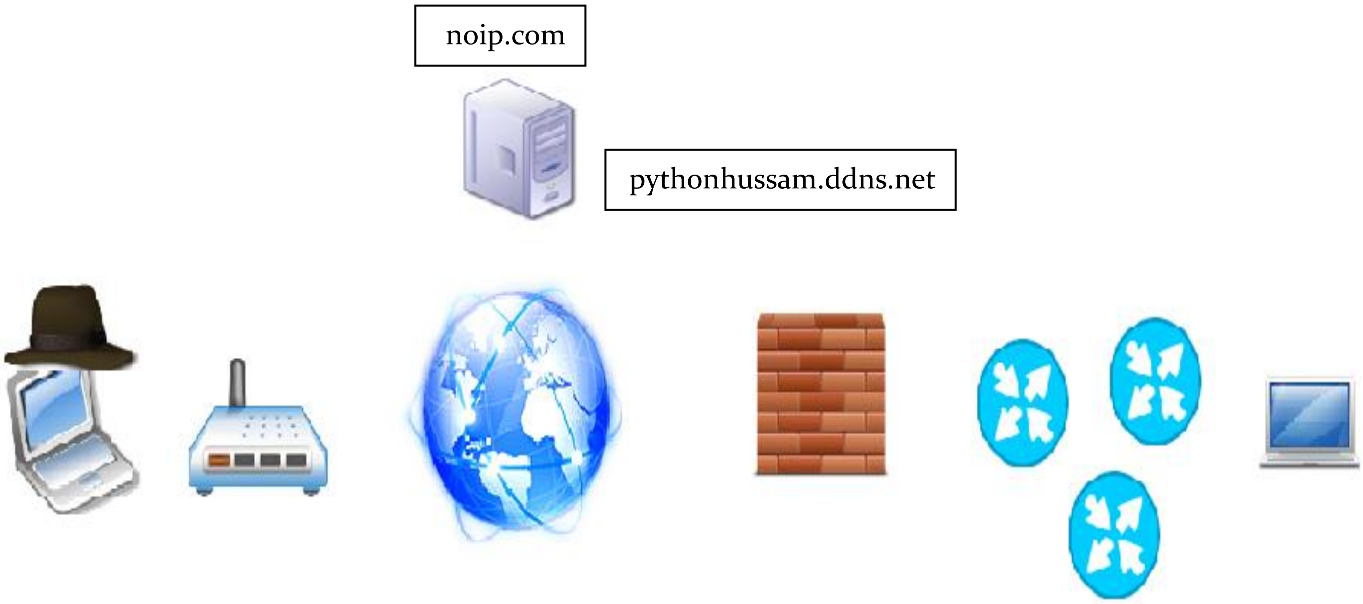
Module 3

Dynamic DNS (DDNS) Outline



Installing noip agent

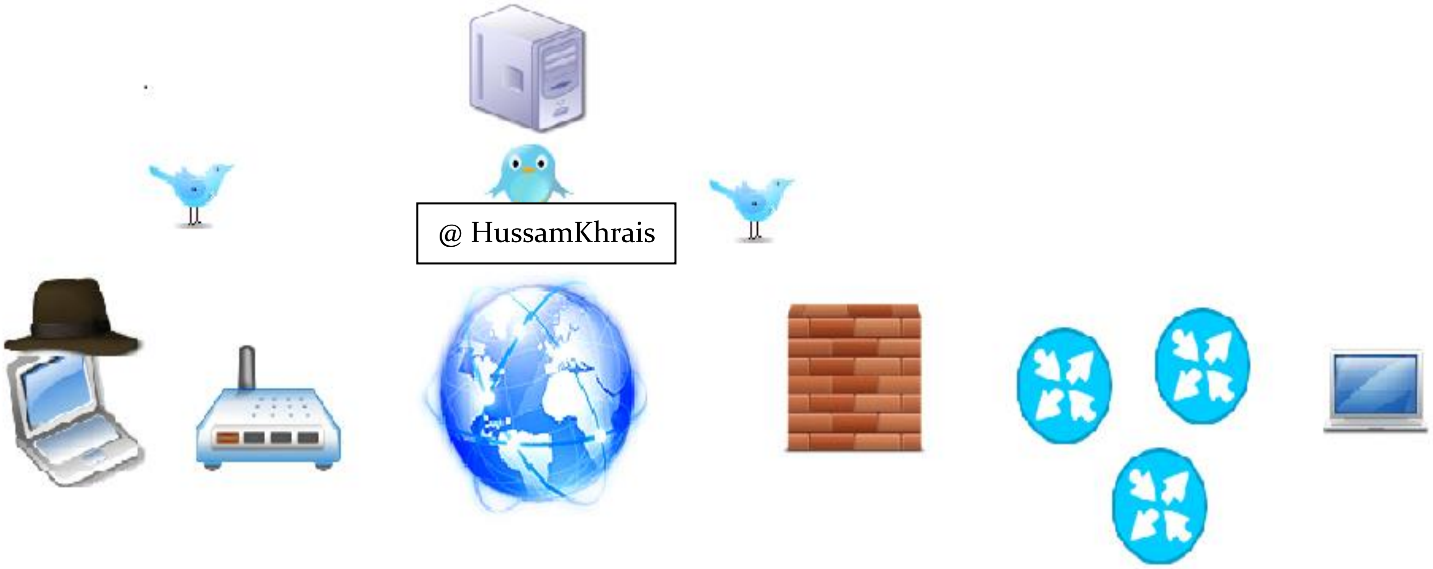
http://www.noip.com/support/knowledgebase/installing-the-linux-dynamic-update-client-on-ubuntu/

Module 3

Interacting with Twitter

Case study – Russian Malware

https://www2.fireeye.com/APT29-HAMMERTOSS-WEB-2015-RPT.html



Module 3

Parsing Tweets in 3 Lines!

BeautifulSoup 3.2.1 download link

https://pypi.python.org/pypi/BeautifulSoup

Case study – Russian Malware

https://www2.fireeye.com/APT29-HAMMERTOSS-WEB-2015-RPT.html

Module 3 – Exercise

Tweet your Kali IP:Port

*Avoid using your personal twitter account during penetration testing.

1. Create a new twitter account for testing purposes.
2. Tweet your kali IP address and a port number of your choice.
3. From the Windows (Target) machine retrieve the HTML of https://twitter.com/<YourAccount> and parse the tweet which you have just created.
4. Use our previous persistent reverse TCP shell (which we created in module 2) and instead of hardcoding the IP address of the Kali, make it dynamically changing based on the latest tweet.
5. Verify that the connection to the kali is successful.
6. Restart the windows PC and make sure that all other features like persistency, number of connections attempts are working as expected.
7. Which one is better, DDNS or using Twitter and why?

Module 3 – Countermeasures

This is NOT an easy task to do!

- We can block twitter but ….
  - What if there's a business needs
  - It's not only limited to twitter!

- We can terminate SSL and see the traffic in clear text but..
  - How many resources do we need to check each single packet going back and forth from our network to twitter?
  - How to distinguish between the good and bad one?
  - What if the Tweet itself was encrypted?
  - What if the attacker adds more 100 fake account to mislead anyone watching traffic?
  - What if the attacker tweet another IP to create a chain of connections?

✓ Think again, how could that malicious software reach our internal network in the first place? **Yes the same countermeasures for module 2 are still valid here.**

Module 3 – Exercise

Enhance Your Scanner

1. Modify your script so it can support UDP scanning.

2. Add a new method to detect if the client is alive or not, like generate an icmp ping first.

3. Control the scanning speed by adding sleep timer between each scan activity.

4. Search why we can't create a stealth scanner, in other words why we can't generate a TCP RST once we got a SYN ACK from the scanned machine?