Mission 3 README

Jared Grimm and Roger Wirkala

File Explanations

cc.py - This is the command and control python program. In order for this to work it must be run on one of the three command and control bots (10.4.9.1, 10.4.9.3, and 10.4.9.4). This program accepts two ASCII characters and encodes them into the TTL field of the DNS records for the malicious host names. This would be the program run by the botmaster to propagate commands through his botnet.

getTTL.py - this program is run on every bot, and its job is to receive messages from the command and control bot. This program makes DNS queries for malicious hostnames on the infected command and control machines. Once it sees that the TTL field has been updated it will decode the TTL value it pulls to determine the ASCII characters sent. If these characters match a command the bot will execute the command. If the listening bot also happens to be a command and control enabled bot (have BIND9 installed). It will update its TTL to match the message it received to further propagate the message throughout the botnet.

Regulator.py - This is the defense of the project, it is located on 10.4.9.2, which is the router of the network, and sees all traffic. It utilizes netfilterQueue, and Scapy to filter and edit packets. Its main functionality is to extract the TTL of all DNS Resource Records and “regulate” them by rounding them to the nearest minute. It filters all DNS Resource Records into a queue, analyzes the packets in the queue, generates a new TTL, and constructs an identical packet with the new TTL to route to the intended destination.

How to Execute

```
sudo python cc.py <dns record file> <char1> <char2>
Example: sudo python cc.py /etc/bind/zones/db.innocent.com w m
            // This would wake any bots listening to innocent.com
```

```
sudo python getTTL.py <hostname> <dns server>
Example: sudo python getTTL.py innocent.com 10.4.9.1
            // This would listen for any messages sent through innocent.com on machine 10.4.9.1
```

```
sudo python  regulator.py
         //This would immediately filter any DNSRR packets on the network and round their TTL
to the nearest minute
```