Hacking Botnets

Table of Content

BotNet Overview: The zombies are coming! The zombies are coming...No, they are already here, and they don't want to eat your brains, but your data! A zombie is when a hacker takes control of many computers without the user's knowledge forming what is called a "BotNet," aka Zombie Army. The term BotNet is a combination of words "robot" and "network" and a hacker uses that army of computers to attack other networks. BotNet's are everywhere on the web helping people and companies to collect data, to infecting networks with viruses, or just learning your every click as you surf-the-web. A BotNet can be invisible to the average user and to the IT professional alike because they can be designed to do good work, or take-down an entire network depending on the goals of the designer. Some of the best BotNets have been found on thousands and even millions of servers.

The Zeus BotNet was one of the most powerful financial malware viruses on the internet. Its primary function was to steal online credentials, especially banking related information. Zeus used stealth techniques to hide itself from anti-virus software which is one of the reasons it was found on some 3.6 million computers. Another one called the Simda BotNet infected more than 770.000 computers in over 100 countries. It had set up some fourteen command and control servers in the Netherlands, United State. Russia and Poland to name a few, and it took Interpol, the FBI, Kaspersky Lab, Trend Micro, Cyber Defense Institute and others working together to counteract the cybercriminals BotNet network.

How To Know If You're Infected

- Is your computer running slower than normal over the internet?
- Does your computer behave erratically? Does it crash?
- Do you receive unexplained error messages?
- Did the fan kick into overdrive when your computer is idle and the hard-drive or flash drive is running all the time?
- Do you notice unusual internet activity (like high network usage) and added connections?
- Does your browser crash unexpectedly, then restart?
- Did your computer take a long time to shut down or giving you errors as it shutting down?
- Take a look as to what is running on your system by looking at the **Task Manager** or better yet download **SysInternals** from Microsoft, its free! It's the task manager on steroids, and lets you do in-depth discovery of what is running, and where it's running on your system. The link is below!

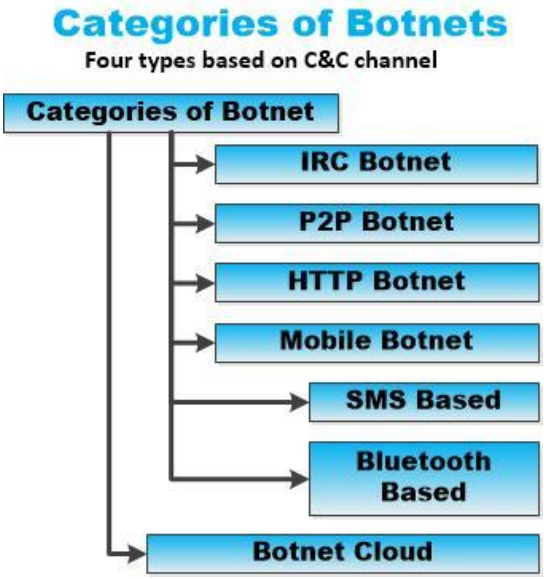
  https://docs.microsoft.com/en-us/sysinternals/downloads/sysinternals-suite

Key Commands For Testing: **Here is a short list of some of the key commands you can also run to see what is happening on your system..**

Here is a list of useful Netstat Commands, you can run on the command prompt	
netstat –a | more	Lists all the Listening ports of TCP and UDP
netstat -at	Lists all TCP port connections
netstat -au	Lists all UDP port connections
netstat –l	Display protocol statistics and current TCP/IP connections
netstat –s	Display statistics by protocol
nbtstat –n	Displays NetBios over TCP/IP
	NBTSTAT [[-a RemoteName] [-A IP address] [-c] [-n] [-r] [-R] [-RR] [-s] [-S] [interval]]
Other useful Windows commands	
ipconfig	View and configure network address setting
perfmon.exe	Monitor the performance of local and remote computers
resmon.exe	Monitor the performance of a local computer
shutdown –m \\computername	Replace the computer name with the name of the computer you wish to shutdown.
taskkill	Kills a program that running, ex: “taskkill/im chrome.exe /F”
taskkill /PID 2704 /F	Kill a PID running replace the 2704 with the PID you want to kill
pathping	Is like tracert, but better
	pathping [-g host-list] [-h maximum_hops] [-i address] [-n] [-p period] [-q num_queries] [-w timeout] [-4] [-6] target_name
ncpa.cpl	Display network connection like Bluetooth
wf.msc	Will show windows firewall setting

Infection lifecycle: A BotNet is made up of four phases

1. Initial information - find a vulnerability and exploit
2. Inject scripts to infect the victim
3. Establish connection to Command & Control Channel (C&C)
4. BotNet request commands from C&C to the begin attack



Firewalls: One of the best ways to stop a BotNet from even getting into your system is to have a firewall installed and configured. But there are many different types of firewalls and each has its strength and weaknesses when it comes to detecting and stopping an attack. First off have a firewall that understands both IPv4 and IPv6 protocols on your network stack, because if you don't a BotNet could just bypass your firewall without even logging an event. There are software type firewalls and hardware firewalls. Hardware firewalls are faster, but less flexible; software firewalls are more flexible and can be updated more quickly to address an ever changing threat, but can be hacked more easily.

Firewalls Types:	**Firewall Architecture Types:**
• Packet Filtering Firewall • Stateful Firewall • Application Proxy Firewall • Dynamic Packet Filtering Firewall • Kernel proxy Firewall	• Screened Host • Multi-Homed • Screened Subnet

Packet Filtering Firewall: One of the first generation firewalls and makes it decisions based on protocol header values. It's your basic rudimentary firewall looking at source destination IP addresses, Port numbers, Protocol types, and In-Bound/Out-Bound traffic direction. The firewall sits at the network interface of the device using the Transport Layers and Network layers of the OSI model working with the Access Control List (ACL). Think of the ACL as a bouncer who looks at the guest list of the party you want to attend; if you're not on the list, you're not getting in! Packet filtering is build into most firewalls these days, but that should not be the only protection you have since BotNet's can circumvent the ACL list and spoof the header file.

There are many disadvantages to Packet Filter Firewalls:

- It can't stop application-specific vulnerability
- Limited Logging
- Does not support advanced authentication schemes
- Most likely can't detect packet fragmentation attacks

Packet filtering is a stateless inspection firewall. It does not understand the contexts of the packets and does not fully comprehend the communications going on between the two systems.

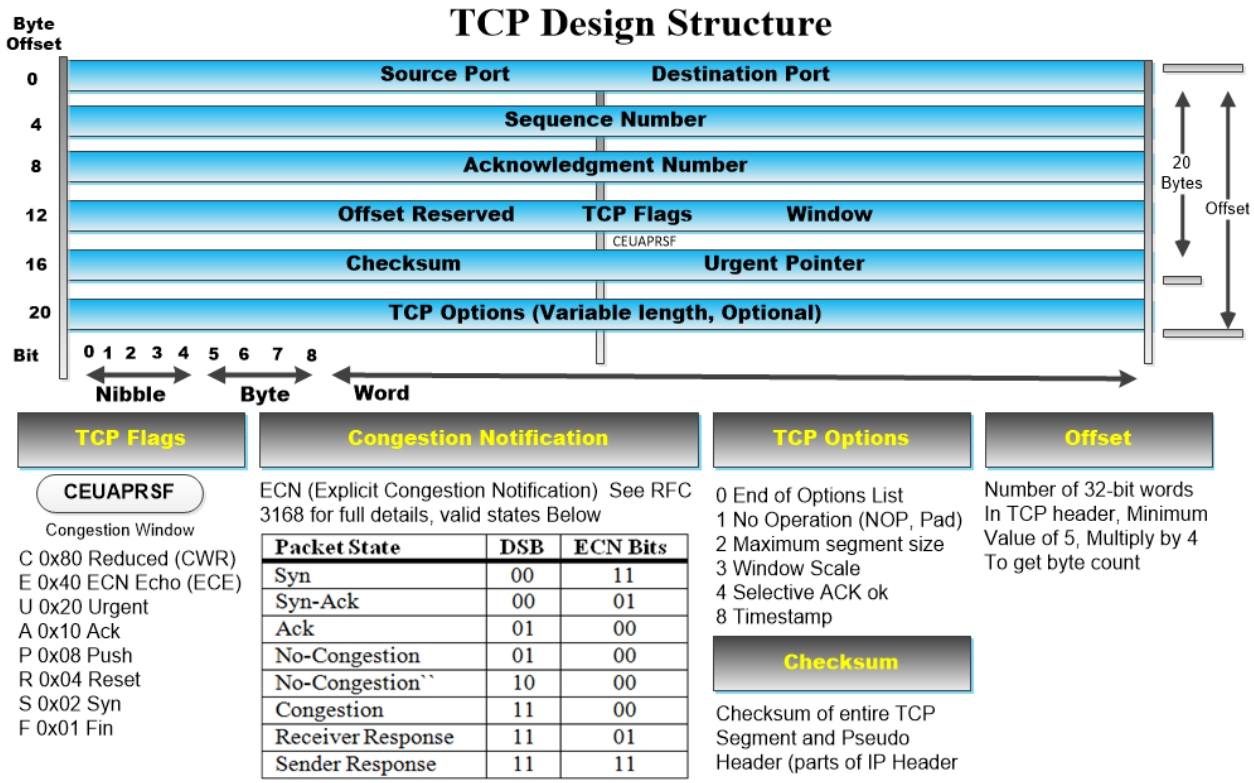
Some of the advantages:

- Packet filters are scalable and not application dependent.
- Packet filters are high performance firewalls since they do not carry out extensive processing on each package.
- Good to use as a first line of defense to weed out the most obvious malicious code, but should not be your only defense especially when trying to stop a BotNet attack.
- A BotNet's will most like get through this first line of defense depending what the BotNet is trying to accomplish.

Stateful Firewall: A Stateful firewalls looks at the data within the individual packet and tries to match incoming and outgoing packets as to which network communication session it belongs too? In this way a Stateful firewall has a much more complete picture of the network session and can reject packets that might be based on a network protocol attack.

A Stateful firewall keeps track of who is talking to whom, and maintains a state table of activities within the network. Many know of the (SYN, SYN/ACK, ACK) that computers use to setup a communication session. But there is a lot more going on in the background then SYN, SYN/ ACK...etc. It sets the SYN and ACK flags within the packet header set to 1. In addition, the systems agree upon sequence numbers, amount of data sent at a time, potential transmission errors to be identified by CRC values and others within the TCP header.

For example if all the TCP flag values within a packet are turned to 1 something malicious might be taking place. There is no legitimate reason for all the flags to be set to 1, so an attacker could be testing the Stateful firewall to see how it will react. But since Stateful firewalls track these types of connections it would reject it because the Stateful firewall examines all header payloads, and trailers. Below is a graphic of the TCP structure showing flag setting and options.



TCP Connection-Oriented Protocol Steps	
1) LISTEN	2) SYN-SENT
3) SYN-RECEIVED	4) ESTABLISHED
5) FIN-WAIT-1	6) FIN-WAIT-2
7) CLOSE-WAIT	8) CLOSING
9) LASTACK	10) TIME-WAIT
11) CLOSED	

In addition, remember UDP is a connectionless protocol and none of the steps above reflex its structure. This makes it hard for the Stateful firewall to keep track of what is happening. A Stateful firewall only keeps track of the source and destination addresses, UDP headers and some of the ACL rules within the UDP protocol. The firewall just time-outs after a period of time since its tracking ability on UDP protocols are limited to specifics periods of in-activity.

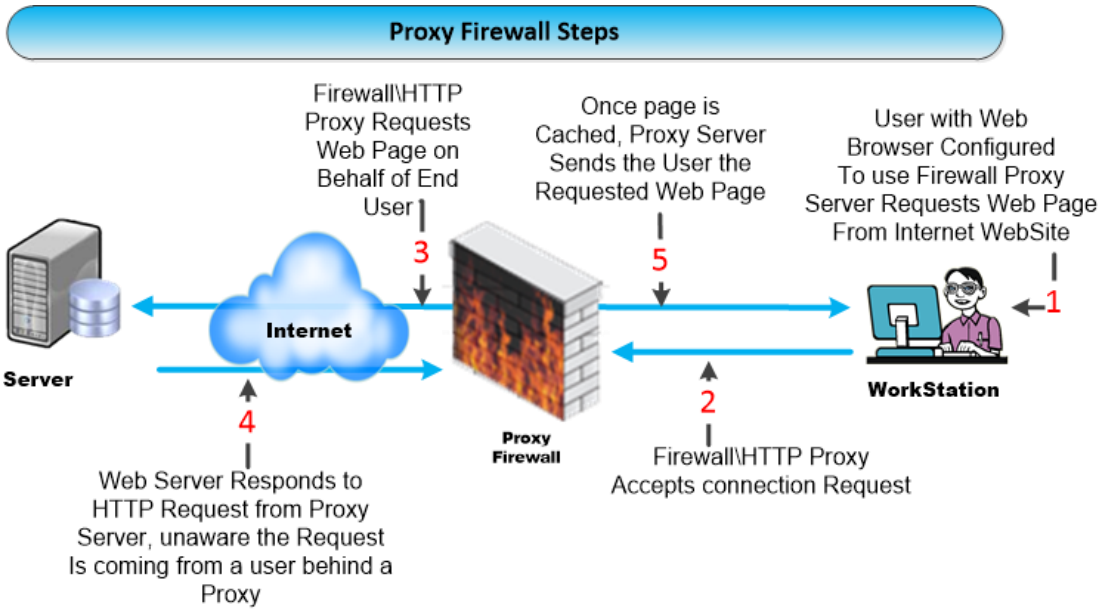
An interesting side note; since UDP is a connectionless protocol ICMP functions plays a part in helping to tell the two connected computers the speed at which to communicate, and can help in other ways by changing values in the header. However, if a network admin turns off a computers

capacity to utilize the ICMP protocol; one or both computers could end-up crashing if too much information overwhelms the connection.

Stateful Firewalls summary:

- Maintains a state table that tracks everything within the session
- Has a better performance matrix then an application proxy Firewalls with a higher degree of security
- Is Scalable
- Is Transparent to users
- Tracks UDP protocols using ICMP functions
- Updates and stores the context of data within the packets.

Application Proxy Firewall: A Proxy firewall works as a middleman and sits between the inside and outside of your network. What is import to note is that there is no direct connection between the two networks. In a packet-filtering device which just monitors traffic as it crosses the network. A proxy ends the communication session at one end, and restarts a new one on behalf of the sending system; meaning it starts a whole new session based on the external users interface. The external web server replies to the request that hits the external interface of the proxy firewall. The proxy firewall then deems if the communication is safe or not, then starts a new session from itself to the internal system acting as the middleman. In addition, a Proxy firewall does not depend solely on the access-control-list (ACL) rules because proxy firewalls can work on multiple levels of the OSI model. When it works at the lower layers of the OSI it is a Circuit-Level Proxy. If working at the application layer, it's called an Application-Level Proxy. A circuit-level proxy works at the session layer of the OSI watching traffic from a network view. It does not look at the contents of the packet and makes access decisions only based on the header and session information. So it does not make deep-packet inspections or understand the application layer protocols.



Application-Level proxies inspect packets through the application layer. It understands various services and protocols and can make decisions based on the content of the packet. For example; it can distinguish between "FTP GET," and "FTP PUT" commands looking at processes within the packet at a granular level.

To summarize; the circuit-level proxy handles a wider array of protocols and services, but does not look deep into the packets. The application-level proxy looks at each packet in greater detail. A circuit-level proxy makes access decision based on address, port, and protocol type header values. It does not know if the contents of the packet is safe or not, because it does not look that close. So when it comes to BotNet's, you might want to use an Application-level proxy given it looks within the contents of the packet other than just looking at the address as to where it came from and where it's going.

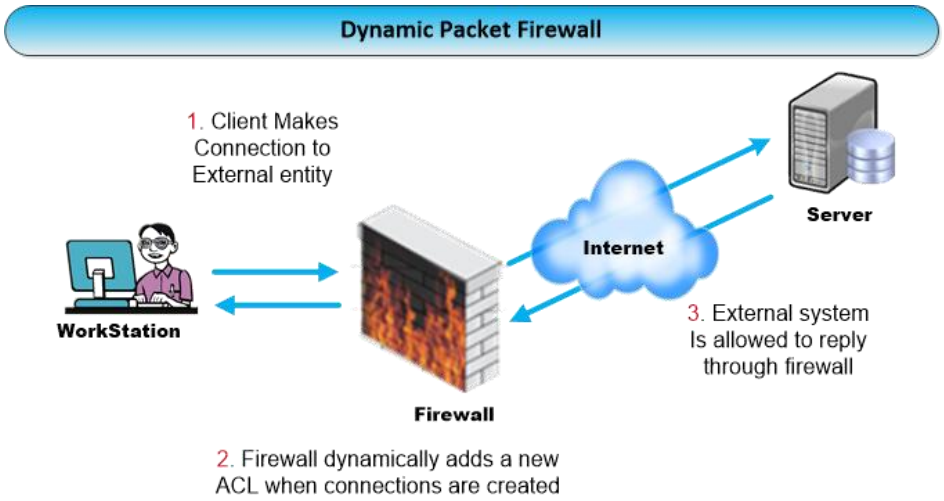
Application-Level Proxy Firewalls main advantages

- They have extensive logging capabilities.
- They can authenticate users directly which is unlike other firewalls that only use system authentication
- They don't just work on layer 3 of the OSI, so can address other issues like spoofing attacks and other types of attacks a BotNet might use.

Application-Level Proxy disadvantages

- Not great in high-bandwidth, real-time applications
- May have limited support for new network protocol applications
- Performance issues since it looks at each packet.

Dynamic Packet-Filtering Firewalls: Dynamic Packet-Filtering Firewalls also known as a "Stateful Inspection Firewalls," operate at the Network, and Transport layers (Layers 3 and 4) of the OSI model. They are known as third-generation firewalls. When your network needs to communicate with the outside world it chooses a source port so the outside network knows how to respond. Well-Known Ports – Ports 0 through 1023 are considered to be well known ports on most computer systems in the world. Registered Ports – Ports 1024 through 49151 can be registered with IANA by application developers. Dynamic or Private Ports – Ports 49152 through 65535 can be freely used by applications. A Dynamic Firewall evaluates the context of network traffic. It looks at source, destination addresses, application usages, origin, and relationship between current/previous packets. They operate more efficiently than application-Level firewalls. In addition, a Dynamic packet-filtering firewall can in real-time modify its filtering rules based on traffic content; something very useful when scanning for BotNet's and the many different types of bots.



Kernel Proxy Firewalls: This is known as a fifth-generation firewall. It can create dynamic, customizable virtual network stacks in order to scrutinize every layer of the packet as it arrives. This means it looks at the network header, transport header, session layer, and application layer. If it deems any one of the layers to be unsafe it discards the whole packet. Since all the processing is done within the kernel it is faster than application-level firewalls. This firewall like others is a proxy-based system so it acts as a middleman between the external and internal systems. It can also act as a NAT by changing the source address. This could be your best choice when it comes to a BotNet attack since many have to communicate with their command and control center from time to time.

Next-Generation Firewalls: The next-generation firewalls will have to do a lot more and combine not only the best parts of the previous firewalls. For example: build-in signature-based IPS engines that can learn traffic patterns, but also understand behavior patterns across networks and knows if the network is working according to the protocol rules or needs to change in real-time based on learned patterns. It can create and spot specific indicator patterns using advanced AI algorithms. This means knowing when the network is behaving normally and when it is not! Next-Generation firewalls will have the ability to connect to Active Directories, White-Lists, Black-List and other policy servers linked across cloud networks.

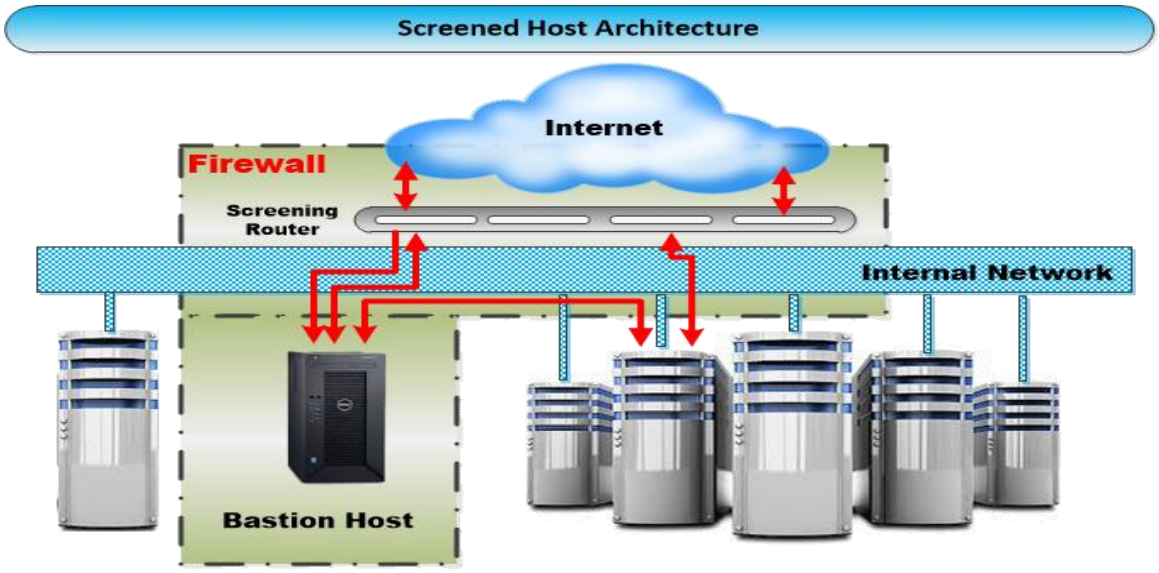
It will even be able to cook you breakfast or empty your bank account if you're ever mean to it….no…just kidding! Maybe someday when the Internet becomes self-aware which is something computer sciences are now theorizing could happen?

Firewall Architecture Types:

- Screened Host
- Multi/Dual-Homed
- Screened Subnet

Screened Host: The screened host firewall architecture uses a host called a bastion host in which all outside hosts must connect too. In this type of configuration a filtering router is configured and all internal connection from outside go directly to the bastion host. A single-homed bastion hosts can be configured as either a circuit-level or application-level gateway and

is by all others measures a proxy server. The primary security is provided by packet filtering. It's a way to prevent people from going around the proxy server to make a direct connection. Below is a graph of a simple version of a Screened Host architecture. The bastion host sits on the internal network and the packet filtering on the screening router. The bastion host is the only system on the internal network that a host on the Internet can open a connection too. The primary security is provided by packet filtering. It's a way to prevent people from going around the proxy server to make a direct connection. The bastion host sits on the internal network and the packet filtering on the screening router.



The Packet filtering configuration can do one of the following

- Allow internal hosts to open connection to hosts on the Internet for services
- Disallow all connection from internal hosts, forcing those host to use proxy services via the bastion host.

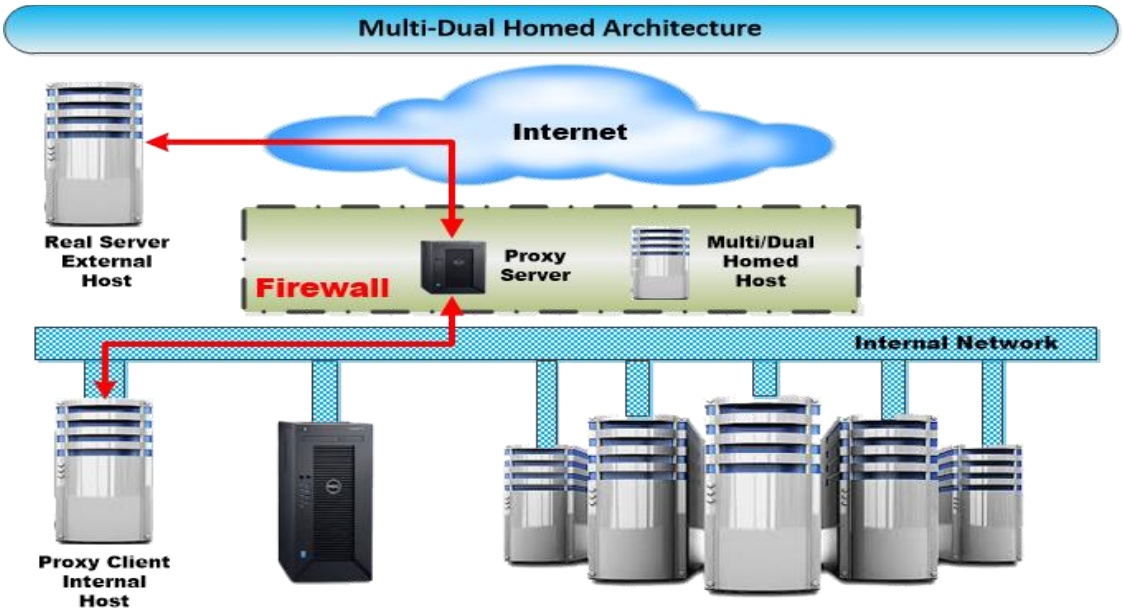
Disadvantages' of a Screened Host

- If an attacker manages to break into the bastion host no other security measures stands between the attacker and the internal host.
- The router is another single point of failure; if the router is compromised the attacker could gain access to the entire network.
- The additional flexibility of the screened host firewall is cause for two concerns.

- First, there are now two systems, the router and the bastion host that need to be configured carefully or could be a gateway for an attack.

Screened-Host firewall architecture offers only a single line of defense against possible attack. Because it allows a single host the bastion host to receive all incoming information. Thus making it a key target for hackers, and would not be a great choice for a defense-in-depth design methodology since it only provides for a single line of defense. The best use for a screened host architecture is when few connections are coming from the Internet (in particular if the screened host is a public web server) or if the network being protected has a relatively high level of host security.

Multi/Dual-Homed Architecture: A Multi/Dual-Homed firewall might not be as flexibility as a screened host, but some say it's technically impossible to pass traffic through the Multi/Dual-homed gateway unless there is a corresponding proxy service. A Multi/Dual-homed host has more than one network interface, and each interface is connected to separate network segments. One interface is connected to an internal or trusted network, and the other is connected to an untrusted or external network. In this type of setup the firewall is usually disabled so the IP packets from one network are not routed from to another network.



With two NICs, all traffic must physically go through the firewall to move between the internal and external networks. This type of configuration often makes use of NAT; mapping real, valid, external IP addresses to special ranges of non-routable internal IP addresses thus adding another layers of intrusion protection from external hackers. If you create a multi-homed bastion host, it translates between the true external IP addresses used by the organization to the public network naming authorities and the internally assigned, non-routable IP addresses. NAT dynamically assigns addresses to internal communications, tracking the connection with sessions to determine who is responding to whom.

Advantages of Multi/Dual-homed Host

- It has the ability to translate many different protocols at the data link layers, including Ethernet, Token Ring, Fiber Distributed Data Interface, and Asynchronous Transfer Method (ATM)

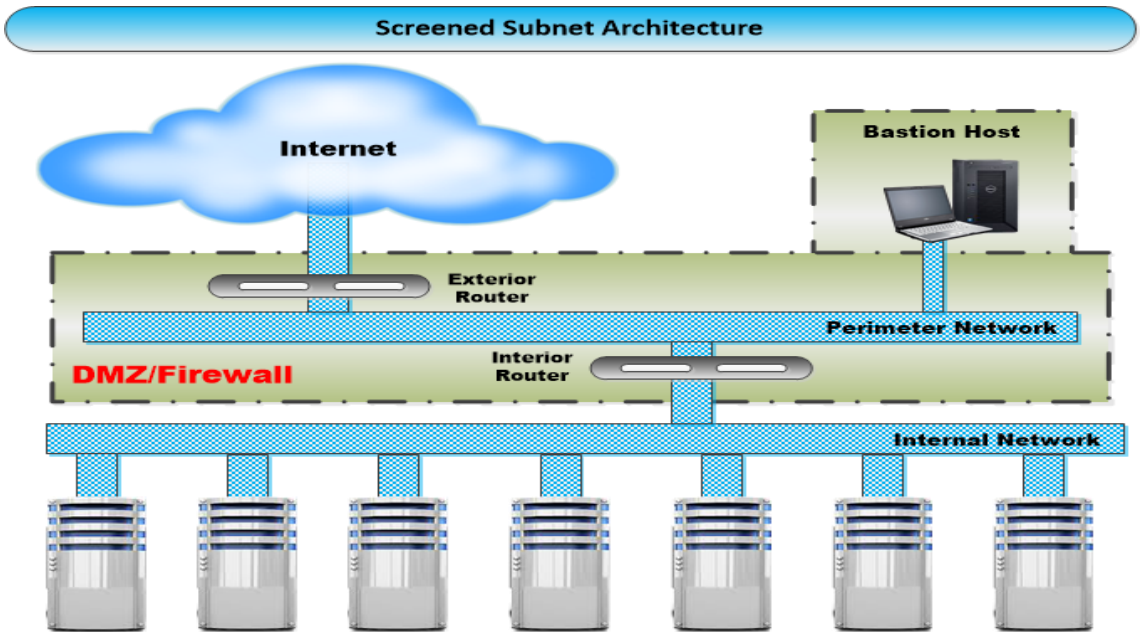
Disadvantages of Multi/Dual-Homed Host

- If the network is compromised it will likely disable the connection to the external host.
- Another problem is that if the host firewall is overloaded it could cause a reduction in filtered traffic.

Screened Subnet Architecture: A screened subnet is one of the most secure types of host architectures. It has a DMZ that protects information from the outside. It protects internal networks by limiting how external connections gain access. In addition because of its complexity it is more costly to operate, manage, and more difficult to configure. On the other hand the DMZ can have an area for an extranet where additional authorization and authentication protocol can exist to provide access for the general public. A subnet architecture can consist of one to two or more internal bastion hosts behind a filtering router. There are many different types of configurations of subnet architecture, but the most common consists of two filtering routers with a dual-homed bastion hosts between them as the example below shows.

- Connection traffic from the Internet is routed through an external router within the DMZ/firewall
- Which is the perimeter network also sitting within the DMZ/firewall

- This is where the Bastion Host which is also connected to the perimeter network within the DMZ/firewall
- Next it hits the Interior filtered router within the DMZ/firewall, before finally making it to the internal network.



A screened subnet protects the network segments by performs two functions:

- One it protects the DMZ systems and information.
- Two protects the internal network by limiting how external connections gain access
  In addition, it provides for the creation of an area known as an extranet. This provides for added access authentication and authorization controls.

Think of it as online retailers that let people shop online, but when it comes time to check out, added authentication/authorization is applied before the order can be placed

Additional Methods to Combat BotNets: You can start by turning-off services and processes you do not need, and disable autorun which will automatically install software updates that could be carrying a BotNet. In addition, compartmentalize your network setting up VLANS, and isolate or disable computers from automatically connecting to each other. Work on the premise of least privileges and don't let users be their own administrator. Moreover, install a host-based intrusion prevention system (IPS) to keep an eye on network layers, hardware and software so a BotNet can never gain root access to your systems. Create policies for enhanced monitoring of in-coming, but specially out-going communications also known as egress traffic filtering; because a BotNet typically needs to establish communications with a remote command & control (C&C) servers. Additionally, know what normal user activity is and what is abnormal. Likewise force traffic to go through proxies which will give you a secondary check point for monitoring and controlling web traffic. Monitor DNS Queries and how workstations are responding to DNS queries, and if responses become very low Time-To-Live (TTL) values which can be a leading indicator that you're infected.

What's more, keep an eye on the ports that a BotNet might be using to communicate with its command & control center. Here is a list of some of the main ports that might be of interest to watch.

Common Ports for TCP/UDP		
Port: 1863	MSN Chat	Microsoft
Port: 5050	Yahoo Messenger	Yahoo
Port: 5190	AIM/ICQ	
Port: 5222-5223	XMPP/Jabber	
Port: 6679/6697	IRC over SSL	
Port: 6891-6901	Windows Live	
Port: 7648-7649	CU-SeeMe	
Port: 8767	TeamSpeak	
Port: 9119	MXit	
Port: 25999	Xfire	
Port: 411-412	Direct Connect	
Port: 1214	Kazaa	
Port: 1337	WASTE	
Port: 4672	eMule	
Port: 6257	WinMx	
Port: 6346-6347	Gnutella	
Port: 6679/6697	Napster	
Port: 6881-6999	BitTorrent	
Port: 1755	MS Media Server	Microsoft
Port: 3784-3785	Ventrilo	
Port: 5001	Slingbox	
Port: 5060	SIP	
Port: 5004-5005	RTP	
Port: 6970	Quicktime	
Port: 8000	Internet Radio	
Port: 24800	Synergy	

Legend: Purple-Chat/ Brown-Peer to Peer/ Streaming

BotNet Command & Control:

BotNets use command and control (C&C) channels through which a botmaster tells the bot to carry out some type of activity on the systems it infected. There is a number of other ways to detect BotNets other then firewalls,

network policies, IDS systems, and server configurations as outlined above. Take a look at how the BotNet communicates with its C&C channel and how its styled e.g., IRC based, HTTP-base, or peer-to-peer (P2P) based for example are used.

Given that BotNets are organized networks of computers running bot code, they still have to run home to the command and control channel for instructions from time to time. So just what is this command and control channel (C&C) and what methods does it use to communicate?

There are four main types of C&C channels

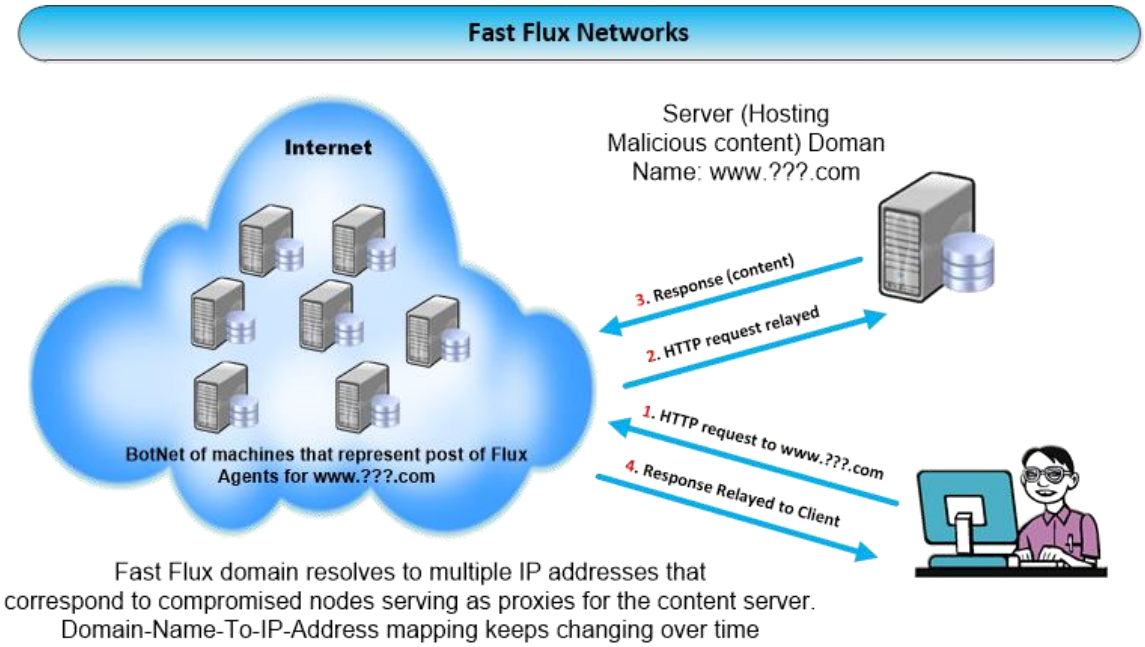
1. Internet Relay Chat (IRC)-Based, which is a push-based model of communication which the C&C pushes out new instructions to the bot over the network.
2. HTTP-based which the bot polls the C&C channel for instructions, a pull-based communication method.
3. Peer-to-Peer (P2P) based C&C channel of communications.
4. Hybrid of one or all of the communication channels above.

Below is a more in-depth chart from the World Academy of Science, Engineering and Technology Vol: 8, No9, 2014.

Comparison Chart of BotNet Categories					
Type Of BotNet	**C&C Protocol Channel**	**Structure**	**Strength**	**Weakness**	**Examples**
IRC	IRC	Centralized	Low latency communications, Flexible, Botmaster have Real Time control over the Bots	The entire BotNet can be collapsed by shutting down the IRC server	GTbot, SDbot, AgoBot, Spybot...etc.
P2P	WASTE, P2P, Self-defined	Decentralized/Distributed (P2P) Centralized	Free from single point of failure, more robust	High Latency Communication	Nugache, Storm..etc
HTTP	HTTP	Centralized	Bots hide their communication flows in the normal HTTP traffic	Botmasters do not have real time control over the Bots, the entire BotNet can be collapsed by shutting down the web server	Bobax, Clickbot..etc.
Mobile BotNet (SMS Based)	SMS	Tree Topology	Communications realized in tree topology. Difficult to detect the bot communication	SMS-based C&C requires a node list to be operated on infected phones.	iKee.B..etc
Mobile BotNet (Bluetooth)	Bluetooth	Changing Topology	Faster Communication, Data transfer is cost free	Difficult to construct the channel if the devices are out of range. Short data	ZitMo

				transfer due to consumption of more battery power	
Bot Cloud	Cloud Resources	Changing Topology	Fully utilize the resources without interruption, construction time is very less. But cloud is always online and ready to use.	Weakness of Cloud computer	

Fast Flux Networks: A BotNet can use a method called Fast Flux which is a Domain Name Server (DNS) technique used to hide the central C&C server. It uses an array of hacked servers to proxy messages between the central C&C server and the BotNet. The Botmaster continually changes proxy servers to which a domain name points to, and the bots find the C&C proxy by looking it up in the DNS. This programming technique allows the creator to not have to hard code any static IP address into the bot. It also makes it extremely difficult to find the C&C source controller of the bot.



Intrusion Detection Systems: There are three categories of intrusion detections systems (IDS) one might use to detect BotNets.

1. Host-Based
2. Network-Based
3. Hybrid of the two.

Host-Based systems typically use signature or behavior-based methods to track BotNet signatures within the network traffic. But it's limited on spotting BotNet infections because it relies on having some knowledge of its behavior which is not always possible. Still Host-Based systems are easy to deploy and can spot single bot infections most of the time. Network-based methods correlate information and behaviors across different hosts on the network. They don't need prior knowledge of bots signatures. Network-Based systems look at the characteristics of bots and how infected systems compare to uninfected systems and how they are reacting on the network. In a sense they rely on multiple hosts on the same network to become infected before an intrusion is detected. So many intrusion detection systems rely on anomaly detection to discover bot infections. They compare normal activity to abnormal behavior like CPU usage, modifications to file systems, and adverse traffic going-out/coming-in on the network. Since any new infection will cause changes on some part of a network's file structure and CPU usage.

Many BotNet today are starting to encrypt there C&C messages in order to obfuscate their structure and code. This can greatly complicate discovery and tracking down the infected systems. The IDS system must first build a whitelist of legitimate destination, and a base of normal operations. In this way it can spot any new persistent communications between the host and the C&C server and raise the alarm.

Mobile Systems: Botmasters now have an eye on your mobile device given all the weakness within wireless and Bluetooth platforms. This along with cloud networks which a lot of these devices depend on to communicate makes it a perfect environment for exploitation. Bluetooth along with HTTP, FTP, P2P, ICMP, DNS, IRC and other IEEE standards make it a perfect target for a Botmaster to create a C&C center. This would enable the Botmaster to create a massive zombie army using Bluetooth and the cloud.

Summary: In the end these methods above are but a few ways to protect you against the onslaught of botnets being created every day. Remember, all systems are run on software, and software by its own design is done by people, and no one is a perfect programmer! It is said that for every one thousand lines of code there are at least fifteen major errors, from security holes in the software to the operating system it's running on. Design flaws are everywhere and so are botnets. The Internet was not designed as a secure platform, it was designed to share information and in that respect it does it brilliantly.

References:

- https://www.esecurityplanet.com/trends/article.php/3920881/11-Ways-to-Combat-BotNet-the-Invisible-Threat.htm

- David Zhao, IssaTraore, BassamSayed, Wei Lu, SherifSaad, Ali Ghorbani and Dan Garant,"BotNet detection based on traffic behavior analysis and flow intervals,"*Elsevier Computers & Security*, vol. 39, pp. 2-16,November 2013.

- Maryam Feily, AlirezaShahrestani and SureswaranRamadass," A survey of BotNet and BotNet detection," in*2009Proc.IEEE third international conference on emerging security information, systemsand technologies*, pp.268-273.

- Abdullah J. Alzahrani and Ali A. Ghorbani, "SMS mobile BotNet detection using a multi-agent system: research in progress,"in*2014Proc. ACM ACySE1st International Workshop on Agents and Cyber Security*, no.2.

- Heloise Pieterse and Martin S. Olivier," Bluetooth Command and Control channel,"*Elsevier Computers & Security*, vol. 45, pp. 75-83, September 2014.

- Jerome Francois, Shaonan Wang, Walter Bronzi, Radu State and Thomas Engel, "BotCloud:Detecting BotNet using MapReduce," in*2011 Proc. IEEE International Workshop on Information Forensicsand Security*, pp. 1–6
- Peter Wurzinger and Leyla Bilge. Automatically Generating Models for BotNet Detection, European Symposium on Research in Computer Security, 2009.

- Liberios Vokorokos, Anton Balaz, Martin Chovanec. Intrusion Detection System

- Using Self Organizing Map, Acta Electrotechnica et Informatica, 2006.

- Frederic Giroire, Jaideep Chandrashekar, Nina Taft, Eve Schooler, Dina Papaginnaki. Exploiting Temporal Persistence to Detect Covert BotNet Channels, Recent Advances in Intrusion Detection, 2009.

- Anirudh Ramachandran, Yogesh Mundada, Mukarram Bin Tariq, Nick Feamster. Securing Enterprise Networks Using Traffic Tainting, Special Interest Group on Data Communication, 2008.

- Guofei Gu, Junjie Zhang, Wenke Lee. BotSniffer: Detecting BotNet Command and Control Channels in Network Traffic, Network and Distributed System Security,

- Guofei Gu, Roberto Perdisci, Junjie Zhang, Wenke Lee. BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent BotNet Detection, Proceedings of the 17th conference on Security symposium, 2008.

- Su Chang and Thomas Daniels. P2P BotNet Detection using Behavior Clustering & Statistical Tests

- Clam AntiVirus. http://www.clamav.net.

- Weka 3 Data Mining and Machine Learning Software. http://www.cs.waikato.ac.nz/ml/weka/.

- 11 Ways to Combat BotNet, the Invisible Threat; By Peyton Engel, Posted January 14, 2011

- John John, Alexander Moshchuk, Steven Gribble, Arvind Krishnamurthy. Studying Spamming BotNet Using Botlab, Network Systems Design and Implementation, 2009.

- Yuanyuan Zeng, Xin Hu, Kang Shin. Detection of BotNet Using Combined Hostand Network-Level Information, International Conference on Dependable Systems & Networks, 2008.

- Joe Stewart. Inside the Storm: Protocols and Encryption of the Storm BotNet, http://www.blackhat.com/presentations/bh-usa-08/Stewart/BH US 08 Stewart Protocols of the Storm.pdf, 2008.

- Andreas Pitsillidis, Kirill Levchenko, Chritian Kreibich, Chris Kanich, Goeffrey Voelker, Vern Paxson, Nicholas Weaver, Stefan Savage. BotNet Judo: Fighting Spam

with Itself, Network and Distributed System Security, 2009.

- Phillip Porras, Hassen Saidi, Vinod Yegneswaran. A Multi-perspective Analysis of the Storm (Peacomm) Worm http://www.cyber-ta.org/pubs/StormWorm/report, 2007.

- Thorsten Holz, Moritz Steiner, Frederic Dahl, Ernst Biersack, Felix Freiling. Measurements and Mitigation of Peer-to-Peer-based BotNet: A Case Study on Storm Worm, USENIX Workshop on Large-Scale Exploits and Emergent Threats, 2008.