Level 1.2 - DNS resolving of a known malware domain

Previously our IOC (Indicator of compromise) was a known malicious IP. This time, the IOC is a known malicious domain name.

Password

`0j3x0h3sxf`

Instructions

1. Level 1.2 can be found at the attacker's UI
2. As before - execute both malicious and non-malicious trigger, and understand how to identify the malicious trigger
    - We can define it as "DNS query packets that have the question domain name set to 'virus.com'"
    - An example alert output could be: `*ALERT*: DNS query for malicious domain virus.com from 10.0.0.1`
3. Build off your code from part 1.1 to implement a detection for malicious DNS packets
    - Define a new function, detect*malicious*dns(), that implements this detection
4. Remember to call detect*malicious*dns() in the wrapper function detect*malicious*traffic()!
    - Make sure to check for a DNS layer (similar to the IP detection)
    - Also ensure that that you only inspect DNS queries! (Hint: DNS qr)
5. As before, the NIDS should be updatable - so implement a list of known malicious domains and check them all (even though we only know one domain right now)

Notes

GUIDING QUESTIONS Use the guiding questions below to identify the malicious trigger and structure your code's detection:

- Which specific packet contains the DNS query?
  - Are there specific characteristics of this packet e.g. flags, keywords, etc.?
- How can you extract the DNS query from the packet?
  - What is the layer structure of DNS for Scapy?

To submit

Submit file `main.py`.

Sentinel