1. **Background**
   1. **Botnet**:
      1. **Definition**: [1]

* **Botnet:** A network of *bots* under the control of a *botmaster*. *Botnets* recruit vulnerable machines using basic methods of malware distribution. The botmaster controls his bots by using *command and control (C&C) channels*.
* **Bot:** Infected end-host.
* **Botmaster:** The human operator of a *botnet*.
* **Command and control channel:** These channels are used by the *botmaster* to distribute his commands to his bots. These channels can use multiple communication mechanisms like P2P or Internet Relay Chat (IRC). A vast majority are using IRC.
  + 1. **Revenue Models**: [2]

**How can revenue be generated using botnets (attacker and master):**

* renting and trading of botnets
* distributed denial-of-service attacks (DDoS)
* SMTP mail relays for spam (Spambot)
* ad click fraud
* theft of application serial numbers, login IDs, and financial information such as credit card numbers and bank accounts
* etc.
  + 1. **Malware (technical)**:

**What is Malware?**A malware instance is a program that has malicious intent. Examples of such programs include viruses, trojans, and worms. [3]

**How does it spread? (technical aspect) [4]**

* **Viruses** are pieces of malicious code that attach to host programs and propagate when an infected program executes.
* **Worms** are particular to networked computers. Instead of attaching to a host program, worms carry out programmed attacks to jump from machine to machine across the network.
* **Trojan Horses**, like viruses, hide malicious intent inside a host program that appears to do something useful.
* **Attack scripts** are programs written by experts that exploit weaknesses, usually across the network, to carry out an attack. Attack scripts exploiting buffer overflows by “smashing the stack” are the most commonly encountered variety.
* **Java attack applets** are programs embedded in Web pages that achieve foothold through a Web browser.
* **Dangerous ActiveX controls** are program components that allow a malicious code fragment to control applications or the operating system.

The distinctions between these categories have been bleeding together, making classification difficult.

**How does command and control work? [5]**Many existing botnet C&Cs are based on IRC (Internet Relay Chat) protocol, which provides a centralized com- mand and control mechanism. The botmaster can interact with the bots (e.g., issuing commands and receiving responses) in real-time by using IRC PRIVMSG messages. This simple IRC-based C&C mechanism has proven to be highly successful and has been adopted by many botnets. There are also a few botnets that use the HTTP protocol for C&C. HTTP-based C&C is still centralized, but the botmaster does not directly interact with the bots using chat-like mechanisms. Instead, the bots periodically contact the C&C server(s) to obtain their commands.

**Where do such services get offered? [\*]**Entire IRC networks—networks, not just single servers are dedicated to the underground economy. There are 35 to 40 particularly active servers, all of which are easy to find. Furthermore, IRC isn’t the only Internet vehicle they use. Other conduits include, but are not limited to, HTTP, Instant Messaging, and Peer-to-Peer (P2P). Increasingly, many of the miscreants utilize encryption in these services, such as VPNs or SSL. […] The miscreants in the underground economy are typically self-policing. Each IRC network will normally have a channel, such as #help or #rippers, dedicated to the reporting of those who are known to conduct fraudulent deals. The operators of these networks will ban the nicknames of those who have a proven history of fraud. This is a form of self-regulation that ensures the sellers and buyers have a “pleasant” experience and attempts to elicit repeat visits. The miscreants keep meticulous records of those who have defrauded them, and they are quick to share those records with everyone. As with all criminal societies, there is a fair amount of fraudulent dealings and “ripping” (bad business deals).

**Architecture: Centralized, decentralized, unstructured**The oldest type of topology is the **centralized model**. In this model, one central point is responsible for exchanging commands and data between the BotMaster and Bots. Many well-known Bots, such as AgoBot, SDBot, Zotob and RBot used this model. In this model, BotMaster chooses a host (usually high bandwidth computer) to be the central point (Command-and-Control) server of all the Bots. The C&C server runs certain network services such as IRC or HTTP. The main advantage of this model is small message latency which cause BotMaster easily arranges Botnet and launch attacks. [6]

**Decentralized:** […] attackers exploit the idea of Peer-to-Peer (P2P) communication as a Command-and-Control (C&C) pattern which is more resilient to failure in the network. The P2P based C&C model will be used dramatically in Botnets in the near future, and definitely Botnets that use P2P based C&C model impose much bigger challenge for defense of networks. Since P2P based communication is more robust than Centralized C&C communication, more Botnets will move to use P2P protocol for their communication.   
In the P2P model, as shown in Fig. 5, there is no Centralized point for communication. Each Bot keeps some connections to the other Bots of the Botnet. Bots act as both Clients and servers. A new Bot must know some addresses of the Botnet to connect there. If Bots in the Botnet are taken offline, the Botnet can still continue to operate under the control of BotMaster. [6]

**Hybrid:** [6]

**Unstructured:** A botnet communication system could also be based on the principle that no single bot knows about any more than one other bot. In this topology a bot or controller that wanted to send a message would encrypt it and then randomly scan the Internet and pass along the message when it detected another bot. The design of such a system would be relatively simple and the detection of a single bot would never compromise the full botnet. However, the message latency would be extremely high, with no guarantee of delivery. In addition, the random probing behavior could be detectable. [7]

**DDoS Boosters (Memcached?) [8]**

* 1. **DDoS Defense Systems (Life inside a NOC - Network Operation Center)**:
     1. **Detection**:

**Honeynets/Honeypots etc. (economic detection) [2]**

**Examples of systems/methods [9]**

* + 1. **Prevention**:
* Redundancy
* Example: CloudFlare [10]
* Honeynets/Honeypots etc. (economic prevention) [2]
* examples of systems/methods [9]
  + 1. **Reaction**:
* Example: GitHub attack 2018 [11]–[14]
* examples of systems/methods

[1] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, “A multifaceted approach to understanding the botnet phenomenon,” in *Proceedings of the 6th ACM SIGCOMM on Internet measurement - IMC ’06*, Rio de Janeriro, Brazil, 2006, p. 41.

[2] Z. Li, Q. Liao, and A. Striegel, “Botnet Economics: Uncertainty Matters,” in *Managing Information Risk and the Economics of Security*, M. E. Johnson, Ed. Boston, MA: Springer US, 2009, pp. 245–267.

[3] M. Christodorescu, S. Jha, S. A. Seshia, D. Song, and R. E. Bryant, “Semantics-aware malware detection,” in *2005 IEEE Symposium on Security and Privacy (S P’05)*, 2005, pp. 32–46.

[4] G. McGraw and G. Morrisett, “Attacking Malicious Code: A Report to the Infosec Research Council,” *IEEE Software*, vol. 17, no. 5, pp. 33–41, Sep. 2000.

[5] G. Gu, J. Zhang, and W. Lee, “BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic,” p. 19.

[6] H. R. Zeidanloo and A. A. Manaf, “Botnet Command and Control Mechanisms,” in *2009 Second International Conference on Computer and Electrical Engineering*, 2009, vol. 1, pp. 564–568.

[7] E. Cooke, F. Jahanian, and D. McPherson, “The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets,” Jul-2005. [Online]. Available: https://www.usenix.org/legacy/event/sruti05/tech/full\_papers/cooke/cooke\_html/. [Accessed: 30-Oct-2019].

[8] “Memcached DDoS Attack,” *Cloudflare*. [Online]. Available: https://www.cloudflare.com/learning/ddos/memcached-ddos-attack/. [Accessed: 30-Oct-2019].

[9] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, p. 39, Apr. 2004.

[10] “What is Cloudflare?,” *Cloudflare*. [Online]. Available: https://www.cloudflare.com/learning/what-is-cloudflare/. [Accessed: 30-Oct-2019].

[11] “A 1.3-Tbs DDoS Hit GitHub, the Largest Yet Recorded | WIRED.” [Online]. Available: https://www.wired.com/story/github-ddos-memcached/. [Accessed: 30-Oct-2019].

[12] “Famous DDoS Attacks,” *Cloudflare*. [Online]. Available: https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/. [Accessed: 30-Oct-2019].

[13] “February 28th DDoS Incident Report,” *The GitHub Blog*, 01-Mar-2018. .

[14] “The world’s largest DDoS attack took GitHub offline for fewer than 10 minutes | TechCrunch.” [Online]. Available: https://techcrunch.com/2018/03/02/the-worlds-largest-ddos-attack-took-github-offline-for-less-than-tens-minutes/. [Accessed: 30-Oct-2019].

[\*] Thomas, Rob, and Jerry Martin. 2006. "The Underground Economy: Priceless." USENIX ;login, 31(6): 7-16.