ZAMBIA INFORMATION COMMUNICATION TECHNOLOGY (ICT) JOURNAL

Volume 1 (Issue 1) (2017) Pages 63-68

Identifying Botnets Intrusion & Prevention – A Review

Lubasi Kakwete Musambo[1]
Melissa K. Chinyemba[2]
School of Engineering
Dept. of Electrical & Electronics Engineering
The University of Zambia
Lusaka, Zambia
[1]*lubasimusambo@gmail.com,*
[2]*kaemelissa@gmail.com.*

Jackson Phiri[3]
School of Natural Sciences
Dept. of Computer Science
The University of Zambia
Lusaka, Zambia
[3] *jackson.phiri@cs.unza.zm*.

***Abstract*—Systems and networks will be compromised, almost always regardless of what network engineers do. It is then paramount to install and manage systems that have a capacity to identify all forms of intrusions and possibly prevent those intrusions and related attacks on a computer network especially bots. Bots are at the midpoint of most network problems because almost all major cybercrimes and breaches can be traced back to them. This paper reviews three ways in which an intrusion may occur on networks with a focus on botnets and ways in which botnet mitigation may be enhanced.**

Keywords— Intrusion, Prevention, Network, Bots, Botnets.

I. Introduction

An intrusion is an activity or a set of actions that attempt to compromise the basic network security goals like confidentiality, integrity and availability of a computing/networking resource by gaining un-authorised access to them and attempting to perform any unauthorised activity during that access. Intrusion Detection is a process of monitoring events that take place in a computer system or a network, analysing them for signs of security threats and violations to security [1]. Intrusion Detection Systems (IDSs) can detect when an attacker has penetrated a system by exploiting an uncorrected or un-correctable flaws possibly prevent all attempts to gain access into or compromise network resources. Intrusion detection and Prevention systems can be combined into what is called Intrusion Detection and Prevention Systems (IDPs). Intrusions occur in many forms and most apparent and devastating effect is an intrusion through a botnet.

II. Intrusion Threats

A. Physical Theft

An elementary intrusion can be seen as a basic theft of an IT resource. A theft of a computer presents [a] an unauthorised access [b] denies access to the legal owner of the IT resource. The effect of this threat is; (i) a direct loss of information contained on the device (ii) direct loss of an IT resource through theft. An extended effect of the second threat is the effect of the use of information contained on the device, for instance if the information contained is of medical nature or records of a high profile individual [2][3]. Other forms of extended effects may be blackmailing the owner of the IT resource into paying huge sums of money in order to recover the resource among others.

B. Abuse of Privileges (The Insider Threat)

Another form of intrusion that can be analysed is the insider threat. In here an individual who is allowed to access organisational resources from within the organisation and has sufficient privileges to use to the resources may abuse his/her authority or privileges and steal organisational information, plant botnet software applications on organisational computers or networks in order to harm the organisation. Other forms in which this type of intrusion or threat may be created if an authorised individual who has sufficient rights and privileges to access organisational resources, decides to deliberately destabilize the organisational networks or computing resources by disabling security features of the network so as to allow harmful applications such as botnets into the organisation. Other forms of this intrusion threats can be the unwarranted disclosure of information by privileged individuals, such information may be sensitive personally identifiable information (PII) such as social security numbers, national identity numbers, remuneration details etc. [4].

C. Unauthorised Access by an Outsider

An outsider is an individual not permitted to have access to organisational computing resources and its related architecture [5]. This can be an individual who does NOT have an employment contract with an organisation or an individual employed by an organisation but not permitted to access certain organisational resources. Should an outsider have access to a computing resources and its related resources, this individual may introduce botnets into the organisation. Outsiders, normally attempt intruding into organisational networks through a dedicated effort of hacking [6] [7]. Attack forms normally take the form of administrative privileges where an attacker may attempt to take charge of a network in some way or an installation of bot malware onto the network.

III. Malware Infection

Malware stands for **mal**icious soft**ware**. Malware is the vital enabler for cybercrime which poses a severe threat to the globe. Various forms of malware in form of bots or botnets may infect and injure a network's operations.

1) *Classifying Malware*

Malware can be classified as follows [8] [9] [10] [11]:

Fast flux: This is used by botnets to conceal phishing and malware distribution sites behind a continuously changing network of compromised host systems utilized as proxies.

0-Day: An unknown or undiscovered vulnerability in an application that once discovered by an attacker causes harm to a computing resource.

BackDoor: This may occur in a computer system, cryptosystem or algorithm and may be understood as a means of by-passing authentication that secures a computing resource. It has capabilities of securing remote computer access and can access computer resources while remaining to be undetected. A backdoor can be installed on software or hardware.

Crimeware: This type of malware is designed to automate financial crime by performing identity theft to access online accounts of users at financial institutions and online retailers for the express purpose of stealing funds from those accounts or performing unauthorized transactions to the benefit of the thief controlling the crimeware. Crimeware normally exports private information from a network for financial exploitation.

Computer virus: These programs can replicate themselves and infect a computer without the consent or knowledge of a computer user. A virus spreads from one system to another through an executable code when its host is transferred to a target computer such as being sent over Internet, email or transported within the network via removable media such as a USB drive. Infected files residing in a computer network file system increase the chances of spreading a virus infection.

Computer worm: This program sends copies of itself within a computer network without any involvement by a user. A worm doesn't need to attach itself to an existing program in order to spread. Worms may reduce network performance due to their bandwidth consuming abilities.

Email spoofing: A fraudulent email activity in which parts of the email header and the sender address are modified, appearing as if the email was sent from another source. The principle is to conceal the origin of an email message.

Exploit: This can be seen as a portion of software, data, or string of commands that take advantage of a computer bug, glitch or vulnerability disrupting normal behavior on computer software, hardware or other electronic device.

Phishing: A fraudulent process of collecting sensitive information such as usernames, passwords and credit card details by pretending to be a trustworthy entity (network administrator, social media friend etc.) in an electronic communication.

Smishing: Originatess from "**SM**s ph**ISHING**". Smishing uses text messages on cell phones to lure a user into revealing personal information. The method used to actually "capture" user's information in the text message could be a website URL or a phone number that connects to an automated voice response system.

Spamware: Imports thousands of email addresses, generates random email addresses, inserts fraudulent headers into messages, uses multiple mail servers at once, and uses open relays. Spamware can also be used to locate email addresses to build lists for spamming or to sell to spammers.

Spyware: Computer software that is installed on a user's computer without the user's consent with the purpose of collecting information about the user, their computer or browsing habits. Spyware can cause other interference by changing computer settings that slow connection speeds, load different home pages, and lose Internet connectivity or program functionality.

Trojan horse: This is a type of malware that appears to have a normal function but actually conceals malicious functions that it performs without authorized access to the host system. A Trojan can allow the ability to save their files on the user's computer or monitor the user's screen and control his computer.

Botnet: This is a collection of software robots, or bots, that are automatic and self-directed. Botnet malware controls a botnet which is a collection of compromised computing devices called zombies that perform an action(s) commanded by the botnet malware.

Several malware applications exists, however, this paper concentrates on reviewing how botnet malware applications operate and how they may be identified and or mitigated.

IV. Botnet Systems

Botnet are highly publicized due to its enormousness and capability to cause massive financial loss through Distributed Denial of Service (DDoS), phishing, spam or identity theft. Due to its ability to be controlled remotely by a hacker, usually, the command and control (C & C) is achieved via Internet Relay Chat (IRC) channel and peer-to-peer (P2P) connections. Much as the size and danger of botnet threat is recognized globally, the actual number of machines involved is not easy to estimate [12].

1) *Origins of Botnets*

Before botnets, the main motivation for Internet attacks was notoriety and prominence, by contrast botnets are built with proposition of distributing the attacker's control over his victims. This long-term control is accomplished by a bot being crafty during every part of its lifecycle [10] [13]. When a bot is in place, the only obligatory traffic comprises of incoming commands and outgoing responses, constituting the botnet's Command and Control channel. The notion of a remote-controlled computer bot initiates from Internet Relay Chat where compassionate bots were first introduced to help with tedious directorial tasks such as channel and nickname management. The first implementations of such an IRC bot was Eggdrop, initially developed in 1993 and still one of the most popular IRC bots in existence [7].

2) *Botnet Topologies, Protocols & Lifecycle*

Adding to the customary IRC-based botnets, several other protocols and topologies have materialized lately, however, the two main known topologies are centralized and peer-to-peer [7].

A. *Centralized*

Among centralized botnets, IRC is still the leading protocol though this drift is diminishing and several recent bots have used Hypertext Transfer Protocol (HTTP) for their C & C channels. The conception of botnets devises from the idea of improving malware with the ability to connect back to a server upon infection. Initial known cases of centralized botnets appeared in 1998/1999 and were tied to the "Global Threat Bot" (GTBot), the remote access toolkit SubSeven as well as the email worm PrettyPark [14]. Despite its advantage of being easy to implement and producing slight overhead, one flaw remains to this type of architecture from a botmaster's view: Shutting down all central C&C instances takes control away instantly and renders the botnet useless, since bots will endeavour to connect to non-existent servers [7].

B. *Peer-to-Peer*

Among P2P botnets, various protocols exist, but the general idea is to use a decentralized assortment of peers so as to disregard the single point of failure found in centralized botnets. To overcome the drawback of dependence on centralized components, experiments with peer-to-peer (P2P) mechanisms in malware date back as far as 2002 to the Slapper Worm [15]. The advantage of this technology is that the C&C channel is embedded into the botnet architecture, thus significantly contributing to resiliency against countermeasures when used correctly. A game-changing event was the appearance of the Nugache Worm, first detected in 2005 and considered to be responsible for the creation of one of the first botnets with a successfully distributed C&C infrastructure, based on a P2P protocol. Since then, other P2P botnets have been observed and analysed [16].

C. *Botnet life cycle*

Regardless of the topology being used, the typical life cycle of a bot is similar and follows suit as follows [7]:

a. Creation: First, the botmaster develops his bot software often by recycling existing code and adding custom features.

b. Infection: Once a victim machine becomes infected with a bot, it is known as a zombie. The target machine gets infected through:

i. *Software vulnerabilities*: When an attacker exploits a vulnerability in a running service to gain access and install his software without any user interface.

ii. *Download Driven:* When an attacker hosts his files on a Web server and lures user to visit the site which spontaneously installs malicious software on user machine.

iii. *Trojan horse:* When an attacker bundles his malicious software with superficially benign and useful software, such as screen savers, antivirus scanners, or games.

iv. *Email attachment*: When an attacker sends an attachment that will spontaneously install the bot software soon as the user opens it.

c. Rallying: After an infection, the bot starts attempts to contact its C & C server(s) in a process known as rallying. In a centralized botnet, this could be an IRC or HTTP server. In a P2P botnet, the bots perform the bootstrapping protocol required to locate other peers and join the network. Some C & C servers are configured to immediately send some original commands to the bot without botmaster' intervention. In an IRC botnet, this is usually done by including the commands in the C & C channel's topic.

d. Waiting: Having joined the C & C network, the bot waits for commands from the botmaster. During this time, very little (if any) traffic is legitimate between the victim and the C & C servers.

e. Executing: Once the bot receives a command from the botmaster, it executes it and returns any results to the botmaster via the C & C network. Following execution of a command, the bot returns to the waiting state to await further directives. If the victim computer is restarted or loses its link to the C & C network, the bot restarts in the rallying state [7]:

V. The Botnet Business Model

Botnets are still inspired by financial profits. Structured crime groups often use them as a source of income because, they are well-financed organizations that can employ the best minds in computers and network security and offer more improved opportunities than the authentic job market. One outrageous example is the Russian Business Network (RBN), a Russian Internet service provider (ISP) that openly supports cyber-criminal activities. They are accountable for the Storm Worm (Peacomm), the March 2007 DDoS attacks on Estonia, and a high-profile attack on the Bank of India in August 2007, along with many other attacks [17].

Botnets are perfect vehicles for criminal undertakings on the Internet because they provide anonymity and scattered access to the Internet. Not long ago, bots were used for producing Bitcoins (BTC), an experimental digital coinage scheme that was published in 2009 [16]. However, criminals are always inventing new and creative ways to profit from botnets including:

Spam: Spammers send millions of emails advertising data and login information, or running advance-fee schemes such as the Nigerian 419 scam [2].

DDoS and extortion: Having combined a large number of bots, a DDoS attack can be launched for some days [2].

Identity theft: Once a bot has a position on a victim's machine, it has comprehensive control such that keyloggers can be installed to record login credentials and other valuable data and pharming. The attacker can then falsify data [2].

Click fraud: In this setup, bots are used to repeatedly click Web advertising links, generating per-click revenue for the attacker, which is fraud because only the clicks of human users with a legitimate interest are appreciated to advertisers since bots will not buy the merchandise [2].

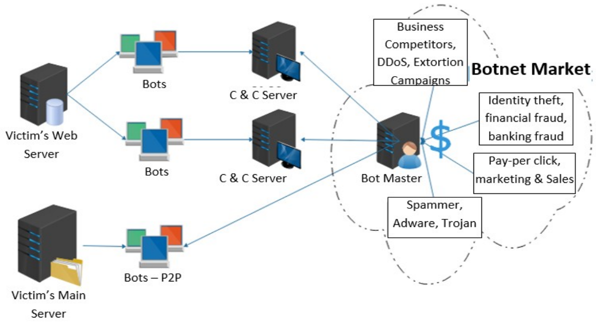

Fig. 2. Botnet business model [18] [19]

VI. Botnet Intrusion Detection And Prevention

1) *Anti-malware Software (formally anti-virus)*

These programs form a first line defense in intrusion defense especially against botnets. These programs analyze files and program files for known patterns (or signatures) similar to known botnet operations. All programs that have malicious signatures (bots) are prevented from running and causing harm to a computer system. The effectiveness of these tools lies on their signature database. If a new bot is present in the application whose signature is not yet known may swim into the organizational network without detection.

Another form of anti-malware examine files for any form of malicious behavior (such as one caused by bots) and then take action to stop the program from running or removing it from the computer. Alternatively an antimalware may be designed to house a white list which is a list of well-known normal software activity. Should running software operate outside these parameters then these are flagged as malicious or bots and either removed or stopped from running. False positives are a weakness in white-list anti-malware applications.

2) *Network-Based Intrusion Detection Systems (NIDS)*

These systems monitor a network for any intrusion and take action. They operate in one of three methods: [a] signature detection [b] anomaly detection and [c] hybrid.

A signature-based NIDS examine all the network traffic that passes through them by studying the TCP/IP packets for signatures of known attacks. These NIDS can also monitor networks for known attacks. Modifying a network packet such as a Transmission Control Protocol / Internet Protocol (TCP/IP) header may fool these systems. They are good at picking network anomalies.

Anomaly based NIDS detect abnormalities in network traffic and build statistical or baseline models for the traffic they monitor. They use this data to flag if any monitored traffic has departed from the statistical observations database.

Hybrid systems combine the best properties of both the signature and anomaly NIDS.

3) *Network-Based Intrusion Prevention Susytems (NIPS)*

These are designed to try and prevent a bot network attack from succeeding. A NIPS device is inserted in-line with the traffic it is monitoring. Each network data packet is examined and let loose if it does not match a threat signature. The biggest challenge of using these is that they rely on threat signature data which may be modified.

4) *Host-Based Intrusion Prevention Systems (HIPS)*

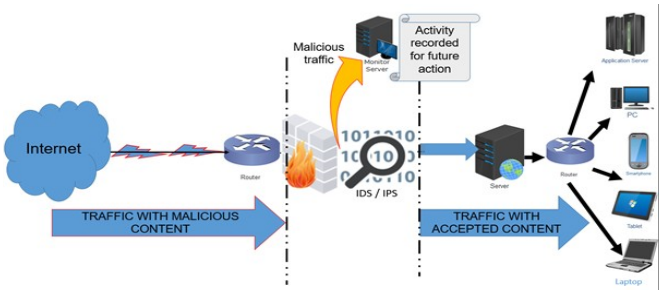

Fig. 2. IDs, IPS Systems architecture [1] [21]

These systems are installed on the protected system (computer or network) to protect it by monitoring and analyzing what other processes on that system are doing at a very detailed level. They observe behavior of installed systems and may use white lists to prevent unauthorized running of systems. They have capabilities of monitoring encrypted network traffic as this traffic would have been decrypted when it reaches the host where the system is installed. Their strength is signature based.

5) *Security Information Management Systems (SIM)*

A SIM system is a centralized database for network data. It collects, collates and organizes information so that information overload is not achieved on a network system. A SIM system performs data normalization also thereby making it easy for analyst to study the data. Information parameters are set up in a SIM to allow it filter out all irrelevant data it reaches. This not only relives a network of excessive traffic but all improves network performance. SIM system reports are used in incident report management to aid in understanding network issues that arise. Bots increase network traffic and a SIM may pick up on this.

6) *Network Session Analysis (NSA)*

During a network communication, time-stamps, nonce, epochs, date, session identifications and frame assembly and fragmentation data is recorded. This information is part of a NSA which can be used in investing an incident on a network (such as an intrusion). Network session data represents a high-level summary of a network communication occurring between computer systems. NSAs can be created by use of honey pots. Honey pots are decoy systems designed to lure a potential attacker away from critical systems. Honey pots have a primary function which is to:

- Divert an attacker from accessing critical systems
- Collect information about the attacker's activity, and
- Encourage the attacker to stay on the system long enough for administrators to respond.

These systems are filled with fabricated information designed to appear valuable but that a legitimate user of the system wouldn't access. Thus, any access to the honey pot is suspected. The system is instrumented with sensitive monitors and event loggers that detect these accesses and collect information about the attacker's activities. Honey nets are a good way to monitor and observe botnet behavior so as to develop solutions towards their attacks.

7) *System Integrity Validation (SIV)*

System integrity validation (SIV) technology is the technology that assess a systems' performance against its rated operational specification so as to understand if any anomaly is present and possibly highlight remedial this is good tool to use in intrusion analysis as it can pick out intrusions. Bots adjust network performance and an SIV may pick up on this.

8) *Botnet Defence*

a. *Detecting and Removing Individual Bots*

The crucial first step in botnet defense is removing individual bots using basic antivirus approach with signature-based detection which is still effective with some bots. However, a more advanced method for eliminating more than one bot at a time using polymorphism is required, because dealing with more sophisticated bot and polymorphic malware detection must be done using behavioral analysis and heuristics [7].

b. *Detecting C & C Traffic*

Botmaster needs to establish a Command and Control (C&C) center to control the zombie machines as in Figure 3.

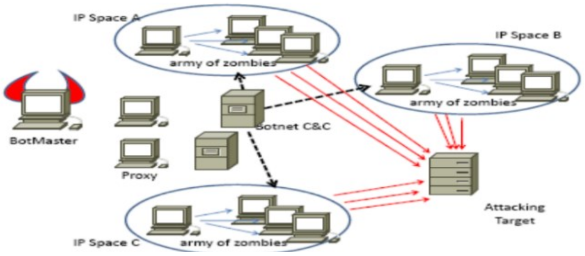

Fig. 3. Creation of Distributed C&C Servers & Attack Launch [7]

To largely mitigate the botnets, network-based detection of the botnet's C & C traffic needs be explored, rather than individual machines. Intrusion Detection System (IDS) technique of anomaly detection to identify unencrypted IRC botnet traffic can be used. Beyond the single network scope, traffic from centralized botnets can be detected at the ISP level based only on transport layer flow statistics. Additionally, it can determine the size of a botnet without joining and can even detect botnets using encrypted C & C.

c. Detecting and Neutralizing the C & C Servers

C & C traffic detection and bot elimination still doesn't sort the entire botnet at once. To achieve this in a centralized botnet, access to the C & C servers must be removed. BotSniffer similar to BotHunter was developed in 2008 [9] an approach that represents several improvements including handling of encrypted traffic, since it doesn't rely only on content inspection to co-relate messages. This approach doesn't require advance knowledge of the bot's signature or the identity of C & C servers. By analyzing network traces, BotSniffer detects the spatial-temporal correlation among C & C traffic belonging to the same botnet. It can therefore detect both the bot members and the C & C server(s) with a low false positive rate [7] [10] [7].

d. Attacking Encrypted C & C Channels

Though some of the approaches can detect encrypted C & C traffic, the presence of encryption makes botnet research and analysis much harder. The first step in dealing with these advanced botnets is to penetrate the encryption that protects the C & C channels [20] [15] [5] [7]. Many encryption schemes that support key exchange like SSL/TLS are susceptible to man-in-the-middle (MITM) attacks. Therefore, the two possible attacks on encrypted C & C channels include: Gray-box analysis, where the bot communicates with a local machine impersonating the C & C server and a full MITM attack, in which the bot communicates with the true C & C server as in Figure 4.

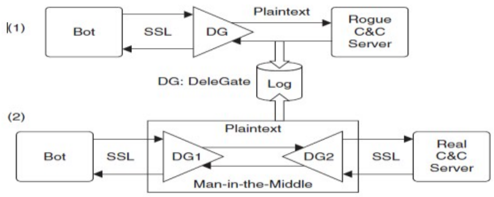

Fig. 4. Setups for MiTM attacks on encrypted C & C channels [7]

The first attack determines the authentication information required to join the live botnet. However, it does not allow the observer to see the interaction with the larger botnet, specifically the botmaster [6].

The second attack reveals the full interaction with the botnet, including all botmaster commands which can allow the observer to literally take over the botnet. He can then log in as the botmaster, issue a command such as Agobot's .bot.remove, to disconnect all bots from botnet and permanently removed from the infected computers. Unfortunately, there are legal issues with this approach because it constitutes unauthorized access to all the botnet computers, despite the fact that it is in fact a benign command to remove the bot software [7].

VII. Botmaster Traceback

The botnet field is quiet challenging with problems such as: encrypted C & C channels, obfuscated binaries, fast-flux proxies protecting central C & C servers, customized communication protocols, and many more as in Figure 5.

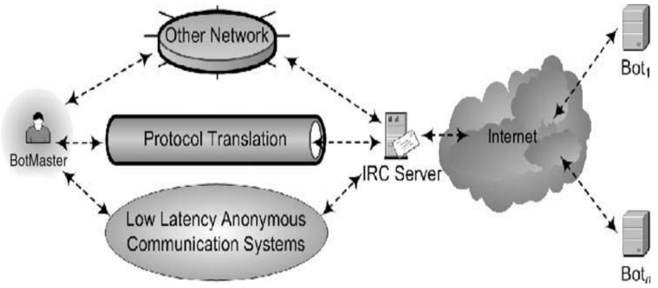

Fig. 5. Botnet C & C traffic laundering [7].

The only permanent solution of the botnet problem is to go after the root cause, being the botmasters, with which the most challenging task is locating them since they are very good at concealing their identities and locations with precautions on multiple levels to ensure that their connections cannot be traced. This is due to the expected disastrous consequences should the trace be successful. As of now, there is no published work that would allow automated botmaster trace back on the Internet, and it remains an open problem [7]. Therefore, the only technique that can help mitigate the Botnet problem is the Intrusion Detection System (IDS) which can identify unencrypted IRC traffic even at ISP level based on transport layer flow statistics.

Traceback Challenges: One way to find the Botmaster is to track the botnet C & C traffic. However, the fact that the botmaster originates the botnet C & C traffic, he hides by disguising his link to the C & C traffic via various traffic-laundering techniques that make tracking C & C traffic more difficult and further conceals his activities by encrypting his traffic to and from the C & C servers. Later on, botmaster only need to be online briefly and send small amounts of traffic to interact with his botnet, reducing the chances of live traceback.

Stepping Stones: These are the intermediate hosts used for traffic laundering. The attacker sets them up in a chain, leading from the botmaster's true location to the C & C server. Stepping stones can be any network redirection services like SSH servers, proxies, IRC bouncers (BNCs) or virtual private network (VPN). These usually run on compromised hosts, which are under the attacker's control and lack audit/logging mechanisms to trace traffic making, manual traceback tedious and time-consuming [7].

The major challenge posed by stepping stones is that all routing information from the previous hop (IP headers, TCP headers, and the like) is stripped from the data before it is sent out on a new separate connection, preserving only the content of the packet, which renders many existing tracing schemes useless.

Low-Latency Anonymous Network: Besides laundering the botnet C & C across stepping stones and different protocols, a sophisticated botmaster could anonymize its C & C traffic by routing it through some low-latency anonymous communication systems. The botmaster could use Tor as a virtual tunnel to anonymize his TCP-based C & C traffic to the IRC server of the botnet as well as utilizing Tor's hidden services to anonymize the IRC server of the botnet [7].

Encryption: Most of the stepping stone chain can be encrypted to protect it against content inspection, which could reveal information about the botnet and botmaster. This can be done using a number of methods, including SSH (Secure Shell) tunneling, SSL/TLS (Secure Socket Layer / Transport Layer Security) enabled BNCs and IPsec tunneling because using encryption defeats all content-based tracing approaches [7] [1] [11].

Traceback Beyond the Internet:

Despite the control measures put to monitor traffic, there are additional traceback challenges beyond the reach of the Internet (see Figure 5). Any IP-based traceback method assumes that the true source IP belongs to the computer being used by the attacker. However, in many scenarios this is not true e.g. Internet-connected mobile phone networks, open wireless (Wi-Fi) networks and public computers, such as those at libraries and Internet cafes. Most modern cell phones support text-messaging services such as Short Message Service (SMS), and many smart phones also have full-featured IM software. Therefore, the botmaster can use a mobile device to control the botnet from any location with cell phone reception using a protocol translation service or a special IRC client for mobile phones [7].

For an IRC botnet, such a service would receive the incoming SMS or IM message, then repackage it as an IRC message and send it on to the C & C server (possibly via more stepping stones), as shown in Figure 6 .

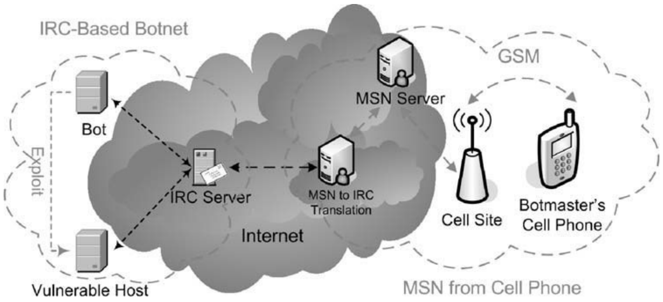

Fig. 6. Using a cell phone to evade Internet-based traceback [7].

To eliminate the need for protocol translation, the botmaster can run a native IRC client on a smart phone with Internet access. However, there are several problems with this approach [7].

To begin with, this trace requires lots of manual work and cooperation of yet another organization, making a real-time trace unlikely. Then the carrier won't be able to determine the name of the subscriber if he is using a prepaid cell phone. Finally, the tracer could obtain an approximate physical location based on cell site triangulation. Even if he can do this in real time, it might not be very useful if the botmaster is in a crowded public place.

VIII. Summary

Intrusion Detection and Prevention Systems are not single tools or products but a series of defence technologies that are applied collectively. To have the most effective intrusion detection and prevention system require investment in a variety of areas from in-line to host or software or hardware based systems. A good systems intrusions policy is needed to guide an organisation on how best to approach intrusions to their networks. Threats must be assessed before rushing to implement an intrusion tool. Intrusion tools work well when combined with detection and prevention tools. To preserve integrity of networks, information security and integrity tools must be implemented.

Botnets are one of the biggest threats to the Internet today, and they are linked to most forms of Internet crimes. A number of botnet countermeasures exist, but most are focused on bot detection and removal at the host and network level. Some approaches exist for Internet-wide detection and disruption of entire botnets, but we still lack effective techniques for combating the root cause of the problem due to the botmasters who conceal their identities and locations behind chains of steppingstone proxies. Short of a perfect solution, even a partial traceback technique could serve as a very effective deterrent for botmasters. With each botmaster that is located and arrested, many botnets can be eliminated at once.

IX. References

[1] W. Stallings, Network Security Essentials, New York: Prentice Hall, 2011.

[2] P. M. Rebecca Bace, *NIST Special Publication on Intrusion Detection Systems,* Carlifornia: infidel, Inc., Scotts Valley, CA, 2007.

[3] J. Phiri, *EEE6612 Lecture,18 April 2017,* Lusaka: University of Zambia, 2017.

[4] J. Shabani, *EEE6635 Lecture, Network Security,* Lusaka: University of Zambia, 2017.

[5] J. Shabani, *Gathering Target Information: Reconnaissance,Footprinting & Social Engineering,* Lusaka, Lusaka: University of Zambia, 2017.

[6] J. M. Stewart, E. Tittel and M. Chapple, Certified Information System Security Professional, Canada: Wiley, 2008.

[7] J. R. Vacca, Computer and information Security Handbook, Amsterdam: MK, 2009.

[8] N. S. Abouzakhar, *Introduction to Intrusion Detection,* Hartfield: University of Hertfordshire, 2012.

[9] Y. X. K. G. H. D. J. Z. Jing Liu, "Botnet: Classification, Attacks, Detection, Tracing,," *EURASIP Jour nal on Wireless Communications and Networ king,* vol. 2009, no. 692654, pp. 1-11, 2009.

[10] F. J. a. D. M. E. Cooke, "The zombie roundup:Understanding, detecting, and disturbing botnets," in *Workshop on Steps to Reducing Unwanted Traffic on the Internet*, 2005.

[11] W. Stallings, Cryptography and Network Security, 4th Edition ed., T. Dunkelberger, Ed., Upper Saddle River: Prentice Hall, 2006.

[12] F.Pfeiffer, "Minerbot Target List," Minerbot Target List, 25 Julky 2017. [Online]. Available: http://www.ax10m.de/minerbot. [Accessed 25 July 2017].

[13] N. I. Hackworth, "Botnets as a vehicle for online crime," in *18th Annual Forum of Incident Response and Security Teams*, Baltimore, 2006.

[14] R. Ferguson, " The history of the botnet – Part 1," http://countermeasures.trendmicro.eu/the--history-of-the-botnet-part-i/, 25 July 2017. [Online]. Available: http://countermeasures.trendmicro.eu/the--history-of-the-botnet-part-i/. [Accessed 25 july 2017].

[15] D. D. a. S. Dietrich., "P2P as botnet command and control: a deeper insight," in *3rd International Conference on Malicious and Unwanted Software*, 2008.

[16] S. M. Kerner, August 2017. [Online]. Available: www.eWeek.com.

[17] A. shevat, August 2017. [Online]. Available: www.chatsbotslife.com/revenue-models-for-bots-and-chatbots-702ca78a1bo7.

[18] Q. L. A. B. A. S. Zhen Li, "Figthing botnets with Economic Uncertainity," *Security and Communications Networks October 2011,* vol. 10, no. 1002, pp. 1-11, October 2011.

[19] Q. L. A. S. Zhen Li, "Botnet Economics: Uncertainity Matters," in *Workshop on the Ecnomics of Information Security*, Hanover, New Hampshire, 2008.

[20] T. Holz, "Measurements and Mitigation of Peer-to-Peer-based," in *1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, 2008*, 2008.

[21] B. S. D. Larry Petersen, Computer Networks: A Systems Approach, San Francisco: Elsevier, 2007.