168.10.5	192.168.10.1	DNS	203	Standard query 0x0000 A htb.com TXT
12	106.512126	192.168.10.1	192.168.10.5	DNS	67	Standard query response 0x0000 Format error A htb.com
13	205.099424	192.168.10.5	192.168.10.1	DNS	104	Standard query 0x0000 A htb.com TXT
14	205.113726	192.168.10.1	192.168.10.5	DNS	67	Standard query response 0x0000 Format error A htb.com
15	206.062238	192.168.10.5	192.168.10.1	DNS	104	Standard query 0x0000 A htb.com TXT

If we were to dig a little deeper, we might notice some out of place text on the lower right-hand side of our screen.



```
0000  2c 30 33 e2 d5 c3 08 00  27 53 0c ba 08 00 45 00   ,03····· 'S····E·
0010  00 6b 00 01 00 00 40 11  e5 2a c0 a8 0a 05 c0 a8   ·k····@· ·*······
0020  0a 01 00 35 00 35 00 57  63 07 00 00 01 00 00 01   ···5·5·W c·······
```

```
0030  00 01 00 00 00 00 03 68  74 62 03 63 6f 6d 00 00   ·······h tb·com··
0040  01 00 01 03 68 74 62 03  63 6f 6d 00 00 10 00 01   ····htb· com·····
0050  00 00 00 0a 00 23 22 48  54 42 7b 54 68 69 73 20   ·····#"H TB{This
0060  69 73 20 6b 69 6e 64 20  6f 66 20 6d 61 6c 69 73   is kind  of malis
0070  63 69 6f 75 73 20 3b 29  7d                       cious ;) }
```

However, in many cases, this data might be encoded or encrypted, and we might notice the following.

```
0000  2c 30 33 e2 d5 c3 08 00  27 53 0c ba 08 00 45 00   ,03····· 'S····E·
0010  00 bd 00 01 00 00 40 11  e4 d8 c0 a8 0a 05 c0 a8   ······@· ········
0020  0a 01 00 35 00 35 00 a9  8d b6 00 00 01 00 00 01   ···5·5·· ········
0030  00 01 00 00 00 00 03 68  74 62 03 63 6f 6d 00 00   ·······h tb·com··
0040  01 00 01 03 68 74 62 03  63 6f 6d 00 00 10 00 01   ····htb· com·····
0050  00 00 00 0a 00 75 74 56  54 42 61 55 31 45 79 56   ·····utV TBaU1EyV
0060  58 68 61 53 46 70 72 56  6a 4e 6f 63 6c 64 45 54   XhaSFprV jNocldET
0070  6e 4e 6b 62 56 4a 58 54  31 63 78 61 55 30 77 62   nNkbVJXT 1cxaU0wb
0080  33 70 58 56 6d 68 4c 59  54 46 6e 65 55 31 58 65   3pXVmhLY TFneU1Xe
0090  46 6c 4e 4d 55 70 32 57  56 5a 6f 54 31 70 74 54   FlNMUp2W VZoT1ptT
00a0  6b 6c 54 62 58 68 72 55  30 5a 4a 4d 56 64 45 54   klTbXhrU 0ZJMVdET
00b0  6b 4e 6a 4d 58 42 59 55  6d 35 77 59 51 70 58 52   kNjMXBYU m5wYQpXR
00c0  45 4a 4d 51 32 63 39 50  51 6f 3d                 EJMQ2c9P Qo=
```

We can retrieve this value from wireshark by locating it like the following and right-clicking the value to specify to copy it.

```
▾ Domain Name System (query)
    Transaction ID: 0x0000
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▸ Queries
  ▾ Answers
    ▾ htb.com: type TXT, class IN
        Name: htb.com
        Type: TXT (Text strings) (16)
        Class: IN (0x0001)
        Time to live: 10 (10 seconds)
        Data length: 117
        TXT Length: 116
        TXT: VTBaU1EyVXhaSFprVjNocldETnNkbVJXT1cxaU0wb3pXVmhLYTFneU1XeFlNMUp2WVZoT1ptTklTbXhrU0ZJMVdETkNjMXBYUm5wYQpXREJMQ2c9PQo=
    [Response In: 12]
```

Then if we were to go into our Linux machine, in this case we could utilize something like `base64 -d` to retrieve the true value.

Peculiar DNS Traffic

```
MisaelMacias@htb[/htb]$ echo 'VTBaU1EyVXhaSFprVjNocldETnNkbVJXT1cxaU0wb3pXVmhLYTFneU1XeFlNMUp2WVZoT

U0ZSQ2UxZHZkV3hrWDNsdmRWOW1iM0ozWVhKa1gyMWxYM1JvYVhOZmNISmxkSFI1WDNCc1pXRnpaWDBLCg==
```

However, in some cases attackers will double if not triple encode the value they are attempting to exfiltrate through DNS tunneling, so we might need to do the following.

Peculiar DNS Traffic

```
MisaelMacias@htb[/htb]$ echo 'VTBaU1EyVXhaSFprVjNocldETnNkbVJXT1cxaU0wb3pXVmhLYTFneU1XeFlNMUp2WVZoT
```

However, we might need to do more than just base64 decode these values, as in many cases as mentioned these values might be encrypted.

Attackers might conduct DNS tunneling for the following reasons:

Step	Description
1. Data Exfiltration	As shown above DNS tunneling can be helpful for attackers trying to get data out of our network without getting caught.
2. Command and Control	Some malware and malicious agents will utilize DNS tunneling on compromised systems in order to communicate back to their command and control servers. Notably, we might see this method of usage in

	botnets.
3. Bypassing Firewalls and Proxies	DNS tunneling allows attackers to bypass firewalls and web proxies that only monitor HTTP/HTTPs traffic. DNS traffic is traditionally allowed to pass through network boundaries. As such, it is important that we monitor and control this traffic.
4. Domain Generation Algorithms (DGAs)	Some more advanced malware will utilize DNS tunnels to communicate back to their command and control servers that use dynamically generated domain names through DGAs. This makes it much more difficult for us to detect and block these domain names.

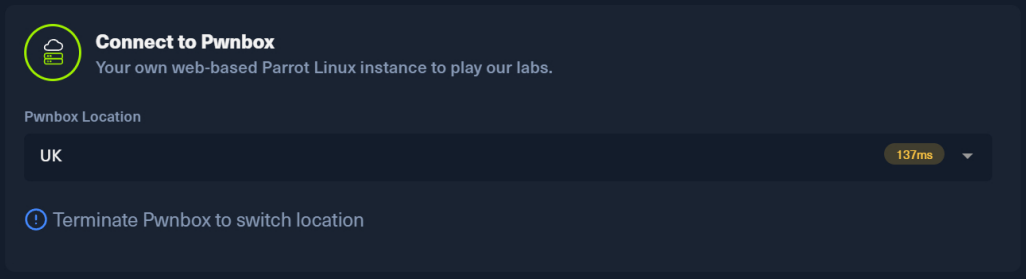
The Interplanetary File System and DNS Tunneling

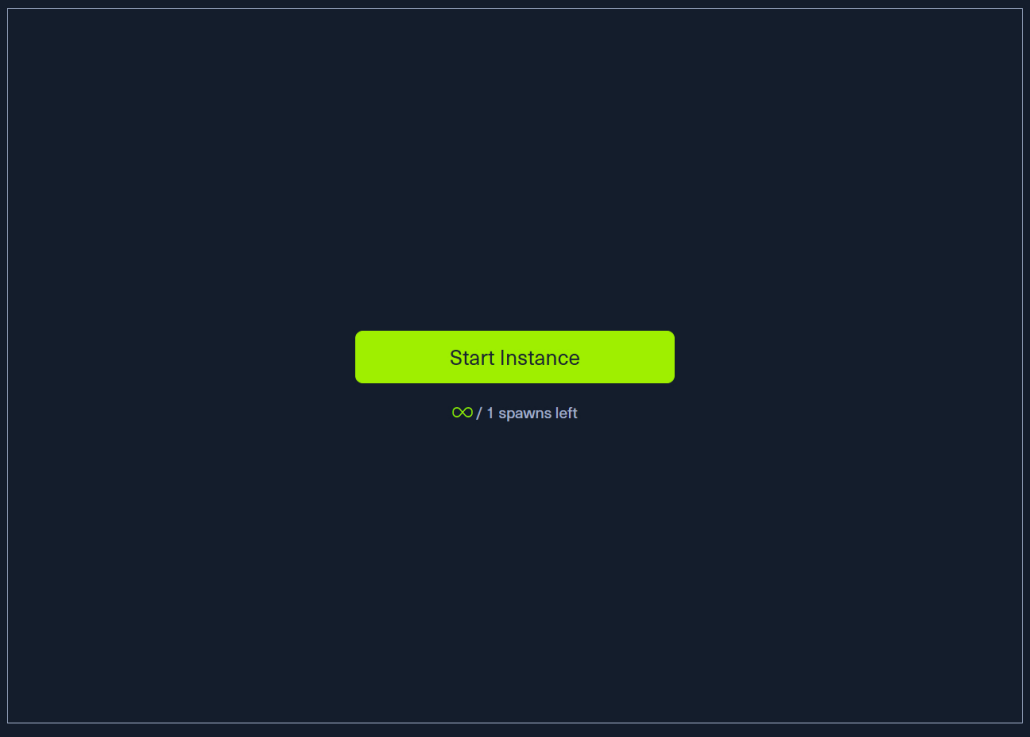
It has been observed in recent years that advanced threat actors will utilize the Interplanetary file System to store and pull malicious files. As such we should always watch out for DNS and HTTP/HTTPs traffic to URIs like the following:

- `https://cloudflare-ipfs.com/ipfs/QmS6eyoGjENZTMxM7UdqBk6Z3U3TZPAVeJXdgp9VK4o1Sz`

These forms of attacks can be exceptionally difficult to detect as IPFS innately operates on a peer to peer basis. To learn more, we can research into IPFS.

Interplanetary File System





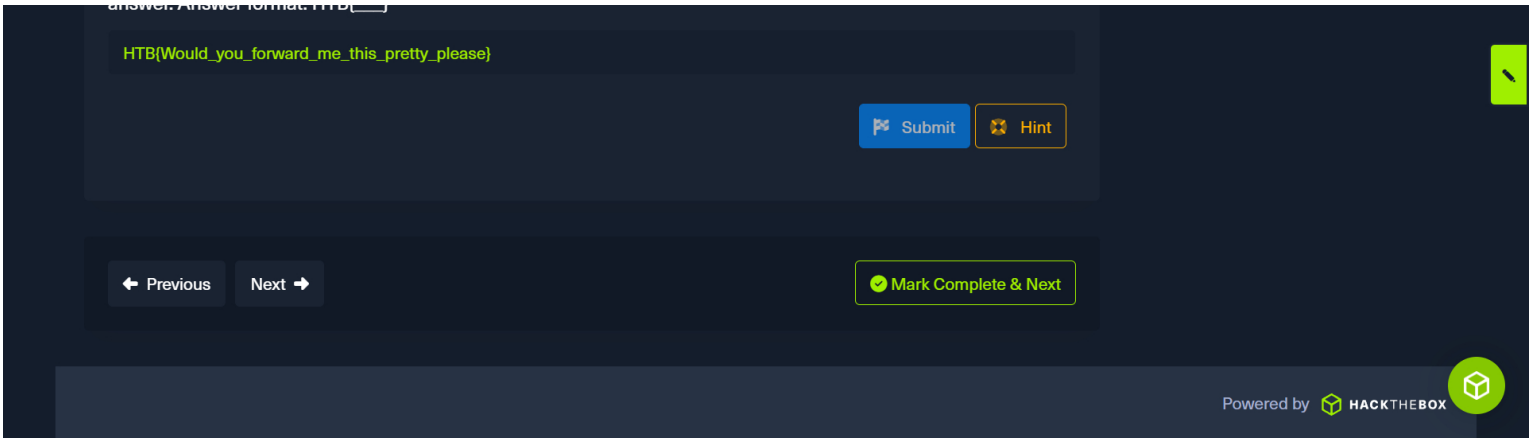
Waiting to start...

Enable step-by-step solutions for all questions

Questions

Answer the question(s) below to complete this Section and earn cubes!

+ 1 Enter the decoded value of the triple base64-encoded string that was mentioned in this section as your answer. Answer format: HTB{ ... }


HTB{Would_you_forward_me_this_pretty_please}
Submit
Hint
Previous
Next
Mark Complete & Next
Powered by HACKTHEBOX