RESEARCH ARTICLE

OPEN ACCESS

# **Generic Architecture for Detecting Botnet**

Anushah Khan<sup>[1]</sup>, Anchit Bijalwan<sup>[2]</sup> Department Of Computer Science and Engineering Uttaranchal University Dehradun – India

## ABSTRACT

Presently, Internet is used all over the world for different purposes and people take advantage of it in almost all possible ways. But at the same time there are large number of attackers and hackers which can harm the user and his/her information that is transmitting through the internet. One of the major internet security threats is Botnet. In order to handle these types of internet security threats, different techniques and tools have been developed. Botnet is the association of large number of compromised computer systems called Bots that work collective in order to perform the malicious purpose. The malicious activities supported by Botnet are Distributed Denial Of Service (DDoS) attacks, Spamming of emails, Phishing and creating the illegal computer systems to cause exchange of harmful material. The Botnet differentiates itself from other malicious software by having the ability to work under its originator called Botmaster or BotHeader that uses the Command and Control(C&C) Server to forward its commands to the Bots. In this paper, we have given the general idea about how Botnet performs the malicious activities and various techniques that are used for the revelation of the Botnet. Later, we have used the tool called Wireshark for detecting the bot and have proposed a generic architecture for detecting the Botnet that helps in securing the network traffic, exchanging over the internet.

*Keywords:-* Botnet, Bot-master, C&C server, DDoS attacks, Honeypots, IRC-based botnet.

#### I. **INTRODUCTION**

unprotected computers ("Botmaster"). It is a collection of software robots, or bots, Cycle consists of five phases .Figure 1 below shows the life which run automatically. They run on groups of zombie cycle of the botnet. computers controlled remotely by the attacker. Bots are used to perform a wide variety of malicious and harmful actions against systems and services like distributed denial of service In the first phase, the Botmaster, which is the attcaker expoits (DDoS) attack, spam campaigns, and phishing activity. The the vulnerable system by sending malicious progarms to it size of the Botnet may differ from tens and hundreds to few like Trojans and therefore, this phase is known as preliminary thousands. Most of the times, the host machine does not know infection phase. This gives back door entry to the BotHearer. that it is compromised [[1],[2],[3]]. In fact, the system which In the second phase, the infected system downloads and we are using can also be a part of Botnet. The attacker first installs the bot binary into itself. Once the bot program is exploits the unprotected system by usually Trojans and once installed in the exploited system, it starts behaving like a Bot the system gets infected, it comes under the control of the and therefore is known as Secondary injection phase. In the Botmaster. The Command and Control(C&C) Server is used third phase, the bot send query to the DNS server in order to for sending command to the bots. The C&C server connects get the address of the C&C Server. The moment the bot gets the Botmaster with the Bots. Botnet may have none, one or the address, it joins to the C&C Server and authenticates itself many C&C Servers. The C&C Server receives the commands to it. The C&C Connection is made by the bot program that from the Botmaster, forwards them to the botnet and then was installed in the victim system which has now become a sends the reports back to the Botmaster. Botnets are used to bot. Once the C&C connection is established, the newly made perform DDOS attacks against the number of targets bot becomes the part of the botmaster's botnet army and is including government and even other botnets. It is possible to now ready to act according to the commands that it receives re-program or update the botnet node software after it has from the C&C Server[[6], [7]] infected a system Polymorphism and Rootkitting are two of the most common techniques in use. In polymorphism, the

malware code changes with every new infection in order to avoid being detected by the anti-virus. In rootkitting, the Botnets are emerging threat with hundreds of millions of installed malware called "rootkit" is activated each time a computers infected. Botnets have become a severe global system boots up. The rootkits are not easy to detect because Internet threat. A "Botnet" consists of a network of they are activated before the Operating System of any system controlled by an attacker has completely booted up [[4], [5], [6]]. The Botnet Life

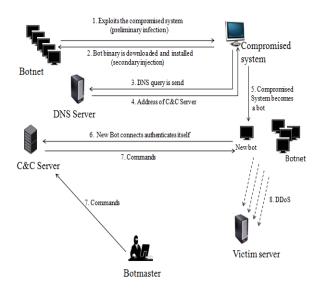


Figure1: Botnet Life Cycle

. In the fourth phase, bot master relays the commands to the bot through various mechanisms such as HTTP or IRC server to direct the bot in performing the attack. The Last Phase is related to the up-gradation and Continuance of the malware so that the botmaster is kept up to date with the botnet army for future co-ordinated attacks.

Section 1 defines the introduction of Botnet , Section 2 demonstrates the related work on the Botnet , Section 3 describes the Botnet Revelation and various Revelation techniques for detecting the Botnets , Section 4 presents the proposed idea , and Section 5 discusses the various research challenges and conclusion.

## II. RELATED WORK

cycle of botnets and architectural designs. It also classifies botnet detection techniques into two categories, host-based and network-based techniques. However [[8], [9]] do not focus on real world botnets.

Botnet detection methods are classified in two categories namely honeynets and passive traffic [10]. Several data sources for botnet detection are enumerated [11]. The evadability of detection methods are also studied [12]. The evasion cost is proposed as a measure of how good each method is. This cost represents the complexity of the evasion technique and the utility lost by the botnet when the evasion technique is successful. The detection techniques are classified into four classes namely signature-based, anomaly-based, Domain Name System (DNS)-based and mining-based techniques [1]. This is the first survey to use capabilities in a comparison table of detection techniques - ability to detect unknown bots, capability of botnet detection regardless of botnet protocol, encrypted command-and-control (C&C) channels and structure, real-time detection and accuracy. Several botnet detection and tracing methods are analyzed [13]. They are separated into honeypot-based, IRC-based and DNS-based methods. The IRC-based category is separated into traffic analysis-based and anomaly activities-based methods. A topology of network-based and anomaly-based detection systems is presented [14]. Another research work has implemented an algorithm for detecting a botnet. The authors mention features of botnet DNS traffic that is distinguishable from legitimate DNS traffic. They defined the key feature of DNS traffic called group activity, as they studied and grasped botnets behavior. They developed an algorithm that differentiates a botnet DNS query by using group activity feature.

## III. BOTNET REVELATION

In order to detect attacks from botnet, many researchers concentrated on analyzing the characteristic of packet [[53], [54], [55]]. Via different methodology of analyzing attacks, attacks from botnet are detected and some standards are computed to evaluate the performance of the methodology

Large number of work has been done on the detection of the [11]. Al-Ahmad et al. [29] used a Sniffer program that botnets. The detection techniques mostly used by the performed monitoring function. All the message that are researchers include Signature based, Anomaly based, Network exchanged between the bots and the botmaster ,the IP header based, Host based and Data mining based techniques. In the of TCP were captured and then discrimination was made earlier days, Signature based techniques were used for between the legal and illegal activities by using statistical detecting the botnets but it quickly lost importance when it chart. Garcia et al. [59] used the EM Clustering algorithm for could not find the unknown bots. A number of passive the detection of synchronization in bots and for the detection techniques like honeypots, analysis of flow records, and of the behaviour of the botnets . The EM algorithm is used for analysis of spam records, packet inspection, and analysis of the clustering of the time slices that have been divided while application log files, DNS-based approaches, and evaluation seeking to the detection of synchronization Jianbo et al. [65] of anti-virus software feedback are examined. Active detection proposed an algorithm based on the analysis of flow. After the techniques like infiltration, detecting fast-flux networks, DNS preprocessing of flow grasped from layer 3 switches, it gets cache snooping, sinkholing, IRC-based botnets detection and three vectors, such as source IP, destination IP and package P2P botnets detection are examined. Various botnet mitigation size, then defines reasonable sliding window of time, does schemes are illustrated too. The survey [8] offers botnets dynamic analysis based on the algorithm of connection rate. history, components of a botnet, characteristics of a bot, life Steinberger et al. [61] used different techniques for the

### International Journal of Computer Science Trends and Technology (IJCST) – Volume 3 Issue 5, Sep-Oct 2015

detection of anomaly and for mitigating the botnets at the generate botnets in the network and generate an early report internet scale. Xiang et al. [60] provided a new mitigation for understanding the consequences of botnets. Nepenthe [18] technique that promoted the development of more efficientis the example of low interaction honeypot that simulate some countermeasures against advanced botnets. Zhao et al. [9]vulnerability and provides some features for the collection of presented a system for the detection of botnet activity in both malware binaries [19]. The drawback of this technique is that the command and control and attack phase. The botnetthe limited scale of exploited activities can be tracked. It can detection techniques can be categorized as follows: Honeypotonly give report for infection machines that are anticipated and and Honeynet, IRC-based detection , and others like IDSput in the network as trap system. It can't give a report for (Intrusion Detection System), Firewall etc. Figure 2 shows thethose computers that are infected with bot in the network [19]. pictorial representation of the botnet detection techniques. It can't capture the bots that use the method of propagation

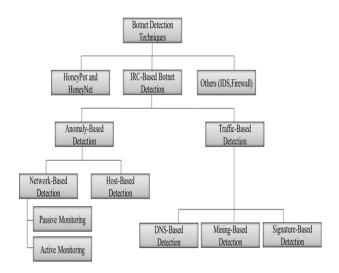


Figure 2: Botnet Detection Techniques

### 3.1 Honeypot and Honeynet

The first and the most general approach for detecting and categorized into: Detection based on traffic Analysis and tracing the botnets is the use of honeypots, where a subset Detection based on Anomaly Activities. pretends to be compromised by a Trojan, but actually

observing the behavior of attackers, enables the controlling 3.2.1 Detection based on Traffic/Flow Analysis

hosts to be identified[15]. Bethencourt et al. have successfully

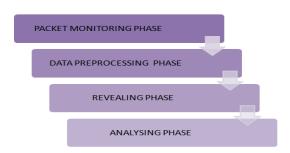
identified honeypots by using intelligent probing according to The main objective is to extract feature information on the public report statistics. Honeypot and active responders are packets from the traffic and match pattern registered in the used to collect bot binaries. Then, pretend to join the botnet as knowledge base of existing bots. Although it is easy to carry a compromised machine by running bots on the honeypots and on by simply comparing every byte in the packet, but it has permitting them to access the IRC Server. In [16], Zou and several demerits [21]. It should always update the knowledge bases are honeypot detctio based on independent software and hardware patched, the new bots may launch attacks. In [22], Sroufe et .The useful information gathered by the honeypot is: Signature al, proposed a different method for detecting the botnets. Their of bots for content-based detection, information of botnet method can effectively and automatically identify the spam or C&C mechanism/Servers, unknown security holes that enable bots. The main idea is to extract the shape of email by applying the bots to penetrate the network, tools and techniques that are the Gaussian Kernel density estimator [22]. In [[23], [24], used by the attack and finally the motivation of the attacker. In [25]], flow/traffic analysis is used to detect the attacks from [17], the author has used honeypot to track and

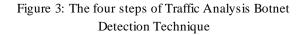
It can't capture the bots that use the method of propagation other than scanning e.g., spam. So we can come to the conclusion that generally in this technique we have to wait until one bot in the network infect our system and then we can track or analyze the machine.

### 3.2 IRC-Based Detection

One of the simplest ways to detect this kind of botnets is to sniff traffic on common IRC ports, and then check if the payloads march the strings in the knowledge database [15]. Racine found IRC-based bots were oftidle and only responded upon receiving a specific instruction [20] .Therefore; the connections with such features can be marked as potential enemies. In [3], Rajab et al. introduced a modified IRC client called IRC tracker that was able to connect the IRC Server and reply the queries automatically. The IRC tracker could instantiate a new IRC session to the IRC Server, if the template and the relevant fingerprint are given.

In [21], the real traffic on IRC communication ports ranging from 6666 to 6669 was observed by authors. It was found that some IRC client repeated sending the login information while the denied their connections.Depending on the results of the experiment, they claimed that the bots would repeat these actions at certain intervals after denying by the IRC Server, and those time intervals are different. Nevertheless, they did not consider a real IRC-based botnet attack into their experiment. IRC-based Detection technique can be botnet. It can be divided into the four steps: Packet monitoring analyzing different domain attributes such as the lifetime of phase, Data preprocessing phase, revealing phase, and the domain, TTL of the query, page ranking of domains, and Analysis phase, the diagram of which is shown below inhow frequently a query is applied. Figure 3.





3.2.1.2 Data Mining Based Detection – This technique uses the data clustering, machine learning and classification for the revelation of botnets. Identifying botnet C&C traffic is one of the effective methods for detecting the botnets. Botnet C&C traffic is different to detect. Since normal protocols are used by the botnets for C&C communication; the C&C traffic is not high volume and does not cause high network latency. Thus anomaly-based methods are not useful to identify botnet C&C Server traffic. The common approach which applies data mining technique for the detection of botnet C&C traffic is Botminner [28]. It is an improvement and advancement of similar malicious traffic Botsniffer [29]. The and communication traffic are gathered by Botminner. After that, it performs the cross cluster correlation in order to identify the hosts that share both similar communication patterns and similar malicious activity patterns .It has the capability to detect the real world botnets including IRC-based, HTTP

In Packet Monitoring Phase, packet sniffer is used to monitorbased, and P2P botnet with a very low false positive rate [28]. the packets. In Data Preprocessing Phase, the data is

recalculated in the form so that they can be used to detect the 3.2.2 Detection Based on Anomaly Activities attacks. In revealing Phase, the normal data and abnormal data

technique can be categorized into the following:

attacks. In revealing mase, the normal data and are distinguished. In the Analysis Phase, the performance of the data attacks and the performance of the data attacks and the second data attacks and the second data attacks and the second data attacks attacks and the second data attacks studying the normal behavior and statistics of the system. The characteristics studied are high volume of data, high network

latency, traffic on unusual ports, etc. Therefore it can be 3.2.1.1 Signature Based Detection - This technique maintains concluded that this technique can also the unknown bots. This a database of known bots or attacks and compares the characteristics of netwok traffic with the known bots present comprise of two phases- Training and Detection phase. In the method is very efficient in detecting unknown bots and in the database. This technique is considered as an efficient training phase, the normal behavior system (in the absence of technique for detecting known bots. The bots are detected an attack) is observed and a profile is created, using machine quickly with almost zero false positive rates and needs less learning techniques. In the detection phase, the current system resources. The major drawback of this technique is that behavior of the system is compared to the created profile. it can't be used for detecting the unknown bots. For example However, it may use a lot of system resources as it has to Snort, which is an Intrusion Detection System, monitors constantly update the user and system profiles and it also network traffic to find signature of existing bots. generates a high false positive alarm [30]. The encrypted

botnet communication can also be detected by this approach. 3.2.1.2 DNS Based Detection - This detection technique is The Anomaly Based Detection Technique can be categorized based on the particular DNS information that is shared by the into Host based detection technique and Network based botnet and C&C. These are similar to anomaly detection detection technique. The Host based technique is used to techniques. Bots typically initiate connection with C&Canalyze and monitor the internals of the computer system Server to get commands. For accessing the C&C Server, bots instead of the network traffic on its external interfaces [30]. perform DNS queries in order to locate the particular C&CThe Network based technique is used to detect the botnets by Server which is hosted by the DDNS provider. Therefore it is monitoring the network traffics and can be categorized into possible to Detect botnet DNS traffic by DNS monitoring and Active monitoring and Passive monitoring. Passive monitoring detect DNS traffic anomalies[[26], [27]].During this stage, ais based on the ability to inject test packets into the network, detection mechanism is provided to analyze DNS traffic, servers or application for measuring the reactions of network. detect possible communication instabilities and detect DNSThus it can produce extra traffics. The Active monitoring uses anomalies (Choi, Lee et al. 2007; Villamarín-Salomón and some devices to inspect the traffics as they pass by. It does not Brustoloni 2008). Normally bots communicate within a single increase the traffics on the network for inspection. This administrative domain and it is easy to measure the strategy usually requires a long time to inspect multiple stages relationship between the bots and the C&C mechanism by or rounds of Botnet communication and activities to detect

Botnets. Majority of Botnet detections that currently exist are such based on passive network monitoring.

as Java, .NET languages, and scripting languages generally use a wrapper; no such wrappers are provided by libpcap or WinPcap itself. C++ programs may link directly to the C API or use an object-oriented wrapper

#### IV. **PROPOSED WORK**

Presently the network traffic compromises of various types of <sup>1</sup>. Data can be captured "from the wire" from a live network data. For example web contents, e-mails, files, real-time connection or read from a file of already-captured packets.

audio/video data stream and many more. Depending upon the 2. Live data can be read from a number of types of network, type of transmission needed either UDP or TCP is used as a including Ethernet, IEEE 802.11, PPP, and loopback. transport layer protocol .For instance, for the transmission of

web content, e-mails and files, TCP is used as a transport layer3. Network data can be browsed via a GUI, or via the terminal protocol as it is more reliable protocol. But for the transfer of (command line) version of the utility, TShark. time sensitive application like real time audio/video streams,

UDP is used. The applications that used TCP protocol4. Captured files can be programmatically edited or converted maintain a full duplex communication between the sender and via command-line switches to the "editcap" program. the receiver and there is also the sequenced flow control

between the two. To make a TCP connection between the5. Data display can be refined using a display filter.

sender and the receiver, the sender first sends the SYN packet

to the receiver to initiate the session. After the initiation of 6. Plug-ins can be created for dissecting new protocols. connection, an [SYN, ACK] packet is sent by the sender

indicating that a connection is maintained and now the sender7. VoIP calls in the captured traffic can be detected. If can receive the packets without overwhelming and invading encoded in a compatible encoding, the media flow can even be any of the internal buffer. At the end, the ACK packet is sent.played.

This process is known as TCP 3 way Handshaking. Due to this

ACK, the TCP protocol is more reliable than UDP protocol;<sup>8</sup>. Raw USB traffic can be captured.

still most of the P2P applications use UDP protocol for  $\frac{1}{12}$ . Wireless connections can also be filtered as long as they communication purposes. Due to the use of various kinds of protocols for capturing the data from different applications,

there has been the diverge inconsistency found in the volume 10. Various settings, timers, and filters can be set that ensure of traffic and in the time measured. Also some of them are only triggered traffic appear. unidirectional in nature.

Wireshark's native network trace file format is the libpcap 4.1 Detection of Bot from the network traffic by using the format supported by libpcap and WinPcap, so it can exchange Wireshark captured network traces with other applications that use the

In this section we have captured the packets of the malware same format, including tcpdump and CA NetMaster. It can transmitting over the network and have analyzed the bot read captures from other network analyzers, such infected host by using a tool called Wireshark. Wireshark is as snoop, Network General's Sniffer, and Microsoft Network Monitor. The user typically sees packets highlighted in green, a free and open-source packet analyzer. It is used and blue, and black. Wireshark uses colors to help the user identify for network troubleshooting, analysis, software the types of traffic at a glance. By default, green is TCP communications protocol development, education. and Originally named Ethereal, the project was black identifies TCP packets with problems - for example, Wireshark in May 2006 due to trademark issues. Wireshark is black identifies TCP packets with problems — for example, very similar to tcpdump, but has a graphical front-end, plus they could have been delivered out-of-order. Users can change some integrated sorting and filtering options. Wireshark is software that "understands" the structure (encapsulation) of

different networking protocols. It can parse and display the We have created the Virtual Box in our system and have used fields, along with their meanings as specified by different Oracle. Then Ubantu operating system is being installed on it. networking protocols. Wireshark uses pcap to capture packets, Thus we have created a virtual environment so as to keep the so it can only capture packets on the types of networks that system protected. The Wireshark is also installed in on of computer network Ubantu. We execute the malware in this virtual environment. supports. In the field pcap administration, pcap (packet capture) consists <sup>of</sup>The packets that were captured by Wireshark are analyzed in an application programming interface (API) for capturing this section. traffic. Unix-like systems implement pcap network in

the libpcap library; Windows uses a port of libpcap known as In figure 17 there are number of devices that are being WinPcap. The pcap API is written in C, so other languages scanned by 10.129.211.13. We see the number of handshake packets going out to all these target addresses or systems. These are all TCP scans taking place. We also see the port going out to: which is NetBIOS port (139).

Contraction of the second s	Destinution 10 129, 102, 24	Protocol	Infe
		TCP	And the second
129,211,13			isoipsigport-2 > microsoft-ds [SYN] Seg=0 win=
the second se	19.129.102.25	TCP	ratio-adp > microsoft-ds [SYN] Seq=0 Win=64240
.129.211.13	0.129.102.26	TCP	kpop > microsoft-ds [SVN] Seq=0 Win=64240 [TCM
.129.211.13	0.129.102.27	TCP	webadmstart > microsoft-ds [SYN] Seq=0 Win=64.
129.211.13	10.129.102.28		Imsocialserver > microsoft-ds [SYN] Seq=0 Win-
.129.211.13	0.129.102.29		icp > microsoft-ds [SYN] Seq=0 Win=64240 [TCP
.129.211.13	0.129.102.30		<pre>ltp-deepspace &gt; microsoft-ds [SYN] Seq=0 Winst</pre>
129.211.13	D. 129.102.3Y		mini-sql > microsoft-ds [SYN] Seq=0 Win=64240
.129.211.13	1.25.102.0		ardus-trns > netbios-ssn [SYN] Seq=0 Win=64240
.129.211.13	19.25.102.1		ardus-cntl > netbios-ssn [SYN] Seq=0 Win=64240
129.211.13	10.25.102.		ardus-mtrns > netbios-ssn [SYN] Seq=0 Win=6424
.129.211.13	10.25.102.		sacred > netbios-ssn [SYN] Seq=0 Win=64240 [To
129.102.3	6.129.211.13		Destination unreachable (Port unreachable)
.129.211.13	19.25.102.4		bnetgame > netbios-ssn [SYN] Seq=0 Win=64240
.129.211.13			bnetfile > netbios-ssn [SYN] Seq=0 Win=64240
129.211.13	a second s		rmpp > netbios-ssn [SYM Seg=0 Win=64240 [TCP
129.211.13			availant-mgr > netbios-ssn [SVN] Seq=0 Win=64.
.129.211.13			murray > netbios-ssn [SYN] Seq=0 Win=64240 [To
the second se			hpvmmcontrol > netbios-ssn [SYN] Seq=0 Win=64
The second s			hpvmmagent > netbios-ssn [SYN] Seq=0 Win=64240
a particular a second a second and a second			hpvmmdata > netbios-ssn [SYN] Seq=0 Win=64240
		12	kwdb-comm > netbios-ssn [SVN] Seq=0 win=64240
		Mart.	saphostctr1 > netbios-ssn [SYN] Seq=0 Win=6424
329 211 13 1	10.25.102.14	TCP	sanhostetels > nethios-sen [SVN] Secul Winu64
	129,211,13 129,212,212,212,212,212,212,212,212,212,	$\begin{array}{c} 1.29, 211, 13 \\ 129, 211, 13 \\ 129, 211, 13 \\ 10, 129, 102, 28 \\ 129, 211, 13 \\ 10, 129, 102, 29 \\ 129, 211, 13 \\ 10, 129, 102, 30 \\ 129, 211, 13 \\ 10, 129, 102, 30 \\ 129, 211, 13 \\ 11, 25, 102, 0 \\ 129, 211, 13 \\ 11, 25, 102, 0 \\ 129, 211, 13 \\ 11, 25, 102, 1 \\ 129, 211, 13 \\ 11, 25, 102, 1 \\ 129, 211, 13 \\ 11, 25, 102, 1 \\ 129, 211, 13 \\ 11, 25, 102, 1 \\ 129, 211, 13 \\ 11, 25, 102, 1 \\ 129, 211, 13 \\ 10, 25, 102, 1 \\ 129, 211, 14 \\ 10, 25, 102, 13 \\ 10, 25, 102, 1 \\ 10, 25,$	129,211.13       10.129.102.28       TCP         129,211.13       10.129.102.29       TCP         129,211.13       10.129.102.30       TCP         129,211.13       10.129.102.30       TCP         129,211.13       10.129.102.30       TCP         129,211.13       10.25.102.0       TCP         129,211.13       10.25.102.1       TCP         129,211.13       11.25.102.1       TCP         129,211.13       10.25.102.1       TCP         129,211.13       10.25.102.4       TCP         129,211.13       10.25.102.4       TCP         129,211.13       10.25.102.4       TCP         129,211.13       10.25.102.4       TCP         129,211.13       10.25.102.5       TCP         129,211.13       10.25.102.4       TCP         129,211.13       10.25.102.7       TCP         129,211.13       10.25.102.7       TCP         129,211.13       10.25.102.7       TCP         129,211.13       10.25.102.7       TCP         129,211.13       10.25.102.10       TCP         129,211.13       10.25.102.11       TCP         129,211.13       10.25.102.10       TCP

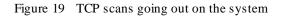
Figure 17 Handshake packets going out to the target systems.

In figure 18 we also see the ICMP destination unreachable responses grouped together. These are all of the different systems responding to the scanning device. When we do a TCP scan on a target system, we send a SYN packet to the target system. We expect to get either a [SYN, ACK] packet or a Resend, but not expect to get an ICMP destination unreachable (port unreachable) message. That may be indication that host is firewall that is why it did not respond as we expected.

e)		1	Typession_ 0	Deag Apply
. Time	Source	Destination	Pystocol	lefe
1 0 000487	10 129 102 0	6 10 129 211 111	TCHP	Destination unreachable (Port unreachable)
2 0.000730	10.129.102.1	7 10.129.211 11	ICMP	Destination unreachable (Port unreachable)
3 0.0004.66	100000000000000000000000000000000000000	8 million 1799 210 million	ICMP.	Destination unreachable (Port unreachable)
4 0.000730	10,129,102,1	9 10.129.211 13	TCHP	Destination unreachable (Port unreachable)
5 0.000486	10.129.102.2	10.129.211.13	ICMP	Destination unreachable (Port unreachable)
6 0.000/30	10.129.102.2	1 10,129,211 13	ICMP	Destination unreachable (Port unreachable)
7 0.000728	10 129 102 2	2 10 129 211 11	ICMP	Destination unreachable (Port unreachable)
8 0 000487	10 129 102 2	3 40 129 211 13	ICMP	Destination unreachable (Port unreachable)
9 0 000/ 30	10.129.102.1	4 10.129.211 11	ICMP	Destination unreachable (Port unreachable)
0 0 000486	10.129.102	2 10 129 211 11	ICMP	Destination unreachable (Port unreachable)
1 0 000/33	10.129.102.2	0 10.129.211 13	ICMP	Destination unreachable (Port unreachable)
2 0 000483	10.129.102.4	10.129.211.13	ICMP	Destination unreachable (Port unreachable)
3 0 0007 30	10.129.102.2	6 10 129 211 11	1CMP 1CMP	Destination unreachable (Port unreachable)
4 0.000/29	10.129.102.2		ICHP	Destination unreachable (Port unreachable)
5 0 000480 6 0 000730	10 129 102		ICHP	Destination unreachable (Port unreachable)
7 0 000487	10 129 102 0	10 120 211 11	ICMP	Destination unreachable (Port unreachable)
8 0 000 24	16 159 105 1	10 120 211 11	TCMP	best mation unreachable (Port unreachable)
9 0 000486	10 129 102	10 136 211 11	ICHP	Destination unreachable (Port unreachable)
0 0 0007 10	10 124 102	10 129 211 11	ICMP	Destination unreachable (Port unreachable)
1 0 000724	10 129 102 4	10.129.211.11	TCMP	Destination unreachable (Port unreachable)
2 0 000487	10 129 102.5	10 129 211 13	TCMP	Destination unreachable (Port unreachable)
3 0 000779	10 129 102 4	10 129 211 11	ICMP	Destination unreachable (Port unreachable)
A DESCRIPTION OF TAXABLE PARTY.				

Source		Destination	Protocol	late .
				fuscript > netbios-ssn [SYN] Seq=0 Win=64240 [
	.211.13	10.25.102.30		x9-icue > netbios-ssn [SYN] Seq=0 Win=64240 [1
				Destination unreachable (Port unreachable)
				audit-transfer > netbios-ssn [SYN] Seq=0 Win=0
				capioverlan > microsoft-ds [SYN] Seq=0 Win=642
				elfiq-repl > microsoft-ds [SYN] Seq=0 Win=6424
				bytsonar > microsoft-ds [SYN] Seq=0 Win=64240
				blaze > microsoft-ds [SYN] Seq=0 win=64240 [TC
				unizensus > microsoft-ds [SYN] Seq=0 win=64240
				winpoplanmess > microsoft-ds [SYN] Seq=0 Win=0
				c1222-acse > microsoft-ds [SVN] Seq=0 Win=6424
				resacommunity > microsoft-ds [SYN] Seq=0 Win=0
				nfa > microsoft-ds [SVN] Seq=0 Win=64240 [TCP
				iascontrol-oms > microsoft-ds [SYN] Seq=0 Win
				iascontrol > microsoft-ds [SYN] Seq=0 Win=6424
078 10.129	.211.13	10.25.102.11	TCP	dbcontrol-oms > microsoft-ds [SYN] Seg=0 Win=6
		10.25.102.12	TCP	oracle-oms > microsoft-ds [SVN] Seq=0 Win=6424
			TCP	olsv > microsoft-ds [SYN] Seq=0 Win=64240 [TCF
			TCP	health-polling > microsoft-ds [SYN] Seg=0 Win-
			TCP	health-trap > microsoft-ds [SYN] Seq=0 Win=642
			TCP	sddp > microsoft-ds [SYN] Seq=0 Win=64240 [TCF
080 10.129	.211.13	10.25.102.17		qsm-proxy > microsoft-ds [SYN] Seq=0 win=64240
				Destination unreachable (Port unreachable)
035 10.129	211.13	10.25.102.18	TCP	nsm-auf > microsoft-ds [SYN] SeawD Win=64240
	10.129           11.6         10.129           11.6         10.129           10.2         10.129           10.8         10.129           10.8         10.129           10.7         10.129           10.6         10.129           10.6         10.129           10.6         10.129 </td <td><math display="block">\begin{array}{c ccccccccccccccccccccccccccccccccccc</math></td> <td><math display="block">\begin{array}{c ccccccccccccccccccccccccccccccccccc</math></td> <td>10.129.211.13         10.25.102.30         TCP           111.10.129.211.13         10.25.102.31         TCP           112.10.129.211.13         10.25.102.31         TCP           110.129.211.13         10.25.102.31         TCP           110.129.211.13         10.25.102.31         TCP           110.129.211.13         10.25.102.31         TCP           10080         10.129.211.13         10.25.102.2         TCP           10070         10.129.211.13         10.25.102.2         TCP           10090         10.129.211.13         10.25.102.4         TCP           10077         10.129.211.13         10.25.102.6         TCP           10079         10.129.211.13         10.25.102.7         TCP           10077         10.129.211.13         10.25.102.6         TCP           10078         10.129.211.13         10.25.102.7         TCP           1078         10.129.211.13         10.25.102.10         TCP           1078         10.129.211.13         10.25.102.10         TCP           1078         10.129.211.13         10.25.102.11         TCP           1078         10.129.211.13         10.25.102.12         TCP           1077         10.129.211.13</td>	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	10.129.211.13         10.25.102.30         TCP           111.10.129.211.13         10.25.102.31         TCP           112.10.129.211.13         10.25.102.31         TCP           110.129.211.13         10.25.102.31         TCP           110.129.211.13         10.25.102.31         TCP           110.129.211.13         10.25.102.31         TCP           10080         10.129.211.13         10.25.102.2         TCP           10070         10.129.211.13         10.25.102.2         TCP           10090         10.129.211.13         10.25.102.4         TCP           10077         10.129.211.13         10.25.102.6         TCP           10079         10.129.211.13         10.25.102.7         TCP           10077         10.129.211.13         10.25.102.6         TCP           10078         10.129.211.13         10.25.102.7         TCP           1078         10.129.211.13         10.25.102.10         TCP           1078         10.129.211.13         10.25.102.10         TCP           1078         10.129.211.13         10.25.102.11         TCP           1078         10.129.211.13         10.25.102.12         TCP           1077         10.129.211.13

Figure 18 ICMP destination unreachable responses



We have got these scans going out on this system as shown in figure 19. Now, we can tell the host is infected with the part a lot of times are just by passively listening to what that host says when nobody is listening to what that host says when nobody is listening at the keyboard.

Here are infected host and the infected host is 10.129.211.13 as shown in figure 20. It first does a DNS query for "bbjj.househot.com". And it gets back a canonical name or an alias response indicating that the alias is "ypgw.wallloan.com.

ner:				apression_ C	Clear, Apply
6	Time	1000	Destination	Protocol	Info
1	0.000000	(10.129.211.13)	10.129.56.6	DNS	Standard query A bhii househot.com
2	0.237997	10.129.55.6	10.129.211.13	DNS	Standard query response CNAME ypgw.walloan.c
3	0.001861	10.129.211.13	216.234.235.16	5 TCP	neod1 > 18067 [SYN] Seq=0 WTN=64240 [1 P CHEC
	0.000549	216.234.235.16	5 10.129.211.13	ICMP	Destination unreachable (Port unreachable)
:5	2.999536	10.129.211.13	216.234.235.16	5 TCP	neod1 > 18067 [SYN] Seq=0 Win=64240 [TCP CHEC
Ð	0.000633	216.234.235.16	5 10.129.211.13	ICMP	Destination unreachable (Port unreachable)
in T	5.933724	10.129.211.13	216.234.235.16		neod1 > 18067 [SYN] Seq=0 Win=64240 [TCP CHEC
	0.000710	210.234.235.16	5 10, 129, 211, 13	TCM5.	Destination unreachable (Port unreachable)
	328.35307		10.129.56.6	DNS	Standard query A ypgw.wallloan.com
	0.228953	10.129.56.6	10.129.211.13	DNS	Standard query response A 61.189.243.240 A 61
	0.006457	10.129.211.13	61.189.243.240		neod2 > 18067 [SYN] Seq=0 Win=64240 [TCP CHEC
	0.396606	61.189.243.240		TCP	18067 > neod2 [SYN, ACK] Seq=0 Ack=1 Win=6553
	0.000185	10.129.211.13	61.189.243.240		neod2 > 18067 [ACK] Seq=1 Ack=1 Win=64240 [TC
	0.000095	10.129.211.13	61.189.243.240		neod2 > 18067 [PSH, ACK] Seq=1 Ack=1 Win=6424
	0.559178	61.189.243.240		TCP	18067 > neod2 [ACK] Seq=1 Ack=14 Win=65522 [T
	0.000050	10.129.211.13	61.189.243.240		neod2 > 18067 [PSH, ACK] Seq=14 Ack=1 Win=642
	0.402661	61.189.243.240		TCP	18067 > neod2 [PSH, ACK] Seq=1 Ack=31 Win=655
	0.000108	10.129.211.13	61.189.243.240		neod2 > 18067 [PSH, ACK] Seq=31 Ack=24 Win=64
	0.484319	61.189.243.240		TCP	18067 > neod2 [PSH, ACK] Seq=24 Ack=52 Win=65
	0.000058	10.129.211.13	61.189.243.240		neod2 > 18067 [PSH, ACK] Seq=52 Ack=80 Win=64
	0.398523 0.184217	61.189.243.240 10.129.211.13		TCP	18067 > neod2 [PSH, ACK] Seq=80 Ack=70 Win=65
	0.175701	10.129.211.13	61.189.243.240	DNS	<pre>neod2 &gt; 18067 [ACK] Seq=70 Ack=283 Win=63958 Standard guery A hometown.aol.com</pre>
	0 001193	10 129 56 6	10 129 211 13	DNS	Standard query & hometown.ao1.com
-		in the term	111 124 201 13		Standard deproversion a 2011 tax 220 246 4 2

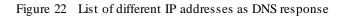
Figure 20 DNS query by the infected Host

ie Dr		Cepture Analyze Statistics	Telephony Iools Help	-							
a ana				Expression_ C	Q Q Q 🔟			4.8			
-	Time	Source	Destination	Protocol	Info						
	0,000000	10, 129, 211, 13	10,129,56,6	DNS		Query	A bb	i hour	ehot.com		
- 2	0.237997	10.129.56.6	11110 P-1-10-2 5 00 8-1	DNS	Standard	query	CC5-D	mse C/	AME VERW .	all loar	
3	0.001861	10.129.211.13	216.234.235.1	65 TCP	neod1 >	18067	[SYN]	Seq=0	win=64240	TCP CH	ECK
4	0.000549	216.234.235.16	5 10 129 211 13	ICMP	Destinat	1-0/1 U/U	reach	uble ()	Port unread	chable)	-
3.	2.999530	10,129,211,13	210.234.235.1	D3 TCP	neod1 >	19001	[SYN]	Seq=0	W10=04240	TCP CH	<b>IECK</b>
7	5,933724	10,129,211,13	216,234,235.10	65 TCP	neod1 >	18067	[SYN]	Seg=0	Win=64240	TCP CH	ECK
19	0.000710	216.234.235.16	5 10 129 211 13	ICMP	Destinat	ni ora - ura	neach	ble (	Port unreal	chable)	
.9	328.35307	3 10.129.211.13	10.129.56.6	DNS	Standard	query	A VD	w.wal	lloan.com		
	ain Name : Request In Time: 0.23 ransaction	7997000 seconds 1D: 0x0006 80 (Standard que	) 1			ack jack	(102	5)			
	nswer RRs:										
	uthority R dditional ueries nswers uthoritati dditional	RRs: 3									

Figure 21 Unusual numbers of Answer Resource Records in the DNS response

If we look at the response as shown in figure 21, there is a classical sign that may be a problem on the network. The response that came back has four portions: Questions, Answers, Authority, Additional RR (Resources Records). In response we get question restated back to us and we should get one or may be two (max.4) answer resource records. It is unusual to see 12 answer resource records and that is always a trigger that we want to pay attention.

Plant				• E	pression_ C	ieac Apply						
ko Time	Source		Destination	2	Protocol	Julia -						-
1 0.000000	10.129.211	.13	10.129.	56.6	DNS	Standard	d query	A bb;	j.hous	ehot.com		
2 0. 23/99/	10.129.56.0	12	10.129	210 11	5 TCP	Standare	1 query	respo	mse CN	win=64240	wallloan.	<u>co</u>
4 0 000540			10 129	.233.10	3 SCP	neod1 >	18007	LSYN	Sequel	windered	CINE CHE	5.80
5 2.999536	10.129.211	.13	216.234	4.235.16	S TCP	neod1 >	18067	[SYN]	Seq=0	win=64240	TCP CHE	CK
6 0.000633	216.234.23	5.165	10.129	211.13	ICMP	Destinat	cliber un	neach	ebile (P	ont unnea	chable)	
7 5.933724	10.129.211	.13	216.234	1.235.16	S TCP	neod1 >	18067	[SYN]	Seq=0	win=64240	TCP CHE	CK
9 328 3530	3 10 129 211	13	10 129	55.6	DNS	Standar	1 miler	A UT	the same 1 1	loan.com	CTHED FE 2	
					- Drives	Se c an rotat i				Total Tractoria		
Authority Additional	12 RRs: 2	_	_	_								
Authority Additional	12 RRs: 2	_	_									
Authority Additional @ Queries @ Answers	RRs: 2 RRs: 3					$\rightarrow$		_				
Answer BRO Authority Additional @ Queries @ Answers @ bbjj.hou	RRs: 2 RRs: 3 sehot.com: ty					pgw.wal	loan.co					
Answer RRS Authority Additional @ Queries @ Answers @ bbjj.hou @ ypgw.wal	RRs: 2 RRs: 3 sehot.com: ty lloan.com: ty	pe A,	class	IN, add	1 =0.2	94.233.10	loan.co					
Answer BR Authority Additional Queries Answers bbjj.hou ypgw.wal ypgw.wal	RRs: 2 RRs: 3 sehot.com: ty lloan.com: ty	pe A, pe A,	class class	IN, add IN, add	151.1	18.6.55	$\models$					
Answer RRC Authority Additional @ Queries @ Answers @ bbjj.hou @ ypgw.wal @ ypgw.wal @ ypgw.wal	RRs: 2 RRs: 3 sehot.com: ty lloan.com: ty lloan.com: ty lloan.com: ty	/pe A, /pe A, /pe A,	class class class	IN, add IN, add IN, add	151.19	14.233.10 18.6.55 14.247.19						
Answer BR Authority Additional Queries Answers bjj.hou gypgw.wal gypgw.wal gypgw.wal gypgw.wal	RRs: 2 RRs: 3 sehot.com: ty lloan.com: ty lloan.com: ty lloan.com: ty	pe A, pe A, pe A, pe A,	class class class class	IN, add IN, add IN, add IN, add	151.19 216.2 68.11	94.233.10 98.6.55 94.247.19 2.229.228	T					
Answer ERC Authority Additional Queries Answers bbjj.hou ypgw.wal ypgw.wal ypgw.wal ypgw.wal ypgw.wal	RRs: 2 RRs: 3 sehot.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty	pe A, pe A, pe A, pe A, pe A,	class class class class class class	IN, add IN, add IN, add IN, add IN, add	151.19 216.2 68.11 61.18	08.6.55 04.247.19 2.229.228 0.243.240	T					
Answer RDC Authority Additional @ Queries @ Answers @ bbj.hou @ ypgw.wal @ ypgw.wal @ ypgw.wal @ ypgw.wal @ ypgw.wal	RRs: 2 RRs: 3 sehot.com: ty lloan.com: ty lloan.com: ty lloan.com: ty	/pe A, /pe A, /pe A, /pe A, /pe A,	class class class class class class class	IN, add IN, add IN, add IN, add IN, add IN, add	151.19 216.2 68.11 61.18 218.1	04.233.10 08.6.55 04.247.19 2.229.228 0.243.240 2.94.58	T					
Answer ER Authority Additional Queries Answers bjj.hou ypgw.wal ypgw.wal ypgw.wal ypgw.wal ypgw.wal ypgw.wal ypgw.wal ypgw.wal	RRs: 2 RRs: 3 sehot.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty	/pe A, /pe A, /pe A, /pe A, /pe A, /pe A,	class class class class class class class class	IN, add IN, add IN, add IN, add IN, add IN, add IN, add	151.19 216.2 68.11 61.189 218.11 61.14	04.235.10 08.6.55 04.247.19 2.229.228 0.243.240 2.94.58 0.119.63	T					
Answer ERC Authority Additional Queries Answers bbjj.hou ypgw.wal ypgw.wal ypgw.wal ypgw.wal ypgw.wal ypgw.wal ypgw.wal ypgw.wal	RRs: 2 RRs: 3 sehot.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty	/pe A, /pe A, /pe A, /pe A, /pe A, /pe A,	class class class class class class class class class	IN, add IN, add IN, add IN, add IN, add IN, add IN, add IN, add	151.19 216.2 68.11 61.18 218.11 61.14 61.14 202.9	04.235.10 08.6.55 04.247.19 2.229.228 0.243.240 2.94.58 0.119.63 0.223.87	F	<u>A</u> IIII				
Answer RDC Authority Additional @ Queries @ Answers @ bbj.hou @ ypgw.wal @ ypgw.wal @ ypgw.wal @ ypgw.wal @ ypgw.wal @ ypgw.wal @ ypgw.wal	RRs: 2 RRs: 3 sehot.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty lloan.com: ty	/pe A, /pe A, /pe A, /pe A, /pe A, /pe A, /pe A,	class class class class class class class class class class	IN, add IN, add IN, add IN, add IN, add IN, add IN, add IN, add IN, add	151.19 216.2 68.11 61.18 218.1 61.14 202.9 218.2	04.233.10 08.6.55 14.247.19 1.229.228 0.243.240 2.94.58 5.119.63 5.223.87 19.83.118	F					



But when we open the answer section as shown in figure 22, we see the "ypgw.wallloan.com" that is alias for "bbjj.househot.com" and here are all of the different IP addresses that are assigned to "ypgw.wallloan.com". Now the presence of lot of IP addresses makes us very concern because it is very unusual to see that. Most of the times the presence of many IP addresses, is a list of IRC Servers. In packet number 3, the client goes out and does a SYN to port "18067". Anything can run on this or any port that is why port filtering devices are very limited because we can go round that by using other ports for our services.

Phane		• t <sub>il</sub>	ression_ C	wag Apply						
lo Time	Source	Destination	Protocol	Info						
1 0.000000	10.129.211.13	10.129.56.6	DNS	Standard	query	A bbj	j.hous	ehot.com		_
3 0,001861	10.129.56.6	016 324 335 165	TCP	neod1 >	18067	E COMPLET	Face Chi	win=64240	ETCD Chi	- 60
4 0.000549			TOMP	Destinat	1 on un	LISTRI	blette	art unread	annalia	CLA
5 2.999536	10.129.211.13	216.234.235.165	ТСР 🖌	neod1 >	18067	[SYN]	Seq-0 1	win=64240	TCP CH	ECK
6 0 000633	216 234 235 16	500012992101018	ICMP	Sestinat	TOT UT	ireacha	ble (P	ort unread	chable)	-
7 2.933724	10.129.211.15	210.234.235.103	ICP V	neod1 >	1900-	[SYN]	SequU I	#119=04240	TCP CH	ECK
9 328, 35307	3 10, 129, 211, 13	10,129,56.6	DNS	St. ndard	query	A VDO	w.wall	loan.com	7	
Internet Pro	ptocol, Src: 10.1 um Protocol, Src	04:f8:35 (00:90:7 129.56.6 (10.129, Port: domain (53	56.6).	Ost: 10.1	29,211	.13 0	10.129.		db:58:93	: f
Internet Pro	Stocol, Src: 10.1 Im Protocol, Src System (response	29.56.6 (10.129. Port: domain (53	56.6).	Ost: 10.1	29,211	.13 0	10.129.		db:58:93	:f
Internet Pro User Datagra Domain Name [Request I [Time: 0.2	stocol, Src: 10.1 im Protocol, Src System (response n: 11 37997000 seconds	129.56.6 (10.129. Port: domain (53	56.6).	Ost: 10.1	29,211	.13 0	10.129.		db:58:93	is fi
Internet Pro User Datagri Domain Name [Request I [Time: 0.2 Transactio	tocol, Src: 10.1 im Protocol, Src System (response n: 11 37997000 seconds n ID: 0x0006	129.56.6 (10.129. Port: domain (53 e) ]	56.6). ). Dst	Ost: 10.1	29,211	.13 0	10.129.		db:58:93	: 17
Internet Pro User Datagri Domain Name <u>Request I</u> [Time: 0.2 Transactio Flags: 0x8	tocol, Src: 10. im Protocol, Src System (response n: 11 37997000 seconds n ID: 0x0006 580 (Standard qu	129.56.6 (10.129. Port: domain (53	56.6). ). Dst	Ost: 10.1	29,211	.13 0	10.129.		db:58:93	:fi
Internet Pro User Datagra Domain Name [Request I [Time: 0.2 Transactio Flags: 0x8 Questions:	incol, sec: 10.1 um Protocol, Sec System (response n: 11 37997000 seconds n ID: 0x0006 580 (Standard qu 1	129.56.6 (10.129. Port: domain (53 e) ]	56.6). ). Dst	Ost: 10.1	29,211	.13 0	10.129.		db:58:93	: fi
Internet Pro User Datagri Domain Name <u>[Request 1</u> [Time: 0.2 Transactio = Flags: 0x8 Questions: Answer RRs	tocol, src: 10.1 im Protocol, Src System (response n: 11 37997000 seconds n ID: 0x0006 580 (Standard qu 1 : 12	129.56.6 (10.129. Port: domain (53 e) ]	56.6). ). Dst	Ost: 10.1	29,211	.13 0	10.129.		db:58:93	: fi
Internet Pro User Datagri Domain Name <u>[Request I</u> [Time: 0.2 Transactio Flags: 0x8 Questions: Answer RRS Authority	incol, sec: 10. m Protocol, Sec System (response n: 11 37997000 seconds n: 10: 0x0006 580 (Standard qu 1 : 12 RRs: 2	129.56.6 (10.129. Port: domain (53 e) ]	56.6). ). Dst	Ost: 10.1	29,211	.13 0	10.129.		db:58:93	: fi
Internet Pro Joser Datagri Josen In Name <u>Frequest I</u> [Time: 0.2 Transactio Flags: 0x8 Questions: Answer RRs Authority Additional	incol, sec: 10. m Protocol, Sec System (response n: 11 37997000 seconds n: 10: 0x0006 580 (Standard qu 1 : 12 RRs: 2	129.56.6 (10.129. Port: domain (53 e) ]	56.6). ). Dst	Ost: 10.1	29,211	.13 0	10.129.		db:58:93	:f
Internet Provide Provi	incol, sec: 10. m Protocol, Sec System (response n: 11 37997000 seconds n: 10: 0x0006 580 (Standard qu 1 : 12 RRs: 2	129.56.6 (10.129. Port: domain (53 e) ]	56.6). ). Dst	Ost: 10.1	29,211	.13 0	10.129.		db:58:93	:f
Internet Pro User Datagr Domain Name <u>Frequest I</u> [Time: 0.2 Transactio Flags: 0x8 Questions: Answer RRs Authority Additional Queries Answers	m Protocol, Src: 10.1 m Protocol, Src System (response n: 11 37997000 seconds n ID: 0x0006 580 (Standard qu 1 : 12 RRs: 2 RRs: 3	29.56.6 (10,129, Port: domain (53 5) ] ery response, No	error)	Dst: 10.1 Port: bla	129,211 tckjaci	1.13 (102) (102)	10.129.		db:58:93	i: fi
Internet Provide a construction of the constru	in Protocol, Sec. 10. m Protocol, Sec. System (response n: 11 37997000 seconds n: 10: 0x0006 580 (Standard qu 1 : 12 RRs: 2 RRs: 3 sehot.com: type 0	129.56.6 (10.129. Port: domain (53 e) ]	56.6), ), Dst error)	Dat: 10.1 Port: 51	29,211 ickjaci	1.13 (102) (102)	10.129.		db:58:93	:= f
Internet Pro User Datagr Domain Name <u>Frequest I</u> [Time: 0.2 Transactio # Flags: 0x8 Questions: Answer RRs Authority Additional @ Queries # Answers # bbjj.hou # bbjj.hou	mprotocol, sec: 10. m Protocol, Sec System (response n: 11 37997000 seconds n: 10: 0x0006 580 (Standard qu 1 : 12 RRs: 2 RRs: 3 sehot.com: type ( 10an.com: type (	<pre>[29.56.6 (10,129. Port: domain (53 b) ] ery response, No CNAME, class IN,</pre>	error)	Dat: 10.1 Port: 61	29,211 ickjaci	1.13 (102) (102)	10.129.		db:58:93	i: f
Internet Pro User Datago Domain Name <u>Frequest I</u> [Time: 0.2 Transactio Flags: 0x8 Questions: Answer RRS Authority Additional Queries Asswers bbjj.hou yppw.wal yppw.wal	incool, sec: 10 m Protocol, Sec System (response n: 11 37997000 seconds n ID: 0x0006 580 (Standard qu 1 : 12 RRs: 2 RRs: 3 sehot.com: type ( lloan.com: type ) lloan.com: type (	<pre>[29.56.6 (10,129, Port: domain (53 e) ] ery response, No CNAME, class IN, A, class IN, addr</pre>	cror) (216.2) (216.2) (216.2) (216.2)	percenti 4.235.16 8.6.55	ick jack	1.13 (102) (102)	10.129.		db:58:93	is fi

Figure 23 Unsuccessful TCP Handshakes

Now we look at the response that came back as shown in figure 23, the very first IP address that came in the response is "216.234.235.165" and sure enough that is the first target that the bot infect host wants to make a handshake. Here is the TCP Handshake going out and the destination unreachable (port unreachable) coming back.

ter:				• t <sub>0</sub>	pression_ C	was Apply					
. Time	Source		Destination	0	Protocol	Info					
5 2.999536	10.129.2	11.13	216.234	4.235.163	TCP	neod1	> 18067	[SYN]	Seq=0	Win=64240	TCP CHEC
7 5,933724	10,129,2	235.165	216.23	4.235.16	TCP	neod1	> 18067	[SYN]	Sequ0	win=64240	TCP CHEC
8 0.000710	216.234.	235.165	10,129	2111113	ICMP	Destin		nneach	able G	Pont unnead	
9 328.35307	3 10.129.2	11.13	10.129	.56.6	DNS	Standa	nd quer	A YP	dw.wa1	lloan.com	
1 0.006457	10.129.2	11.13	61.189	.243.240	TCP	neod2	> 18067	[SYN]	Seq=0	win=64240	TCP CHEC
2 0.396606	61.189.2	43.240	10.129	243 240	TCP	18067	> neod2	ESYN.	ACK] :	Seq=0 Ack=]	1 win=6553
[Time: 0.2 Transactio Flags: 0x8	28953000 s n ID: 0x00	07	ry resp	onse, No	error)						
[Time: 0.2 Transactio Flags: 0x8 Questions: Answer RRs Authority	28953000 s n ID: 0x00 580 (Stand 1 : 11 RRs: 0	07	ry resp	onse, No	error)						
[Time: 0.2 Transactio Flags: 0x8 Questions: Answer RRs Authority Additional	28953000 s n ID: 0x00 580 (Stand 1 : 11 RRs: 0	07	ry resp	onse, No	error)						
[Time: 0.2 Transactio Flags: 0x8 Questions: Answer RRs Authority Additional Queries	28953000 s n ID: 0x00 580 (Stand 1 : 11 RRs: 0	07	ry resp	onse, No	error)						
[Time: 0.2 Transactio Flags: 0x8 Questions: Answer RRs Authority Additional Queries	28953000 s n ID: 0x00 580 (Stand 1 : 11 RRs: 0 RRs: 0 RRs: 0	07 ard que				). 243. 2	10				
[Time: 0.2 Transactio Flags: 0x8 Questions: Answer RRs Authority Additional Queries Answers = ypgw.wall = ypgw.wall	28953000 s n ID: 0x00 580 (Stand 1 : 11 RRs: 0 RRs: 0 NRs: 0	07 and que type A type A	class class	IN, addr IN, addr	61.189	5.119.6					
[Time: 0.2 Transactio Flags: 0x8 Questions: Answer RRs Authority Additional Queries Answers = ypgw.wall = ypgw.wall	28953000 s n ID: 0x00 580 (Stand 1 : 11 RRs: 0 RRs: 0 RRs: 0 110an.com: 110an.com:	07 ard que type A type A type A	class class class	IN, addr IN, addr IN, addr	61.189	5.119.6 8.6.55	3				
[Time: 0.2 Transactio Flags: 0x8 Questions: Answer RRs Authority Additional Queries Answers Eypgw.wall Eypgw.wall Eypgw.wall	28953000 s n ID: 0x00 580 (Stand 1 : 11 RRs: 0 RRs: 0 RRs: 0 RRs: 0 110an.com: 110an.com: 110an.com:	07 and que type A type A type A	class class class class	IN, addr IN, addr IN, addr IN, addr	61.189 61.149 151.19 202.98	0.119.6 8.6.55 3.223.8	3				
[Time: 0.2 Transactio Flags: 0x8 Questions: Answer RRs Authority Additional Queries Answers ypgw.wall ypgw.wall ypgw.wall ypgw.wall	28953000 s n ID: 0x00 580 (Stand 1 : 11 RRS: 0 RRS: 0 NRS: 0 NOAN.com: 10aan.com: 10aan.com: 10aan.com:	type A type A type A type A type A	, class , class , class , class , class , class	IN, addr IN, addr IN, addr IN, addr IN, addr	61.189 61.149 151.19 202.98 218.24	0.119.6 08.6.55 0.223.8 19.83.1	3 7 18				
[Time: 0.2 Transactio Flags: 0x8 Questions: Authority Additional Queries Answers ypgw.wall ypgw.wall ypgw.wall ypgw.wall ypgw.wall	28953000 s n ID: 0x00 580 (Stand 1 : 11 RRs: 0 RRs: 0 Iloan.com: Iloan.com: Iloan.com: Iloan.com: Iloan.com:	07 ard que type A type A type A type A type A	class class class class class class class	IN, addr IN, addr IN, addr IN, addr IN, addr IN, addr	61.189 61.149 151.14 202.98 218.24 68.186	5.119.6 N8.6.55 S.223.8 N9.83.1 5.110.1	3 7 18 58				
[Time: 0.2 Transactio Flags: 0x8 Questions: Answer RRs Authority Additional Queries Answers ypgw.wall ypgw.wall ypgw.wall ypgw.wall	28953000 s n ID: 0x00 580 (Stand 1 : 11 RRs: 0 RRs: 0 RRs: 0 Iloan.com: Iloan.com: Iloan.com: Iloan.com: Iloan.com:	07 ard que type A type A type A type A type A	class class class class class class class	IN, addr IN, addr IN, addr IN, addr IN, addr IN, addr	61.189 61.149 151.19 202.98 218.24 68.186 68.112	5.119.6 8.6.55 8.223.8 9.83.1 5.110.1 2.229.2	3 7 18 58				

Figure 24 DNS response for ypgw.wallloan.com

Now this makes us feel that the target system has got some firewall process to something loaded which is responding ICMP instead of TCP reset or TCP [SYN, ACK]. The client tries again, it is unsuccessful, it tries again, and it is unsuccessful. Then the client gives up and does a DNS query for "ypgw.wallloan.com". It is now going after the canonical name. For its DNS reply we will look into the answer section as shown in figure 24

In the answer section we see the "ypgw.wallloan.com" and there are number of different IP addresses associated with that. The list of IP addresses is probably the list of IRC Commanding and Controlling Servers because it is very typical to see.

pter:			- Eas	vession_ C	lang Apply	
N40	Time	Source	Destination	Protocol	lefo	
5	2.999536	10.129.211.13	216.234.235.165	TCP	neod1 > 18067 [SYN] Seq=0 Win=64240 [TCP CHEC	×
0	5 933724	10 129 211 13	216 234 235 165	TCP	<pre>Destination unreachable (Port unreachable) neod1 &gt; 18067 [SVN] Seg=0 win=64240 [TCP CHEC</pre>	10
- 8	0.000710	1410 144 14 14 14 15 14 15 14 15 14 15 15 14 15 15 15 15 15 15 15 15 15 15 15 15 15		ROME	Destination unreachable (Port unreachable)	
9	328.353073	10.129.211.13	10.129.56.6	DNS	Standard query A ypgw.wallloan.com	
_10	0.006457	10.129.56.6	61,189,243,240	TCP	<pre>Standard overv response A 61.189.243.240 A 61 neod2 &gt; 18067 [SVN] Seg=0 win=64240 [TCP CHEC</pre>	
	0.396606	61, 189, 243, 240	10,129,211,13	TCP	<pre>neod2 &gt; 18067 [SVN] Seq=0 Win=64240 [TCP CHEC 18067 &gt; neod2 [SVN, ACK] Seq=0 Ack=1 Win=6553</pre>	
	0.000185	10.129.211.13	61.189.240 240	TCP	neod2 > 18067 [ACK] Seg=1 Ack=1 Win=64240 [TC	
	0.000095	10.129.211.13	61.189.243.240	TCP	neod2 > 18067 [PSH, ACK] Seq=1 Ack=1 Win=6424	
	0.559178	61.189.243.240	10.129.211.13	TCP	18067 > neod2 [ACK] Seq=1 Ack=14 Win=65522 [T	
	0.000050	10.129.211.13 61.189.243.240	61.189.243.240 10.129.211.13	TCP	<pre>neod2 &gt; 18067 [PSH, ACK] Seq=14 Ack=1 Win=642 18067 &gt; neod2 [PSH, ACK] Seq=1 Ack=31 Win=655</pre>	
	0.000108	10,129,211,13	61, 189, 243, 240	TCP	neod2 > 18067 [PSH, ACK] Seq=31 Ack=31 Win=64	
	0.484319	61.189.243.240	10.129.211.13	TCP	18067 > neod2 [PSH, ACK] Seg=24 Ack=52 Win=65	
	0.000058	10.129.211.13	61.189.243.240	TCP	neod2 > 18067 [PSH, ACK] Seq=52 Ack=80 Win=64	
	0.398523	61.189.243.240	10.129.211.13	TCP	18067 > neod2 [PSH, ACK] Seq=80 Ack=70 Win=65	4
22	0.184217	10.129.211.13	61.189.243.240	TCP	neod2 > 18067 [ACK] Seq=70 Ack=283 Win=63958	
-						*
	Request In					
		8953000 seconds]				
		ID: 0x0007				
		80 (Standard que	ry response, No	error)		
	westions:					
	inswer RRs:					
	withority R					
	ueries	RRS: U				
	ueries.					

Figure 25 TCP Handshake between the client and the target system

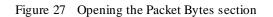
The first address in the list is "61.189.243.240" and sure enough the client goes out and sure enough the client goes out and does a Handshake to that target system as shown in figure 25. There is a SYN packet; it is going out on port number "18067", which we know that anything can run on that port.

Telephony Iools H	elp							
* * * * * *		QQQ		0 🔧 %	100			
-	Expression	Clear Apply						
Destination	Protocol	Info						_
216.234.235.	165 TCP	neod1	> 18067	[SYN]	Seq=0	Win=64240	[TCP CHEC	K:
10.129.211.1	3 ICMP	Desti	nation un	nreach	110 I C 1	Port unread	hable)	
216.234.235.		neodl	> 18067	[SYN]		Win=64240	[TCP CHEC	K:
10.129.211.1	3 ICMP	Destin	nation un	nreach		Port unread	thable)	
10.129.56.6	DNS					lloan.com		
61, 189, 243, 2	and the second se	AND I LOT	> 18067	And in case of the local division of the loc	Contract A	61 189 24 Win=64240	and the second state of th	-
10, 129, 211 1	3 TCP	neod2 18067	> neod2	SYN.		eq=0 Ack=1		
61, 189, 243, 2		neod2				Ack=1 Win=		
61.189.243.2		and the second se				Seg=1 Ack=1		
10,129,211,1		18067	> Seod2			Ack=14 Wir		
61, 189, 243, 2			>-18067	[PSH.		Seg=14 Ack=		
10,129,211,1		18067	> neod2	[PSH.		Seg=1 Ack=3		
61.189.243.2	40 TCP	neod2	> 18067	[PSH,		Seg=31 Ack=		
10.129.211.1		18067	> neod2	[PSH.		Seg=24 Ack=		
61.189.243.2	40 TCP	neod2	> 18067	[PSH.		Seg=52 Ack=		
10.129.211.1	3 TCP	18067	> neod2	[PSH,	ACK] :	Seq=80 Ack=	70 Win=65	41
61.189.243.2	40 TCP	neod2	> 18067	[ACK]	Seg=7	0 Ack=283 W	vin=63958	E-

Figure 26 Push flag sent with the ACK after the successful TCP connection

In this case the client is successful. We see the [SYN, ACK] came back and the ACK and the 3 way Handshake is completed. After that we see that the client immediately sends data up to that server using the Push flag which is also unusual to see as shown in figure 26.

and the second s	Wireless Toolbar		• fap	ression_ C	Clear: Apply
to Ter	Statusber		tion	Pretocol	Info
	Packet List		234.235.165		neod1 > 18067 [SYN] Seq=0 Win=64240 [TCP CHEC
7 5.	Packet Details		29.211.13	TCP	Destination unreachable (Port unreachable)
8.0	P tast Bytes		234.235.105	ICMP	neodl > 18067 [SYN] Seq=0 Win=64240 [TCP CHEC
9 32	Time Display Format		29.56.6	DNS	Standard guery A ypgw.wallloan.com
10 0.	Name Recolution		29.211.13	DNS	Standard query response A 61.189.243.240 A 61
11 0.	Colorize Packet List		89.243.240	TCP	neod2 > 18067 [SYN] Seq=0 Win=64240 [TCP CHEC
12 0.	Auto Scroll in Live Capture		50 243 240	TCP	18067 > neod2 [SYN, ACK] Seq=0 Ack=1 Win=6553
14 0. Q		Chiles	89.243.240	TCP	neod2 > 18067 [PSH, ACK] Seg=1 Ack=1 Win=6424
15 0.	goom an		29.211.13	TCP	18067 > neod2 [ACK] Seq=1 Ack=14 Win=65522 [T
16 0. 9	Loom Out		89.243.240	TCP	<pre>neod2 &gt; 18067 [PSH, ACK] Seq=14 Ack=1 Win=642</pre>
	Normal Size Resize All Columns	Ctrl++ Shift+Ctrl+R	29.211.13	TCP	18067 > neod2 [PSH, ACK] Seq=1 Ack=31 Win=655 neod2 > 18067 [PSH, ACK] Seq=31 Ack=24 Win=64
19 0.	Reside All Columns	Shift+CDS+R	29.211.13	TCP	18067 > neod2 [PSH, ACK] Seq=24 Ack=52 Win=65
20 0.	Eggland Subtrees	Shift+Right	89.243.240	TCP	neod2 > 18067 [PSH, ACK] Seq=52 Ack=80 Win=64
21 0.	Expand All		29.211.13	TCP	18067 > neod2 [PSH, ACK] Seq=80 Ack=70 Win=65
22 0.	Collapse <u>6</u> 8	Ctrl+Left	89.243.240	TCP	neod2 > 18067 [ACK] Seq=70 Ack=283 Win=63958
_	Colorize Conversation				
Frame	Reset Coloring 3-10	Chrl-Space	tes capture	d)	
Ether 🐔	Coloring Rules		Fa (00:0b:d	b:58:93	3:fa), Dst: Watchgua_04:f8:35 (00:90:7f:04:f8:3
Inter	Show Packet in New Window		.13 (10.12	9.211.1	13), Dst: 61,189,243,240 (61,189,243,240)
Trant 🕫	Reined	Ctri+R	: Port: neo	d2 (104	48), Dst Port: 18067 (18067), Seq: 1, Ack: 1, L



In this case the data is not buffered at all and is delivered right away; maybe there is something like a Telnet communication. But we do not recognize and Wireshark recognizes what is running on the port "18067". We will go to "View "option then we will select "Packet Bytes" as shown in figure 27

[ACK] Seq=1 Ack=1 Win=642 [PSH, ACK] Seq=1 Ack=1 Win=65 [PSH, ACK] Seq=1 Ack=14 Win=65 [PSH, ACK] Seq=14 Ack=1 W [PSH, ACK] Seq=14 Ack=31 W [PSH, ACK] Seq=31 Ack=24 [PSH, ACK] Seq=24 Ack=52 [PSH, ACK] Seq=24 Ack=52 [PSH, ACK] Seq=52 Ack=80 [PSH, ACK] Seq=80 Ack=70 [ACK] Seq=70 Ack=283 Win=
[ACK] Seq=1 Ack=14 Win=65 [PSH, ACK] Seq=14 Ack=1 W [PSH, ACK] Seq=1 Ack=31 W [PSH, ACK] Seq=31 Ack=24 [PSH, ACK] Seq=24 Ack=52 [PSH, ACK] Seq=52 Ack=80 [PSH, ACK] Seq=80 Ack=70
[PSH, ACK] Seq=14 Ack=1 W [PSH, ACK] Seq=1 Ack=31 W [PSH, ACK] Seq=31 Ack=24 [PSH, ACK] Seq=24 Ack=52 [PSH, ACK] Seq=52 Ack=80 [PSH, ACK] Seq=80 Ack=70
<pre>[PSH, ACK] Seq=1 Ack=31 W [PSH, ACK] Seq=31 Ack=24 [PSH, ACK] Seq=24 Ack=52 [PSH, ACK] Seq=52 Ack=80 [PSH, ACK] Seq=80 Ack=70</pre>
[PSH, ACK] Seq=31 Ack=24 [PSH, ACK] Seq=24 Ack=52 [PSH, ACK] Seq=52 Ack=80 [PSH, ACK] Seq=80 Ack=70
[PSH, ACK] Seq=24 Ack=52 [PSH, ACK] Seq=52 Ack=80 [PSH, ACK] Seq=80 Ack=70
[PSH, ACK] Seq=52 Ack=80 [PSH, ACK] Seq=80 Ack=70
[PSH, ACK] Seq=80 Ack=70
[ACK] Seq=70 Ack=283 Win=
tchgua_04:f8:35 (00:90:7f: 39.243.240 (61.189.243.240 18067 (18067), Seq: 1, Act
хе.
= .
4LP. eR 1 1 1

Figure 28 Data sent to the destination "61.189.243.240"

Then we will look into the packet bytes section and try to understand what data is going through the packets. We can see the client sent data up to the Server .We can see it is saying "User (space) l (space) l (space) l (space) l" going up to the server as shown in figure 28. Then we see the ACK coming back. Then we see the client sending some additional information as shown in figure 29.

	10.129.	211.13	ILP	10001	> neouz	LOTN,	ALK J SEC	HEO ACKET MI
.13	61.189.	243.240	TCP	neod2	> 18067	[ACK]	Seq=1 Ad	ck=1 Win=642
.13	61.189.	243.240	TCP	neod2	> 18067	[PSH,	ACK] Sec	q=1 Ack=1 Wi
. 240	10.129.	211.13	TCP	18067	> neod2	[ACK]	Seg=1 Ad	ck=14 Win=65
. 13	61.189.	243.240	TCP	neod2	> 18067	[PSH.	ACK] Sec	g=14 Ack=1 W
.240	10.129.	211.13	TCP	18067	> neod2	[PSH.	ACK] Sec	g=1 Ack=31 W
.13	61.189.	243.240	TCP	neod2	> 18067	[PSH.	ACK] Sec	g=31 Ack=24
.240	10.129.	211.13	TCP	18067	> neod2			=24 Ack=52
.13	61,189.	243.240	TCP		> 18067			=52 Ack=80
.240	10,129.		TCP		> neod2			=80 Ack=70
.13		243.240	TCP		> 18067			Ack=283 Win=
						C		
re.	71 bytes	capture	(b					
				3·fa)	Dst. Wat	chaua	04 - 18 - 35	(00:90:7f:0
	0. 33. 1a	100.00.0	0		Dat. Hat	cingua_		
10 1	20 211 1							
		3 (10.12	9.211.	.13), Ds	t: 61.18	9.243.	240 (61.)	189.243.240
		3 (10.12	9.211.	.13), Ds	t: 61.18	9.243.	240 (61.)	
		3 (10.12	9.211.	.13), Ds	t: 61.18	9.243.	240 (61.)	189.243.240
		3 (10.12	9.211.	.13), Ds	t: 61.18	9.243.	240 (61.)	189.243.240
toco	l, Src P	3 (10.12 ort: neo	9.211. d2 (10	13), Ds 048), Do	t: 61.18	9.243. 18067	240 (61. (18067),	189.243.240
toco Ob	1, Src P db 58 9	3 (10.12 ort: neo	9.211. d2 (10	00 ···	t: 61.18	9.243.	240 (61. (18067),	189.243.240
0b 06	db 58 9	3 (10.12 ort: neo 3 fa 08 a 81 d3	9.211. d2 (10 00 45 0d 3d	00 bd .9	t: 61.18 t Port:	9.243. 18067 ХЕ	240 (61. (18067).	189.243.240
0b 06 d8	db 58 9 00 00 0 34 f9 e	3 (10.12 ort: neo 3 fa 08 a 81 d3 d 88 e5	9.211. d2 (10 00 45 0d 3d 4c 50	00 bd .9	t: 61.18 t Port: *@4	9.243. 18067 XE	240 (61. (18067),	189.243.240
0b 06	db 58 9 00 00 0 34 f9 e	3 (10.12 ort: neo 3 fa 08 a 81 d3	9.211. d2 (10 00 45 0d 3d 4c 50	13), Ds 048), Ds 048)	t: 61.18 t Port: 5 .*@4 F4	9.243. 18067 ХЕ	240 (61. (18067),	189.243.240
0b 06 d8	db 58 9 00 00 0 34 f9 e	3 (10.12 ort: neo 3 fa 08 a 81 d3 d 88 e5	9.211. d2 (10 00 45 0d 3d 4c 50	13), Ds 048), Ds 048)	t: 61.18 t Port: *@4	9.243. 18067 XE	240 (61. (18067),	189.243.240
0b 06 d8	db 58 9 00 00 0 34 f9 e	3 (10.12 ort: neo 3 fa 08 a 81 d3 d 88 e5	9.211. d2 (10 00 45 0d 3d 4c 50	13), Ds 048), Ds 048)	t: 61.18 t Port: 5 .*@4 F4	9.243. 18067 XE	240 (61. (18067),	189.243.240
0b 06 d8	db 58 9 00 00 0 34 f9 e	3 (10.12 ort: neo 3 fa 08 a 81 d3 d 88 e5	9.211. d2 (10 00 45 0d 3d 4c 50	13), Ds 048), Ds 048)	t: 61.18 t Port: 5 .*@4 F4	9.243. 18067 XE	240 (61. (18067),	189.243.240

Figure 29 Additional information is sent to the destination "61.189.243.240"

In order to read this information right click on one of those packets and choose to follow the stream. We can follow the TCP stream, UDP stream or follow the SSL stream as shown in figure 30.

. 240	10.152.511.12	ICF	1000/ > HEOUZ [STN, ACK] SEC	T=O WCK=T MI
.13	61.189.243.240	) TCP	neod2 > 18067 [ACK] Seq=1 Ad	ck=1 Win=642
.13	61.189.243.240	) TCP	neod2 > 18067 [PSH, ACK] Sec	q=1 Ack=1 Wi
. 240	10.129.211.13	TCP	18067 > neod2 [ACK] Seq=1 Ad	ck=14 Win=65
13	61.189.243.240	О ТСР	18057 [DCI 16K] See	q=14 Ack=1 W
. 240	10.129.211.13	TCP	Mark Packet (toggle) K] Sec	q=1 Ack=31 W
.13	61.189.243.240	) TCP	Set Time Reference (toggle)     K] Sec	a=31 Ack=24
. 240	10,129,211,13	TCP	ki se	=24 Ack=52
.13	61, 189, 243, 240		Apply as Filter K1 Sec	=52 Ack=80
. 240	10.129.211.13	TCP		=80 Ack=70
.13	61, 189, 243, 240			Ack=283 Win=
			Colorize Conversation	
			SCTP +	
ire.	71 bytes captur	ed)	Follow TCP Stream	
	8:93:fa (00:0b:	and the second se		(00:90:7f:0
-	29.211.13 (10.1			189.243.240
otoco	1, Src Port: ne	eod2 (104	Copy , 06/),	Seq: 14, Ad
		HT .	2 Decode As	
O Ob	db 58 93 fa 08	00 45 0	Print_	
0 06	00 00 0a 81 d3			
e d8	34 f9 ed 88 e5			
e 69	43 4b 20 70 38			
3			196671.	

Figure 30 Following the TCP stream

Here the TCP stream is available for us so we will go to it. When we click on it, a window pops up and it shows exactly what data transferred between the client and the server. The client's data will by default be in "Red" and any data send by the server will by default be in "blue". This is an IRC communication it contains the User command, Nick command, User host command and especially the join command as shown in figure 31.

	Conten							
: a7	001	-00196 p8-00	196671	<b>•</b>	-			
USeR	RHOST	p8-0	019667	1	_	0196671=	1001	0 120
JOIN	1 #p8	3 ihod	c9hi					
:a7	332	p8-00	196671	#p8	: !	Q		
gfca	igihe	ehehad	kcpcpg	gigpgi	ngf	Fhegphho	jocogb	gpgmco
: a7	333	p8-00	196671	#p8	a	1134159	9047	
: a7	366	p8-00	196671	L #p8	=			

Figure 31 IRC communication

So at this port, we can tell this client is automatically connecting to the IRC Server in the background. Now we know that the client is connecting to the IRC Server.

Next, the client goes out and it does a query for "hometown .com" as shown in figure 32. The client gets a response, tries to make a connection, it is an unsuccessful connection attempt and then it begins its scanning process. So probably something during that IRC command exchange, something in the network client begins the scan process.

Best       Destination       Protocol       Infe         16       0.000000       10.129.211.13       01.189.243.240       TCP       neod2       PSH. ACK       Sequel Acks1 wine6424         17       0.402661       61.189.243.240       10.129.211.13       TCP       18067       neod2       PSH. ACK       Sequel Acks1 wine6550         18       0.00108       10.129.211.13       61.189.243.240       TCP       neod2       18067       PSH. ACK       Sequel Acks24 wine6540         19       0.484319       61.189.243.240       10.129.211.13       TCP       neod2       18067       PSH. ACK       Sequel Acks24 wine6540         20       0.000058       10.129.211.13       61.189.243.240       TCP       neod2       18067       Neod2       PSH. ACK       Sequel Acks26       wine6540         21       0.398523       61.189.243.240       10.129.211.13       10.129.211.13       DNS       Standard guery       A hometown.aol.com         24       0.01193       10.129.211.13       205.188.226.248       TCP       neod2       NS       Standard guery       A hometown.aol.com         24       0.01193       10.129.211.13       10.129.102.0       TCP       optima-vnet > netbios -ssn [SvN] Sequel wine64240       TCP	gter					Espression_ G	nac Apply	
<pre>18 0.000108 10.129.211.13 61.189.243.240 TCP 19 0.484319 61.189.243.240 10.129.211.13 TCP 18067 &gt; neod2 &gt; 18067 [PSH, ACK] Seq=31 Ack=24 win=654 18067 &gt; neod2 [PSH, ACK] Seq=52 Ack=52 win=654 18067 &gt; neod2 [PSH, ACK] Seq=52 Ack=52 win=654 18067 &gt; neod2 &gt; 18067 [PSH, ACK] Seq=52 Ack=52 win=654 18067 &gt; neod2 [PSH, ACK] Seq=70 Ack=70 win=654 23 0.175701 10.129.211.13 10.129.211.13 DNS Standard guery A hometown.a0.com 24 0.0004841 0.129.211.13 10.129.56.6 DNS Standard guery A hometown.a0.com 25 0.000841 10.129.211.13 10.129.102.0 TCP optima-vnet &gt; netbios -ssn [SvN] Seq=0 win=64240 [TCP CHECKSUM 27 0.834424 10.129.211.13 10.129.102.0 TCP optima-vnet &gt; netbios -ssn [SvN] Seq=0 win=64240 [TCP CHECKSUM 29 0.000098 10.129.211.13 10.129.102.2 TCP remote-as &gt; netbios -ssn [SvN] Seq=0 win=64240 [TCP checksum 30 0.000078 10.129.211.13 10.129.102.3 TCP ansysimd &gt; netbios -ssn [SvN] Seq=0 win=64240 [TCP checksum 31 0.000078 10.129.211.13 10.129.102.4 TCP ansysimd &gt; netbios -ssn [SvN] Seq=0 win=64240 [TCP checksum 33 0.000076 10.129.211.13 10.129.102.6 TCP vfo &gt; netbios -ssn [SvN] Seq=0 win=64240 [TCP checksum 33 0.000076 10.129.211.13 10.129.102.6 TCP vfo &gt; netbios -ssn [SvN] Seq=0 win=64240 [TCP checksum 33 0.000076 10.129.211.13 10.129.102.6 TCP vfo &gt; netbios -ssn [SvN] Seq=0 win=64240 [TCP checksum 33 0.000076 10.129.211.13 10.129.102.6 TCP vfo &gt; netbios -ssn [SvN] Seq=0 win=64240 [TCP checksum 33 0.000076 10.129.211.13 10.129.102.6 TCP vfo &gt; netbios -ssn [SvN] Seq=0 win=64240 [TCP checksum 33 0.000076 10.129.211.13 10.129.102.6 TCP vfo &gt; netbios -ssn [SvN</pre>	40. x	Time	Source		Destination	Pretocol	3mlo	
Frame 16 (71 bytes on wire, 71 bytes captured) Ethernet II, Src: DellEsgP_58:93:fa (00:0b:58:93:fa), Dst: Watchgua_04:f8:35 (00:90:7f:04:f8:35 Internet Protocol, Src: 10.129.211.13 (10.129.211.13), Dst: 61.189.243.240 (61.189.243.240) Transmission Control Protocol, Src Port: neod2 (1048), Dst Port: 18067 (18067), Seq: 14, Ack: 1, U	18 19 20 21 22 23 24 25 27 28 29 30 31 32	0.000108 0.484319 0.000058 0.398523 0.184217 0.000841 0.000841 0.0000841 0.834424 0.000098 0.000078 0.000078 0.000078	10.129 61.189 10.129 61.189 10.129 10.129 10.129 10.129 10.129 10.129 10.129 10.129	.211.13 .243.240 .211.13 .243.240 .211.13 .211.13 .211.13 .56.6 .211.13	61.189.243. 10.129.211. 61.189.243. 10.129.211. 61.189.243. 10.129.56.6 10.129.56.6 10.129.211. 205.188.226 10.129.102. 10.129.102. 10.129.102.	40 TCP 3 TCP 40 TCP 3 TCP 13 TCP DNS 240 TCP 5 TCP 1 TCP 1 TCP 1 TCP 5 TCP	<pre>neod2 &gt; 18067 [PSH, ACK] Seq=31 Ack 18067 &gt; neod2 [PSH, ACK] Seq=24 Ack neod2 &gt; 18067 [PSH, ACK] Seq=252 Ack 18067 &gt; neod2 [PSH, ACK] Seq=52 Ack 18067 &gt; neod2 [PSH, ACK] Seq=70 Ack=283 Standard query A hometown.aol.com Standard query A hometown.aol.com Standard query response A 205.188.2 Cma &gt; http [SYN] Seq=0 Win=64240 [T Costination unreschable (Kost unro optima-vnet &gt; netbios_ssn [SYN] Seq-0 ddt &gt; netbios_ssn [SYN] Seq=0 win=6 brvnead &gt; netbios_ssn [SYN] Seq=0 wfro &gt; netbios_ssn [SYN] Seq=0 win=6 vfo &gt; netbios_ssn [SYN] Seq=0 win=6</pre>	=24 wime642 =52 wime641 =80 wime641 =70 wime634 wime63958 [ 226,248 A 20 CCP CHECKSUM that CO =0 wime64240 cme64240 [m wime64240 [m wime64240 [m wime64240 [m wime64240 [m]
Ethernet II, Src: DellEsgP_58:93:fa (00:0b:db:58:93:fa), Dst: Watchgua_04:f8:35 (00:90:7f:04:f8:35 Internet Protocol, Src: 10.129.211.13 (10.129.211.13), Dst: 61.189.243.240 (61.189.243.240) Transmission Control Protocol, Src Port: neod2 (1048), Dst Port: 18067 (18067), Seq: 14, Ack: 1, L			24				and a set in the fraction of the	1310 Fren (
Transmission Control Protocol, Src Port: neod2 (1048), Dst Port: 18067 (18067), Seq: 14, Ack: 1, L	Eth	ernet II.	Sec: De	ellesgP_5	8:93:fa (00:0	b:db:58:9		
Data (17 bytes)								
	Dat	a (17 byt	tes)					

Figure 32 Client does a query for "hometown.com" and gets back a DNS response

We have some signatures as shown in figure 33, we have:

- 1. Port 18067, which is unusual port.
- 2. bbjj.househot.com
- 3. ypgw.wallloan.com
- 4. A number of target IP addresses that were given in the DNS response packets on those targets (figure 34).

	QQ 🖸 📓 🖾 🥵 🎉
ion Cle	ear Apply
otocol	Info
NS	Standard query A bbjj.househot.com
NS	Standard query response CNAME ypgw.wallloan.com
CP	neod1 > 18067 [SYN] Seq=0 Win=04240 [TCF CHERK
CMP	Destination and reachable (Port unreachable)
CP	neod1 > 18067 [SYN] Seq=0 Win=64240 [TCP CHECK:
CMP	Destination unreachable (Port unreachable)
CP	neod1 > 18067 [SYN] Seq=0 Win=64240 [TCP CHECK:
СМР	Destination unreachable (Port unreachable)
NS	Standard query A ypgw.wallloan.com
NS	Standard query response A 61.189.243.240 A 61.1
CP	neod2 > 18067 [SYN] Seq=0 Win=64240 [TCP CHECK:
CP	18067 > neod2 [SYN, ACK] Seq=0 Ack=1 Win=65535
CP	neod2 > 18067 [ACK] Seq=1 Ack=1 Win=64240 [TCP
CP	neod2 > 18067 [PSH, ACK] Seq=1 Ack=1 Win=64240

Figure 33 Signatures that indicate the abnormal activity

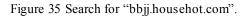
			precision_ C	we when		
40 Time	Source	Destination	Protocol	lefe .		
1 0.000000	10.129.211.13	10.129.56.6	DNS	Standard query	A bbjj.househot.com	
2 0.237997	10.129.56.6	10.129.211.13	DNS	Standard query	nesponse CNAME vpow.wallloan.	CO
3 0.001861	10.129.211.13	210.234.235.105	TCP	neod1 > 18067	[SYN] Seq=0 Win=64240 [TCP CHE	CK
5 2 000536	10,129,211,12	216 234 235 165	TCP	neod1 > 18067	[SYN] Segu0 Winu64240 [TCP CHE	000
D D DODDAY			I COMP		reachable (Port unreachable)	
7 5.933724	10.129.211.13	216.234.235.165	TCP	neod1 > 18067	[SYN] Seg=0 Win=64240 [TCP CHE	CK
8.0.000710	216, 234, 235, 16	10.129.211.13	ICMP		reachable (Port unreachable)	
9 328.35307	3 10.129.211.13	10.129.56.6	DNS		A ypgw.wallloan.com	
10 0.228953	10.129.56.6	10.129.211.13	DNS	Standard query	response A 61.189.243.240 A 6	11
Additional	RRs: 3					
Queries						
Answers						
		NAME, class IN,			2m	
		, class IN, addr				
		, class IN, addr				
		, class IN, at tr				
		, class IN, addr				
		, class IN, addr				
		, class IN, addr				
		, class IN, addr				
		, class IN, addr				
		, class IN, addr				
sypgw.wall	loan.com: type A	, class IN, addr , class IN, addr				

Figure 34 List of IP addresses that came in DNS response

Be careful connecting to those targets because those targets can infect other systems in case they are not protected. Let us now go to the browser and just find out what this client might be infected with.

We have used the browser, Mozilla Firefox and will make a search for "bbjj.househot.com" as shown in figure 35.

	Ubuntu Start Page						÷
	👉 🔲 about:startpa	ige		s 🗸 🗸	🌮 🚰 🔻 Google	Q	
		ubuntu®					
		Google					
		bbjj.househot		c			
U			Ubuntu shop >		aupibys		
704		obuild help			indirity /		
704							
1							



This seems to tell us the definition of "bbjj.househot.com" listed as the Window 32 Mocbot. It is also called SDbot Worm and IRC-Mocbot, as it has different names as shown in figure 36.

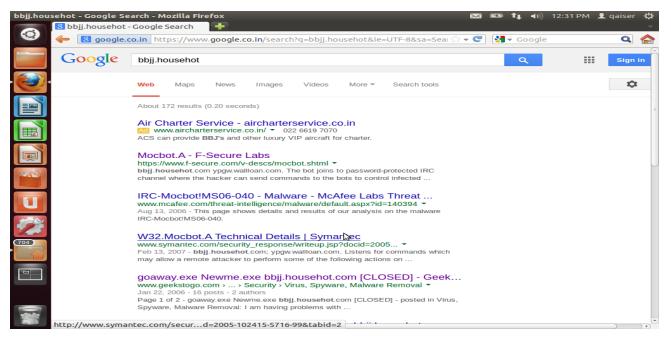


Figure 36 Shows the result for "bbjj.househot.com"

We have used Wireshark and build a filter that will show us when those DNS queries come back and they look a little suspicious.Look at the second packet where we have the Answer Resource Record, "12" answers coming in the record. As already mentioned that answers more than 4 or 5 is not usual because that is so constantly happening in the environment of bot infected host (figure 37).

Next we built a "Butt-Ugly" color filter that will highlight any packet that will have Answer Resource Record value greater than 5 let us say. When we highlight the field inside a packet down below on the status-bar, Wireshark tells us the name of the field is "dns.count.answers".

iick-client.pcap-Wires Eile Edit View Go S	Capture Analyze Statistics	-		2 Q Q 🗹 🛛 🖉 1
Filter: dns.count.answers	== 12	• Eg	pression C	lear Apply
No Time	Source	Destination	Protocol	Info
1 0.000000	10.129.211.13	10.129.56.6	DNS	Standard quer
2 0.237997	10.129.56.6	10.129.211.13	DNS	Standard quer
3 0.001861	10.129.211.13	216.234.235.16	5 TCP	neod1 > 18067
4 0.000549	216.234.235.16	5 10.129.211.13	ICMP	Destination u
5 2.999536	10.129.211.13	216.234.235.16	5 TCP	neod1 > 18067
6 0.000633	216.234.235.16	$5\ 10.129.211.13$	ICMP	Destination u
7 5.933724	10.129.211.13	216.234.235.16	5 TCP	neod1 > 18067
8 0.000710	216.234.235.16	$5\ 10.129.211.13$	ICMP	Destination u
9 328.35307	3 10.129.211.13	10.129.56.6	DNS	Standard quer
10 0.228953	10.129.56.6	10.129.211.13	DNS	Standard quer
11 0 006457	10 120 211 12	61 190 243 240	TCD	nood? > 18067
Transaction	37997000 seconds 1D: 0x0006 580 (Standard qu 1 12	] ery response, No	error)	

Figure 37 Filter is used to get DNS responses having Answers Resource Records greater than 12. We did Right Click on this field and prepare a filter based on the selected value as shown in figure 38.

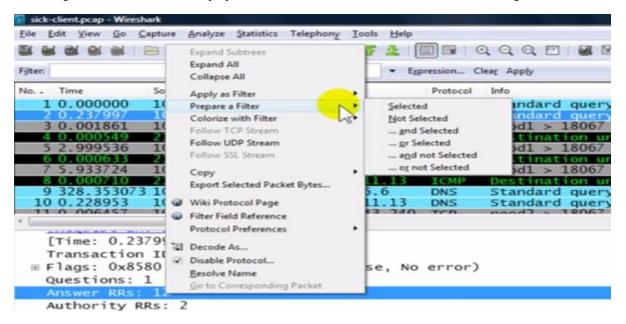


Figure 38 Prepare Filter based on selected value

We made changes in the filter. We wrote "dns.count.answers >5" or "dns.count.answers gt 5". We got two packets having answers greater than 5 as shown in figure 39.

let descount.	answers gt.5		Expression Clear Apply				
Time	Source	Destination	Protocol Info		-		
10 337.5	19039 10.129.56	.6 10.129.211.1 .6 10.129.211.1	DNS Stand	and query n	esponse A 1.	189.243.240	A 61.
			and a strength of the strength				
	-						0
	0.237997000 se	conds]					2
[Time: Transa	0.237997000 se ction ID: 0x000	6					
[Time: Transa Flags:	0.237997000 section ID: 0x0000 0x8580 (Standa		No error)				0
[Time: Transa Flags: Questio	0.237997000 section ID: 0x0000 0x8580 (Standa ons: 1	6	No error)				2
[Time: Transa Flags: Questic Answer	0.237997000 section ID: 0x000 0x8580 (Standa ons: 1 RRs: 12	6	No error)				
[Time: Transa Flags: Questic Answer Author	0.237997000 section ID: 0x000 0x8580 (Standar ons: 1 RRs: 12 ity RRs: 2	6	No error)				2
[Time: Transa Flags: Questic Answer Author Additic	0.237997000 section ID: 0x000 0x8580 (Standar ons: 1 RRs: 12 ity RRs: 2 onal RRs: 3	6	No error)				
[Time: Transa Flags: Questic Answer Author Additic	0.237997000 section ID: 0x000 0x8580 (Standar ons: 1 RRs: 12 ity RRs: 2 onal RRs: 3 s	6	No error)				
[Time: Transa Flags: Questi Answer Author Additi Queries Answer	0.237997000 section ID: 0x0000 0x8580 (Standa ons: 1 RRs: 12 ity RRs: 2 onal RRs: 3 s	6 rd query response, 1					
[Time: Transau Flags: Questi Answer Author Additi Querie Answer bbjj.	0.237997000 se ction ID: 0x000 0x8580 (Standa ons: 1 RRs: 12 ity RRs: 2 onal RRs: 3 s s househot.com: t	6 nd query response, 1 cype CNAME, class IN	, cname ypgw.wa				
[Time: Transau Flags: Questi Answer Author Additi Querie Answer = bbjj. ypgw.	0.237997000 se ction ID: 0x0000 0x8580 (Standa ons: 1 RRs: 12 ity RRs: 2 onal RRs: 3 s househot.com: t wallloan.com: t	6 rd query response, 1 cype CNAME, class IN cype A, class IN, ac	, спате урдw.wa dr 216.234.235.	165			
[Time: Transau Flags: Questic Answer Additic Querie Answer bbjj. ypgw.	0.237997000 sec ction ID: 0x000 0x8580 (Standa ons: 1 RRs: 2 ity RRs: 2 onal RRs: 3 s househot.com: t wallloan.com: t	6 nd query response, 1 cype CNAME, class IM cype A, class IN, ac cype A, class IN, ac	, cname ypgw.wa dr 216.234.235 dr 151.198.6.55	165			1
[Time: Transam Flags: Questi Antswer Author Additi Querie Answer bbjj. ypgw. ypgw.	0.237997000 se ction ID: 0x000 0x8580 (Standa ons: 1 RRs: 2 ity RRs: 2 onal RRs: 3 s s househot.com: t wallloan.com: t wallloan.com: t	6 rd query response, 1 cype CNAME, class IN cype A, class IN, ac cype A, class IN, ac	, cname ypgw.wa dr 216.234.235 dr 151.198.6.53 dr 216.234.247.	165 191			
[Time: Transam Flags: Questin Answer Additi Queries Answer: bbjj. ypgw. ypgw. ypgw. ypgw.	0.237997000 sec ction ID: 0x000 0x8580 (Standar ms: 1 RRs: 2 ity RRs: 2 onal RRs: 3 s househot.com: t wallloan.com: t wallloan.com: t	6 nd query response, 1 cype CNAME, class IM cype A, class IN, ac cype A, class IN, ac	, cname ypgw.wa dr 216.234.235 dr 151.198.6.55 dr 216.234.247. dr 68.112.229.2	165 191 28			

Figure 39 Filter is used to get DNS responses having Answers Resource Records greater than 5

After that we went to the coloring rules area and made a new color by writing (figure 40):

Filter						-
Name	DNS response gt 5				(ip-checksum-bod	UP
String	dns.count.answers gt 5			Expression		1
	Colors	Background Color	Status Disable	ed		
	15		QK	Gancel	ap    ismp	Move selected filte up or down
Manage	ILPOTTOTIN	scp.nage or un	ent il schunder	THE R. L.		op er denn
Import	TCP	tcp				
(Imported)	UDP	udp				
Export	Broadcast	eth(0) & 1				Down
Clear	name	filter				*
Zien	« [	100			•	
Help				Q	K Apply	Gencel
-	-					
Author	ity RRs: 0					

Figure 40 Select the color for foreground area.

Name = dns.count.answers gt 5 and in string area we wrote: Filter = dns.count.answer > 5. We also selected orange as foreground color and green as background color (figure 41, 42).

Wireshark: Edit Color Filter - Pro				22			Order
Manti Exp Bxp Help	Hue: Saturation: Yalue: Color game:	27 0 85 0 98 0 98 0 98 0 98 0	Bed: Green: Blue:	250 132 37		mp E	Move selected filts up or down
E Eleip		-	<u>~</u>		QK	, Деречу	

Figure 41 Orange is selected as foreground color

Filter	DNS response gt 5				-	-
	dns.count.answers gt	5		Expression	lipsheeksum_lia	L LIP
Display	Colors reground Color	Background Color	Status Disablec			
		15	QK	Gancel	np    ismp	Move selected filte up or down
Manage	TEP STREET	scprings or w	ant gropiningen			
Import	TCP	tcp				
(Burdenstein)	UDP	udp				
Export	Broadcast	eth[0] & 1				Down
Clear	name	filter				· ·
( and )	« [					
Help				9	K Apply	Cancel
	-					
Authori	ity RRs: 0					

Figure 42 Select the color for background area

The figure 43 below shows the Edit color filter of the Wireshark. The Name field contains the name of the filter which has orange foreground color and green background color.

Filter	r			$\sim$		
Nam	ne:	DNS re	spons	e gi 5		
Strin	ng: d	Ins.co	unt.an	swers g	it 5	
Disp	lay C	olors				
1	Fore	groun	nd Col	or	[	Backgroun

Figure 43 Edit Color Filter shows the colored foreground and background area

After applying the butt-ugly filter, there is no way we can miss these butt-ugly packets as shown in figure 44.

ptert		· Egress	on Clear, Apply
lo Time 1 0.000000	Source 10.129.211.13		Meed Me S Standard guery A bbjj.househot.com
3 0.001861	10.129.211.13	216.234.235.165 T	
4 0.000549 5 2.999536 5 0.000534	10.129.211.13	216.234.235.165 T	DMP Destination unreschable (Port unreachable) CP needl > 18067 [S.v] Seq=0 win=64240 [TCP CHECK Destination unreachable (Port unreachable)
7 5.933724	10.129.211.13	216.234.235.165 T	
9 328, 35307	3 10.129.211.13		Standard query A ypgw.wallloan.com
11 0.006457 12 0.396606 13 0.000185 14 0.000095 15 0.559178 16 0.000050 17 0.402661 18 0.000108	$\begin{array}{c} 10 & 129 & 211 & 13 \\ 61 & 189 & 243 & 240 \\ 10 & 129 & 211 & 13 \\ 10 & 129 & 211 & 13 \\ 61 & 189 & 243 & 240 \\ 10 & 129 & 211 & 13 \\ 61 & 189 & 243 & 240 \\ 10 & 129 & 211 & 13 \\ \end{array}$	10.129.211.13 T 61.189.243.240 T 10.129.211.13 T 61.189.243.240 T 10.129.211.13 T 10.129.211.13 T	CP         neod2 > 18067         [SYN] Seq=0         win=64240         [TCP CHECK           CP         18067         > neod2         [SYN, ACK] Seq=0         Ack=1         win=65533           CP         062         > 18067         (ACK) Seq=1         Ack=1         win=64240         [TCP           CP         neod2         > 18067         (PSH, ACK] Seq=1         Ack=1         win=64240         [TCP           CP         neod2         > 18067         (PSH, ACK] Seq=1         Ack=1         win=64240         [TCP           CP         neod2         > 18067         (PSH, ACK] Seq=1         Ack=1         win=64240         [TCP           CP         neod2         > 18067         (PSH, ACK] Seq=1         Ack=1         win=64240         [TCP           CP         neod2         > 18067         (PSH, ACK] Seq=1         Ack=1         win=6526         [TCP           CP         neod2         > 18067         (PSH, ACK) Seq=1         Ack=1         win=64240           CP         neod2         > 18067         (PSH, ACK) Seq=1         Ack=1         win=64240           CP         neod2         > 18067         (PSH, ACK) Seq=31         Ack=24         win=64240           CP         neod2
10 0.434310 20 0.000058 21 0.398523 22 0.184217 23 0.175701 24 0.001193 25 0.000841 40 0.0100510 27 0.834424 28 0.000098 29 0.000101	61, 189, 241, 240 10, 129, 211, 13 61, 189, 243, 240 10, 129, 211, 13 10, 129, 211, 13	10.129.211.13 T 61.189.243.240 T 10.129.56.6 D 10.129.211.13 D 205.188.226.248 T 10.129.102.0 T 10.129.102.0 T	P 1800/ > neod/ [541, Acc ] Segred Acked Wine641 P neod2 > 18067 [PSH, ACK] Segred Acked Wine641 P 18067 > neod2 [PSH, ACK] Segred Acked Wine642 P neod2 > 18067 [ACK] Segred Acked Wine64263 Standard Query A hometown.aol.com Standard Query Response A 205.188, 226.248 A 20 P cma > http [57N] Segred Wine64240 [TCP CHECKSUM Contents for manufacture and the segred Wine64240 P of tima-vnet > netbios-ssn [57N] Segred Wine64240 ddt > netbios-ssn [57N] Segred Wine64240 [TCP CHECKSUM ddt > netbios-ssn [57N] Segred Wine64240

Figure 44 Results after applying the Butt-Ugly Filter

As we analyze botnet effected system we see that there is a norder to detect the botnet, we need to follow an effective similarities in the packets they request, the replies that comeway so that we can detect the bots as early as possible. We and also the data of the configuration file downloaded by thehave designed a generic Architecture for effectively detecting bot program used to launch the attacks. Thus we need to stop the bots by monitoring the network traffic over the internet. our system from becoming a bot in a botnet. This can be done The internet is widely used by people all over the world, in two steps. The applications used on

internet can be many like LinkedIn, Google +, Skype, Analyzing the traffic: This is done by seeing the DNS replies Instagram, Twitter, Facebook, YouTube and much more. All and if the answer field has more than few entities then we can these applications will provide a number of benefits but only if just discard and quarantine such packet till the user or system-they are used in a responsible way. At the same time the administrator looks into the contents of the DNS request and attackers are also present on the internet to perform the illegal reply and decide if they are genuine or generated by the activities. All the activities going on the internet will generate malicious program (botnet) that might have infiltrated ourthe network traffic. The incoming and outgoing network system. Discarding such packets will stop the bot programtraffic is first sent to the network traffic assembler containing running on our computer from communicating with the C&Cthe repository where the network packets are stored for the server making it unable to download the configuration file and future use. There are number of tools used for assembling the thus stop the bot from performing the attack.

Analyzer) for capturing the network flow. The captured *Machine Based learning system*: This technique is based on a packets are then passed through the Filter that helps in filtering program, which needs to be trained by using a reducing the traffic burden. There are two methods commonly training set, comprising of the similarities in a botnet used for filtering the network flow, they are White Page and communication steps or the file downloaded. If any of the Black Page filtering techniques. The legitimate packets like communication steps or file downloaded matches the filter of antivirus updates are filtered by White page filtering technique the filter program it quarantines it and thus stopping the bot to and the malicious packets like viruses, Trojans are filtered by perform its attack. Black Paper filtering technique

4.2 Generic Architecture for detecting the Botnet from the network traffic

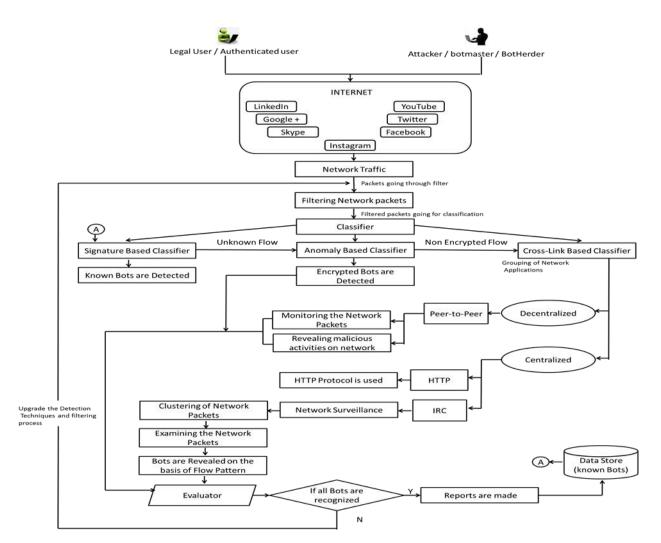


Figure 45 Generic Architecture for detection of botnet from the network traffic

The filtered network flow is passed through the Classifier distributed among the multiple servers or there is no obvious Three types of Classifiers have been used namely Signaturenaster-slave relationship between C&C server and bots. The Based Classifier, Anomaly Based Classifier and Cross Lin P2P traffic is monitored by using the Traffic monitoring Based Classifier (figure 45). The known bots are detected by nodule, in order to discover the group of hosts having same Signature Based Classifier. It helps in minimizing the fals behavior and communication pattern. The possible malicious positive rate as this technique only detects the known bots. The ctivities that are related to the P2P based packets are detected rest of the network traffic is left with unknown flow, which iby the malicious activity detector. The IRC and HTTP network passed through the Anomaly Based Classifier. It detects the traffic is a type of Centralized applications (having single C&C encrypted bots only, leaving behind the non-encrypted networ Server).

traffic. The encrypted traffic detected is then passed through the

Evaluator or the Analyzer. The non-encrypted network flow is the Centralized network traffic (IRC) is sent for the passed through the Cross-Link Based Classifier. It classifies the urveillance or monitoring. The monitored network traffic is non-encrypted network flow into the different network hen clustered or grouped and then examined. After the close applications.

the basis of flow pattern and are passed through the network We have grouped the network flow into the two applicationspacket evaluator, which analyzes the unknown packets so that i.e.; Centralized and Decentralized applications. The P2P (Peemo information is lost. If all the bots are discovered, the reports to-Peer) network traffic is a type of decentralized applicationare generated and are updated into the data store. Else, the where no single unit is accountable for providing or issuin Detection techniques and the Filtering process are upgraded C&C (Command and Control) to bots. Here the bots are eitheand the filtering of the network flow is restarted.

### V. RESEARCH CHALLENGES VI. CONCLUSION

5.1 Detection: Detecting the botnet in a system or the networBotnet is a very distinctive technology used by attacker which is a major task. A botnet is considered to be a group of thes very extensive in nature, thus due to this, the botnet research compromised systems also known as zombies, which are unders still in inception. The botnet discriminates itself from other the control and command of the single botmaster. These botmalware in the ability of its compromised machines to establish keep on forming again and again with the help of the different ommand and control with remote server controlled by human types of the network architecture and various applications and missfeasor. Every stage of the life cycle of botnet must be using topologies and the digital signatures also [134]. The construction of just one stage is interrupted, it will attacks from the botnets and also a Honeypot is used to detect ender the whole botnet detection. This paper surveys state-of-any malicious program and mitigating the attacks. But if thereart botnet research that can be categorized into the areas is a continuous attack going on, then detecting the botnet withmamely, (1) Botnet review and sum up. (2) Botnet revelation the help of these systems will be difficult. So it requires somand botnet revelation techniques. (3) Classification of botnets advanced techniques or systems.

botnet detection techniques have been discussed, among them 5.2 Botnet size: The size of the botnet depends on the numbeonly Signature- based technique is the only one that can't detect of the bots attacking a system. Generally, the size of the botnet which ased on DNS and Data mining can detect real –world botnets expands greatly and moreover, there are various botnets which ased on DNS and Data mining can detect real –world botnets consists of the million bots which can be used to launch largregardless of the botnet protocol and structure with a very low and powerful attack. For example, botnet Zeus has more that alse positive rate. Only Mining-based botnets have the million of bots and botnet Waladac have the strength of sending apability to detect the encrypted botnet. Data mining and 1.5 billion spams per day. Therefore the size of the botnet is machine learning techniques are well suited on flow major challenge [135].

information from bots to interpret their behavior and revelation

5.3 AnalysisS: As the botnets are both reactive and proactive in mechanism. However, a large number of challenges still persist nature therefore, analysis can be done in both the active as well the area of Botnet Detection. as the passive mode. An example of the active analysis is the

honeypot, but due to its difficult setup its use is restricted for number of research works have been done for P2P and IRC the large scale networks. And the passive analysis is performed otnets, but the motivations for using the HTTP protocol are on the network data traffic collected and can also identify many nultiple. For IRC –based botnets, the problem is that we can't botnets at a time but it is limited to some specific types of the source code of the most of the bots. The main issues botnets only. related to P2P botnets are – hiding the botnet topology while

*5.4 Investigation:* For detecting the botnet attack and collecting atterns more often and making it harder for detection. the data about the botnet, various types of the detection betecting the compromised hosts in the botnet will continue to techniques are used to perform an investigation. To present out a challenging task. Anomaly detection is a feasible approach evidence and fulfill our criteria in a court of law, after detecting botnets. The interesting issue about this approach acknowledgement of the attack is being used to precede these time efficiency. If the attack occurs and we can capture the investigation process and thus generating the required result anomaly in the first place and fix the relevant problems before Thus, investigating a botnet is also a one of the majoft is used for performing the abnormal activities, we say challenge.

5.5 Server failure: It is one of the biggest challenges while

detecting the botnets. If the server failed during the process of the botnets are turning to cloud computing to expand their while collecting the packets or required information, then it isotentials. The cloud platform is used by the botnets in two possible that all the data captured or detected is lost anyway – host the C&C server on the cloud or create bots on the and then there will be no proof. Server failure can relate to the loud instead of infecting user machine. The cloud security is DNS failures or the failures related to the name servers [137]. still in a transient stage and most of the existing detection

5.6 Cryptography: One of the important parts of the botnet is to nice cover to botnets for carrying out their malicious activities. maintain the integrity and authentication of the system or the mobile phones can utilize a number of communications entire network, which can be violated by an attacker through fike 3G, 4G which multiplies the possibilities for C&C and any means. Thus in keeping it all confidential throughout the malware propagation.

## REFFERENCES

- Feily, M., A. Shahrestani, et al. (2009). A survey of botnet and botnet detection. Emerging Security Information, Systems and Technologies, 2009. SECURWARE'09. Third International Conference on, IEEE.
- [2] B. Saha and A, Gairola, "Botnet: An overview," CERT-In White PaperCIWP-2005-05, 2005.
- [3] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in Proc. 6th ACM SIGCOMM Conference on Internet Measurement (IMC'06), 2006, pp.41–52
- [4] Soltani, S., S. A. H. Seno, et al. (2014). "A survey on real world botnets and detection mechanisms." International Journal of Information and Network Security (IJINS) 3(2).
- [5] Blunden B. "The Rootkit Arsenal Escape and Evasion in the Dark Corners of the System": Wordware Publishing, Inc.; 2009.
- [6] "GMER- Rootkit Detector and Remover". Available from: http://www.gmer.net/.
- [7] D. Dagon, G. Gu, C.P. Lee, W. Lee, "A Taxonomy of Botnet Structures," in Proc. 23rd Annual Computer Security Applications Conference (ACSAC 2007), 2007, pp. 325-339.
- [8] Moon YH, Kim E, Hur SM, Kim HK. Detection of botnets before activation: an enhanced honeypot system for intentional infection and behavioral observation of malware. Security and Communication Networks 2012, DOI:10.1002/sec.431, Available from http://dx.doi.org/10.1002/sec.431 [Accessed on 9 May 2013].
- [9] Jaikumar P, Kak AC. A graph-theoretic framework for isolating botnets in a network. Security and Communication Networks 2012; 5: 1939–0122, DOI:10.1002/sec.500, Available from:

http://dx.doi. org/10.1002/sec.500 [Accessed on 9 May 2013].

- [10] Zhu, Z., Lu, G., Chen, Y., Roberts, P. and Han, K., "Botnet Research Survey", 32nd Annual IEEE International Conference on Computer Software and Applications, 2008, pp. 967-972
- [11] Bailey M, Cooke E, Jahanian F, Xu Y, Karir M. A survey of botnet technology and defenses. In Cybersecurity Applications & Technology Conference For Homeland Security. IEEE: California, 2009; 299–304. Available from: http://ieeexplore.ieee.org/xpl/articleDetail s.jsp?arnumber=480445 [Accessed on 9 May 2013].
- [12] Stinson E, Mitchell J. Towards systematic evaluation of the evadability of bot/botnet detection methods. In Proceedings of the 2nd Conference on USENIX Workshop on Offensive Technologies
- [13] Liu J, Xiao Y, Ghaboosi K, Deng H, Zhang J. Botnet: classification, attacks, detection, tracing, and preventive measures, 2009; 1–12. Available from: http://mts.hindawi.com/utils/getacceptedm sfile.aspx? msid=692654&vnum=2&ftype=manuscri pt [Accessed on 9 May 2013].
- [14] Lim SY, Jones A. Network anomaly detection system: the state of art of network behaviour analysis, International Conference on Convergence and Hybrid Information Technology, 2008. (ICHIT '08), Daejeon, Korea, 2008; 459–465
- [15] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: Understanding, detecting, and disrupting botnets," in *Proc. Steps to Reducing Unwanted Traffic* on the Internet Workshop (SRUTI'05), 2005, pp. 39-44.
- [16] C. C. Zou and R. Cunningham, "Honeypot-aware advanced botnet construction and maintenance," in *Proceedings of the International Conference on Dependable Systems and*

Networks (DSN '06), pp. 199–208, Philadelphia, Pa, USA, June 2006.

- [17] F. Freiling, T. Holz, and G. Wicherski, "Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks," in Proc. 10th European Symposium on Research in Computer Security (ESORICS), vol. Lecture Notes in Computer Science 3676, September 2005, pp. 319–335.
- [18] P. Baecher, M. Koetter, T. Holz, M. Dornseif., and F. Freiling, "The nepenthes platform: An efficient approach to collect malware," in Proceedings of International Symposium on Recent Advances in Intrusion Detection (RAID'06), (Hamburg), September 2006
- [19] Zeidanloo, H.R, M.J.Z Shooshtari et al. A Taxonomy of Botnet Detection Techniques. Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on,IEEE.
- [20] S. Racine, Analysis of internet relay chat usage by DDoS zombies, M.S. thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, April 2004
- [21] Y. Kugisaki, Y. Kasahara, Y. Hori, and K. Sakurai, "Bot detection based on traffic analysis," in Proceedings of the International Conference on Intelligent Pervasive Computing (IPC '07), pp. 303– 306, Jeju Island, South Korea, October 2007
- [22] P. Sroufe, S. Phithakkitnukoon, R. Dantu, and J. Cangussu, "Email shape analysis for spam botnet detection," in Proceedings of the 6th IEEE Consumer Communications and Networking Conference (CCNC'09), pp. 1–2, Las Vegas, Nev, USA, January 2009
- [23] T. Holz, M. Steiner, F. Dahl, E. W. Biersack, and F. Freiling, "Measurement and mitigation of peer-to-peer-based botnets: a case study on storm worm," in Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, pp. 1–9, San Francisco, Calif, USA, April 2008.

- [24] P. Wang, S. Sparks, and C. C. Zou, "An advanced hybrid peer-to-peer botnet," in Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets, p. 2, Cambridge, Mass, USA, July 2008.
- [25] R. Lemos, "Bot software looks to improve peerage," http://www.securityfocus.com/news/1139 0.
- [26] H. Choi, H. Lee, H. Lee, and H. Kim, "Botnet Detection by Monitoring Group Activities in DNS Traffic," in Proc. 7th IEEE International Conference on Computer and Information Technology (CIT 2007), 2007, pp.715-720
- [27] R.Villamarin-Salomon and J.C. Brustoloni, "Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic," in Proc. 5th IEEE Consumer Communications and Networking Conference (CCNC 2008), 2008, pp. 476-481
- [28] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol- and structure independent botnet detection," in Proc. 17th USENIX Security Symposium, 2008.
- [29] G. Gu, J. Zhang, and W. Lee, "Botsniffer: Detecting botnet command and control channels in network traffic," in Proc. 15th Annual Network and distributed System Security Symposium (NDSS'08), 2008
- [30] Robiah Y, Siti Rahayu S., Mohd Zaki M., Shahrin S., Faizal M. A., Marliza R.; A New Generic Taxonomy on Hybrid Malware Detection Technique. (IJCSIS) International Journal of Computer Science and Information Security, Vol. 5, No. 1, 2009
- [31] Daan, A.F. Shosha, and P. Gladyshev, BREDOLAB: shopping in the cybercrime underworld, in Digital Forensics and Cyber Crime. 2013, Springer. p. 302-313.
- [32] Eschweiler, S. and E. Gerhards-Padilla, Towards Sound Forensic Acquisition of Volatile Data, in Future Security. 2012, Springer. p. 289-298.

[33] Peng, T., C. Leckie, and K. Ramamohanarao, Survey of networkbased defense mechanisms countering the DoS and DDoS problems. ACM Computing Surveys (CSUR), 2007.