RESEARCH ARTICLE OPEN ACCESS

Generic Architecture for Detecting Botnet

Anushah Khan [1], Anchit Bijalwan [2]
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 typ es 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

Botnets are emerging threat with hundreds of millions of computers infected. Botnets have become a severe global Internet threat. A "Botnet" consists of a network of unprotected computers controlled by an attacker ("Botmaster"). It is a collection of software robots, or bots, which run automatically. They run on groups of zombie 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 (DDoS) attack, spam campaigns, and phishing activity. The size of the Botnet may differ from tens and hundreds to few thousands. Most of the times, the host machine does not know that it is compromised [[1],[2],[3]]. In fact, the system which we are using can also be a part of Botnet. The attacker first exploits the unprotected system by usually Trojans and once the system gets infected, it comes under the control of the Botmaster. The Command and Control(C&C) Server is used for sending command to the bots. The C&C server connects the Botmaster with the Bots. Botnet may have none, one or many C&C Servers. The C&C Server receives the commands from the Botmaster, forwards them to the botnet and then sends the reports back to the Botmaster. Botnets are used to perform DDOS attacks against the number of targets including government and even other botnets. It is possible to re-program or update the botnet node software after it has 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 installed malware called "rootkit" is activated each time a system boots up. The rootkits are not easy to detect because they are activated before the Operating System of any system has completely booted up [[4], [5], [6]]. The Botnet Life Cycle consists of five phases .Figure 1 below shows the life cycle of the botnet.

In the first phase, the Botmaster, which is the attcaker expoits the vulnerable system by sending malicious progarms to it like Trojans and therefore, this phase is known as preliminary infection phase.This gives back door entry to the BotHearer. In the second phase, the infected system downloads and installs the bot binary into itself. Once the bot program is installed in the exploited system, it starts behaving like a Bot and therefore is known as Secondary injection phase. In the third phase, the bot send query to the DNS server in order to get the address of the C&C Server. The moment the bot gets the address, it joins to the C&C Server and authenticates itself to it.The C&C Connection is made by the bot program that was installed in the victim system which has now become a bot. Once the C&C connection is established, the newly made bot becomes the part of the botmaster's botnet army and is now ready to act according to the commands that it receives from the C&C Server[[6] , [7]]

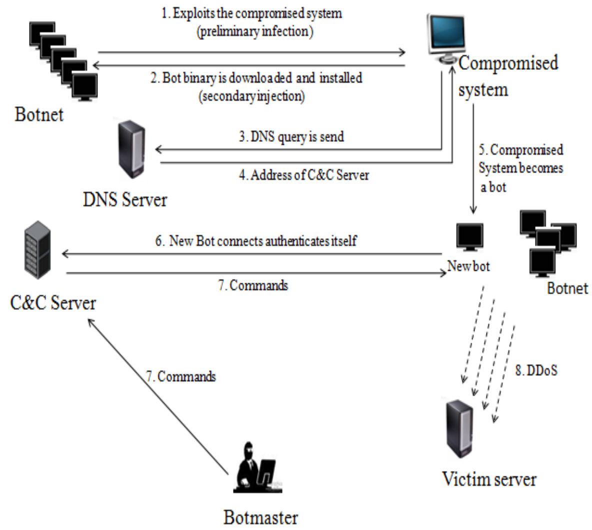

Figure1: Botnet Life Cycle

. 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

Large number of work has been done on the detection of the botnets. The detection techniques mostly used by the researchers include Signature based, Anomaly based, Network based, Host based and Data mining based techniques. In the earlier days, Signature based techniques were used for detecting the botnets but it quickly lost importance when it could not find the unknown bots. A number of passive techniques like honeypots, analysis of flow records, and analysis of spam records, packet inspection, and analysis of application log files, DNS-based approaches, and evaluation of anti-virus software feedback are examined. Active detection techniques like infiltration, detecting fast-flux networks, DNS cache snooping, sinkholing, IRC-based botnets detection and P2P botnets detection are examined. Various botnet mitigation schemes are illustrated too. The survey [8] offers botnets history, components of a botnet, characteristics of a bot, life 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 [11]. Al-Ahmad et al. [29] used a Sniffer program that performed monitoring function. All the message that are exchanged between the bots and the botmaster ,the IP header of TCP were captured and then discrimination was made between the legal and illegal activities by using statistical chart. Garcia et al. [59] used the EM Clustering algorithm for the detection of synchronization in bots and for the detection of the behaviour of the botnets . The EM algorithm is used for the clustering of the time slices that have been divided while seeking to the detection of synchronization Jianbo et al. [65] proposed an algorithm based on the analysis of flow. After the preprocessing of flow grasped from layer 3 switches, it gets three vectors, such as source IP, destination IP and package size, then defines reasonable sliding window of time, does dynamic analysis based on the algorithm of connection rate. Steinberger et al. [61] used different techniques for the detection of anomaly and for mitigating the botnets at the internet scale. Xiang et al. [60] provided a new mitigation technique that promoted the development of more efficient countermeasures against advanced botnets. Zhao et al. [9] presented a system for the detection of botnet activity in both the command and control and attack phase. The botnet detection techniques can be categorized as follows: Honeypot and Honeynet, IRC-based detection , and others like IDS (Intrusion Detection System),Firewall etc. Figure 2 shows the pictorial representation of the botnet detection techniques.

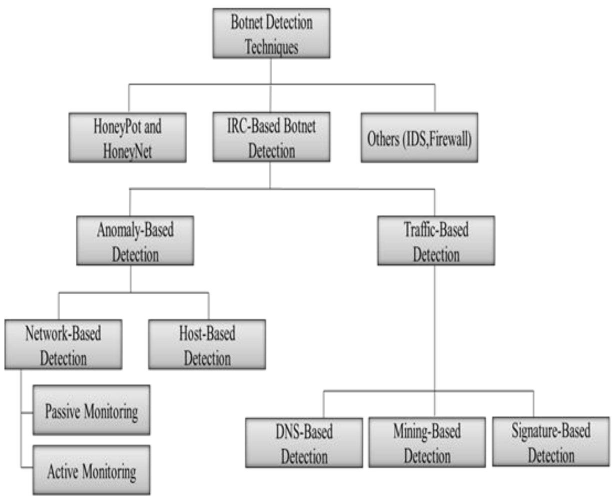

Figure 2: Botnet Detection Techniques

3.1 Honeypot and Honeynet

The first and the most general approach for detecting and tracing the botnets is the use of honeypots, where a subset pretends to be compromised by a Trojan, but actually observing the behavior of attackers, enables the controlling hosts to be identified[15]. Bethencourt et al. have successfully identified honeypots by using intelligent probing according to public report statistics. Honeypot and active responders are used to collect bot binaries. Then, pretend to join the botnet as a compromised machine by running bots on the honeypots and permitting them to access the IRC Server. In [16], Zou and Cunningham have proposed another methodology for honeypot detctio based on independent software and hardware .The useful information gathered by the honeypot is: Signature of bots for content-based detection, information of botnet C&C mechanism/Servers, unknown security holes that enable the bots to penetrate the network, tools and techniques that are used by the attack and finally the motivation of the attacker. In [17], the author has used has used honeypot to track and generate botnets in the network and generate an early report for understanding the consequences of botnets. Nepenthe [18] is the example of low interaction honeypot that simulate some vulnerability and provides some features for the collection of malware binaries [19]. The drawback of this technique is that the limited scale of exploited activities can be tracked. It can only give report for infection machines that are anticipated and put in the network as trap system. It can't give a report for those computers that are infected with bot in the network [19]. 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 categorized into: Detection based on traffic Analysis and Detection based on Anomaly Activities.

3.2.1 Detection based on Traffic/Flow Analysis

The main objective is to extract feature information on the packets from the traffic and match pattern registered in the knowledge base of existing bots. Although it is easy to carry on by simply comparing every byte in the packet, but it has several demerits [21]. It should always update the knowledge base with new signatures. Before the knowledge bases are patched, the new bots may launch attacks. In [22], Sroufe et al, proposed a different method for detecting the botnets. Their method can effectively and automatically identify the spam or bots.The main idea is to extract the shape of email by applying the Gaussian Kernel density estimator [22]. In [[23], [24], [25]], flow/traffic analysis is used to detect the attacks from botnet. It can be divided into the four steps: Packet monitoring phase, Data preprocessing phase, revealing phase, and Analysis phase, the diagram of which is shown below in Figure 3.

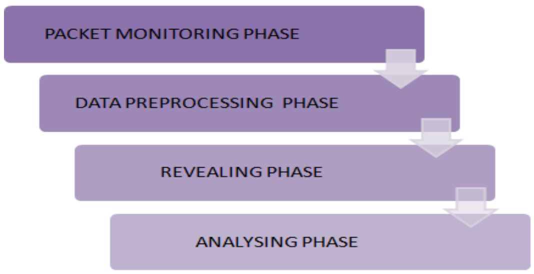

Figure 3: The four steps of Traffic Analysis Botnet Detection Technique

In Packet Monitoring Phase, packet sniffer is used to monitor the packets. In Data Preprocessing Phase, the data is recalculated in the form so that they can be used to detect the attacks. In revealing Phase, the normal data and abnormal data are distinguished. In the Analysis Phase, the performance of the packets is evaluated. The Traffic /Flow Analysis detection technique can be categorized into the following:

3.2.1.1 Signature Based Detection – This technique maintains a database of known bots or attacks and compares the characteristics of netwok traffic with the known bots present in the database. This technique is considered as an efficient technique for detecting known bots. The bots are detected quickly with almost zero false positive rates and needs less system resources. The major drawback of this technique is that it can't be used for detecting the unknown bots.For example Snort, which is an Intrusion Detection System, monitors network traffic to find signature of existing bots.

3.2.1.2 DNS Based Detection – This detection technique is based on the particular DNS information that is shared by the botnet and C&C. These are similar to anomaly detection techniques. Bots typically initiate connection with C&C Server to get commands. For accessing the C&C Server, bots perform DNS queries in order to locate the particular C&C Server which is hosted by the DDNS provider. Therefore it is possible to Detect botnet DNS traffic by DNS monitoring and detect DNS traffic anomalies[[26] , [27]].During this stage, a detection mechanism is provided to analyze DNS traffic, detect possible communication instabilities and detect DNS anomalies (Choi, Lee et al. 2007; Villamarín-Salomón and Brustoloni 2008). Normally bots communicate within a single administrative domain and it is easy to measure the relationship between the bots and the C&C mechanism by analyzing different domain attributes such as the lifetime of the domain, TTL of the query, page ranking of domains, and how frequently a query is applied.

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 Botsniffer [29]. The similar malicious traffic 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 based, and P2P botnet with a very low false positive rate [28].

3.2.2 Detection Based on Anomaly Activities

This technique monitors any behavior that is abnormal by 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 concluded that this technique can also the unknown bots.This method is very efficient in detecting unknown bots and comprise of two phases- Training and Detection phase. In the training phase, the normal behavior system (in the absence of an attack) is observed and a profile is created, using machine learning techniques. In the detection phase, the current behavior of the system is compared to the created profile. However, it may use a lot of system resources as it has to constantly update the user and system profiles and it also generates a high false positive alarm [30]. The encrypted botnet communication can also be detected by this approach. The Anomaly Based Detection Technique can be categorized into Host based detection technique and Network based detection technique. The Host based technique is used to analyze and monitor the internals of the computer system instead of the network traffic on its external interfaces [30]. The Network based technique is used to detect the botnets by monitoring the network traffics and can be categorized into *Active monitoring* and *Passive monitoring* .Passive monitoring is based on the ability to inject test packets into the network, servers or application for measuring the reactions of network. Thus it can produce extra traffics. The Active monitoring uses some devices to inspect the traffics as they pass by. It does not increase the traffics on the network for inspection. This strategy usually requires a long time to inspect multiple stages or rounds of Botnet communication and activities to detect

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

IV. PROPOSED WORK

Presently the network traffic compromises of various types of data. For example web contents, e-mails, files, real-time audio/video data stream and many more. Depending upon the type of transmission needed either UDP or TCP is used as a transport layer protocol .For instance, for the transmission of web content, e-mails and files, TCP is used as a transport layer protocol as it is more reliable protocol. But for the transfer of time sensitive application like real time audio/video streams, UDP is used. The applications that used TCP protocol maintain a full duplex communication between the sender and the receiver and there is also the sequenced flow control between the two. To make a TCP connection between the sender and the receiver, the sender first sends the SYN packet to the receiver to initiate the session. After the initiation of connection, an [SYN, ACK] packet is sent by the sender indicating that a connection is maintained and now the sender can receive the packets without overwhelming and invading any of the internal buffer. At the end, the ACK packet is sent. This process is known as TCP 3 way Handshaking. Due to this ACK, the TCP protocol is more reliable than UDP protocol; still most of the P2P applications use UDP protocol for 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 of traffic and in the time measured. Also some of them are unidirectional in nature.

4.1 Detection of Bot from the network traffic by using the Wireshark

In this section we have captured the packets of the malware transmitting over the network and have analyzed the bot infected host by using a tool called Wireshark. Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues. Wireshark is very similar to tcpdump, but has a graphical front-end, plus some integrated sorting and filtering options. Wireshark is software that "understands" the structure (encapsulation) of different networking protocols. It can parse and display the fields, along with their meanings as specified by different networking protocols. Wireshark uses pcap to capture packets, so it can only capture packets on the types of networks that pcap supports. In the field of computer network administration, pcap (packet capture) consists of an application programming interface (API) for capturing network traffic. Unix-like systems implement pcap in the libpcap library; Windows uses a port of libpcap known as WinPcap. The pcap API is written in C, so other languages such 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

1. Data can be captured "from the wire" from a live network connection or read from a file of already-captured packets.

2. Live data can be read from a number of types of network, including Ethernet, IEEE 802.11, PPP, and loopback.

3. Network data can be browsed via a GUI, or via the terminal (command line) version of the utility, TShark.

4. Captured files can be programmatically edited or converted via command-line switches to the "editcap" program.

5. Data display can be refined using a display filter.

6. Plug-ins can be created for dissecting new protocols.

7. VoIP calls in the captured traffic can be detected. If encoded in a compatible encoding, the media flow can even be played.

8. Raw USB traffic can be captured.

9. Wireless connections can also be filtered as long as they transverse the monitored Ethernet.

10. Various settings, timers, and filters can be set that ensure only triggered traffic appear.

Wireshark's native network trace file format is the libpcap format supported by libpcap and WinPcap, so it can exchange captured network traces with other applications that use the same format, including tcpdump and CA NetMaster. It can also read captures from other network analyzers, such as snoop, Network General's Sniffer, and Microsoft Network Monitor. The user typically sees packets highlighted in green, blue, and black. Wireshark uses colors to help the user identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems — for example, they could have been delivered out-of-order. Users can change existing rules for coloring packets, add new rules, or remove rules.

We have created the Virtual Box in our system and have used Oracle. Then Ubantu operating system is being installed on it. Thus we have created a virtual environment so as to keep the system protected. The Wireshark is also installed in on Ubantu. We execute the malware in this virtual environment. The packets that were captured by Wireshark are analyzed in this section.

In figure 17 there are number of devices that are being 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).

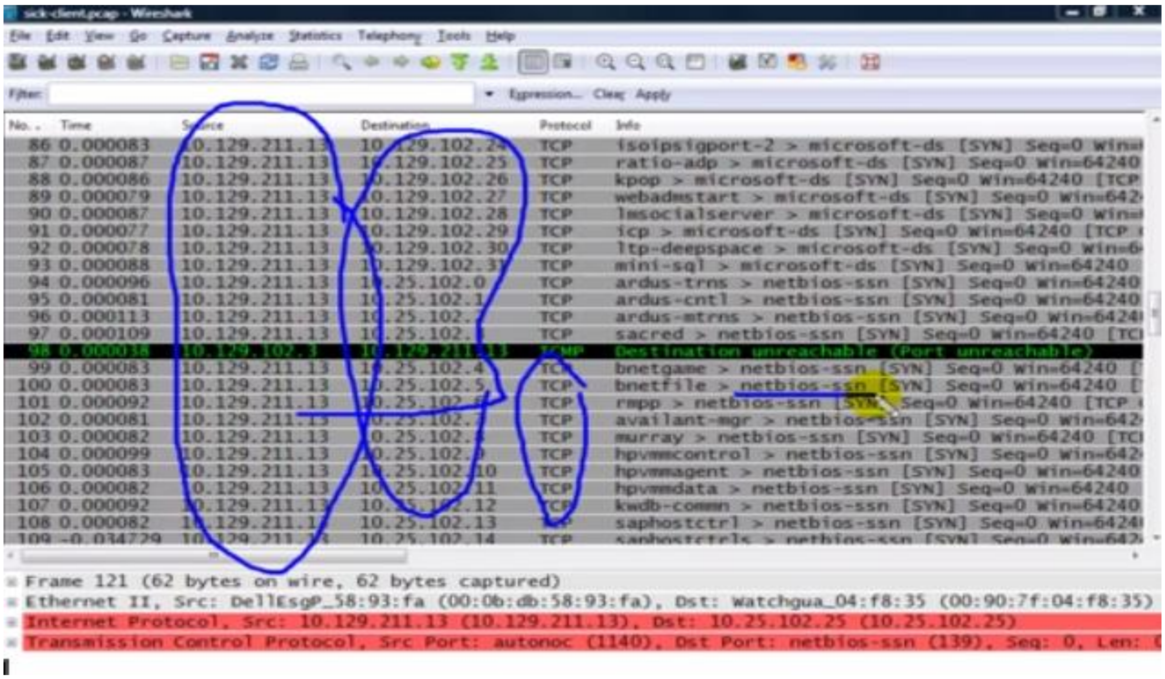

Figure 17 Handshake packets going out to the target systems.

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.

Figure 18 ICMP destination unreachable responses

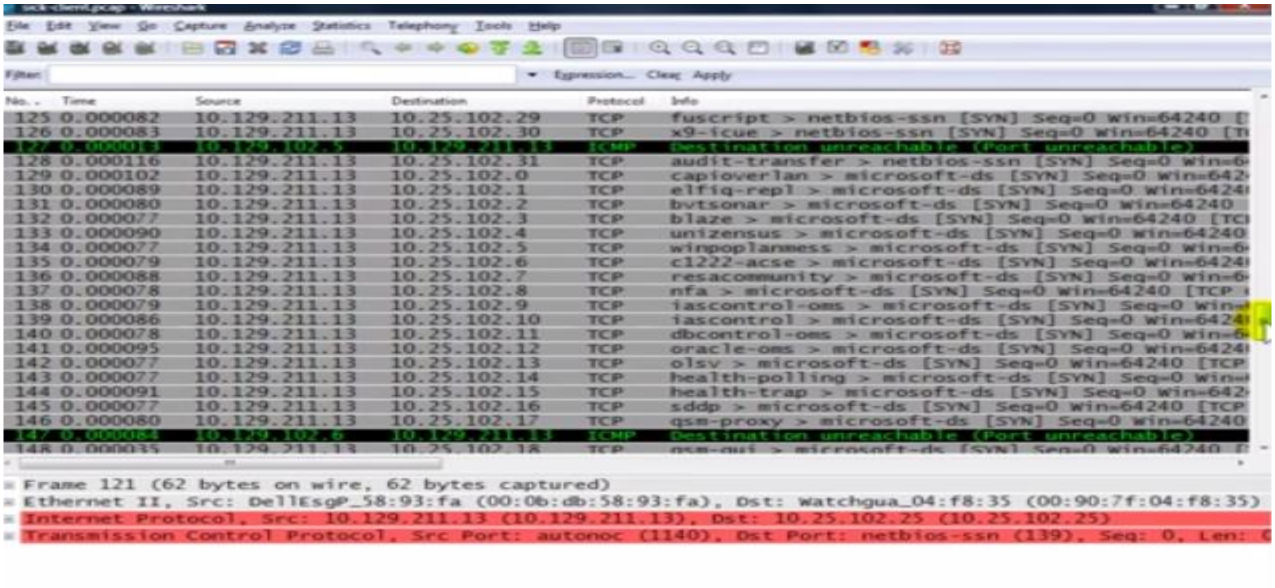

Figure 19 TCP scans going out on the system

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 nobo dy 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.

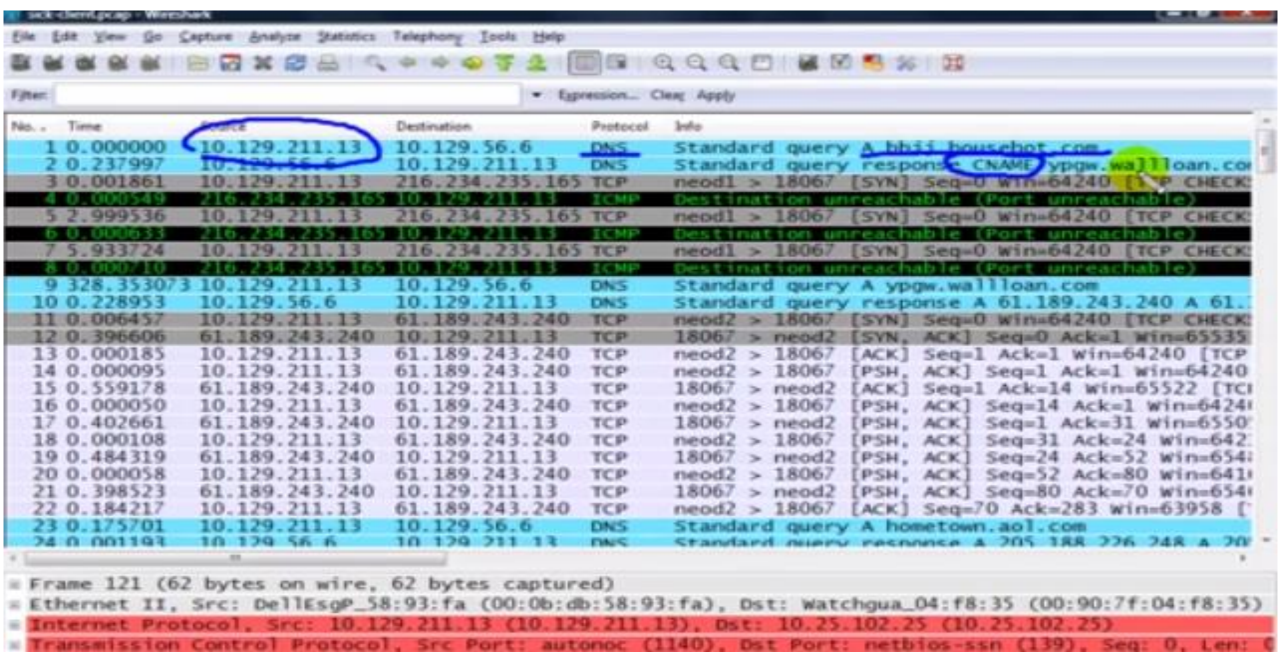

Figure 20 DNS query by the infected Host

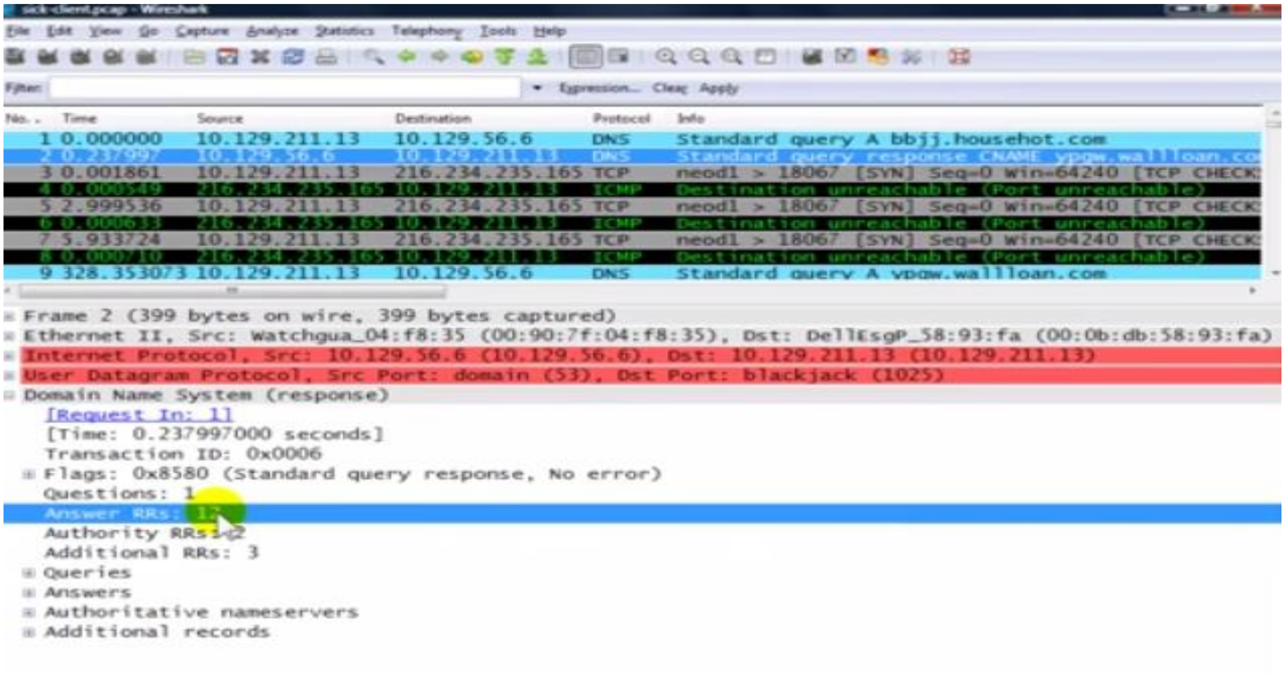

Figure 21 Unusual numbers of Answer Resource Records in the DNS response

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.

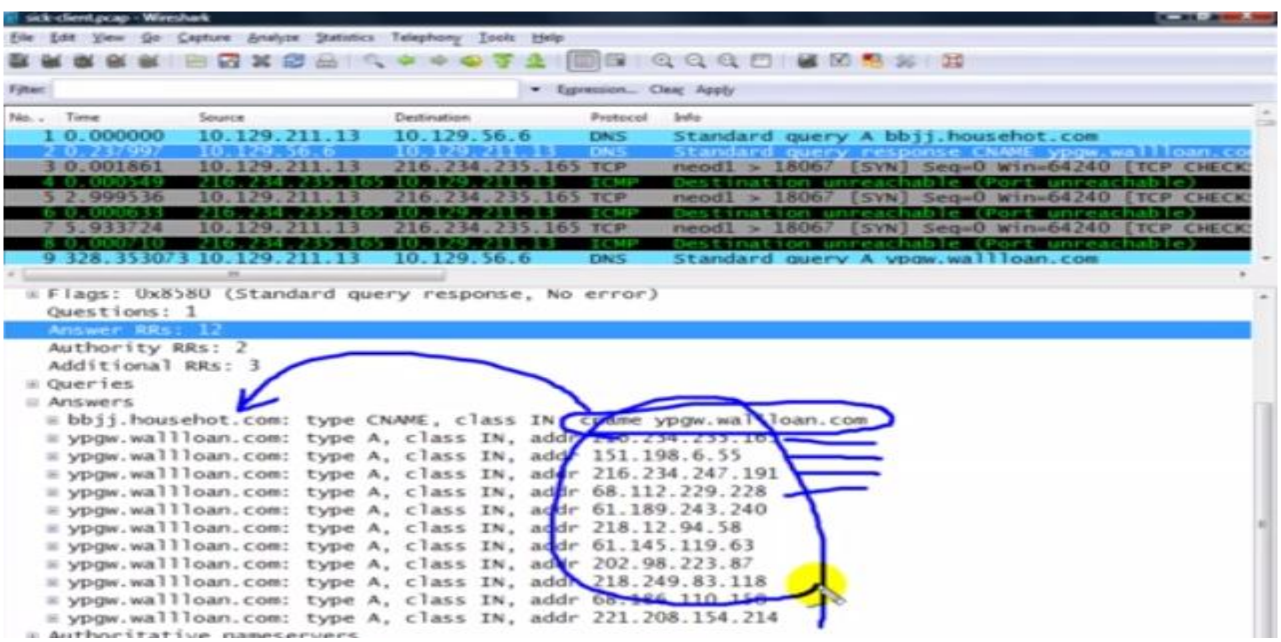

Figure 22 List of different IP addresses as DNS response

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.

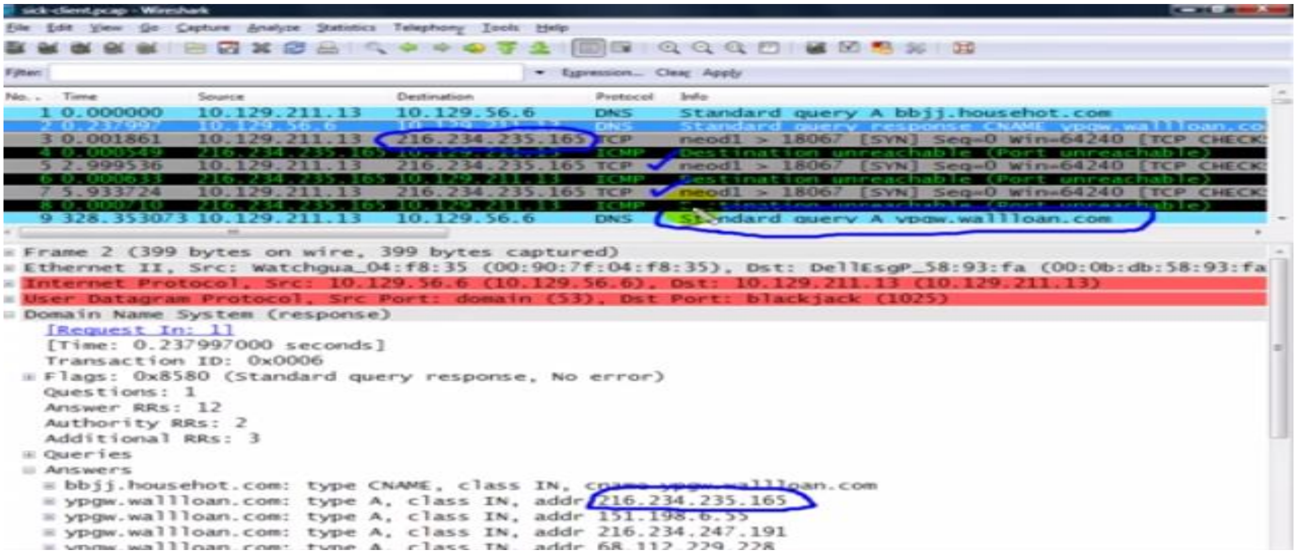

Figure 23 Unsuccessful TCP Handshakes

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.

.

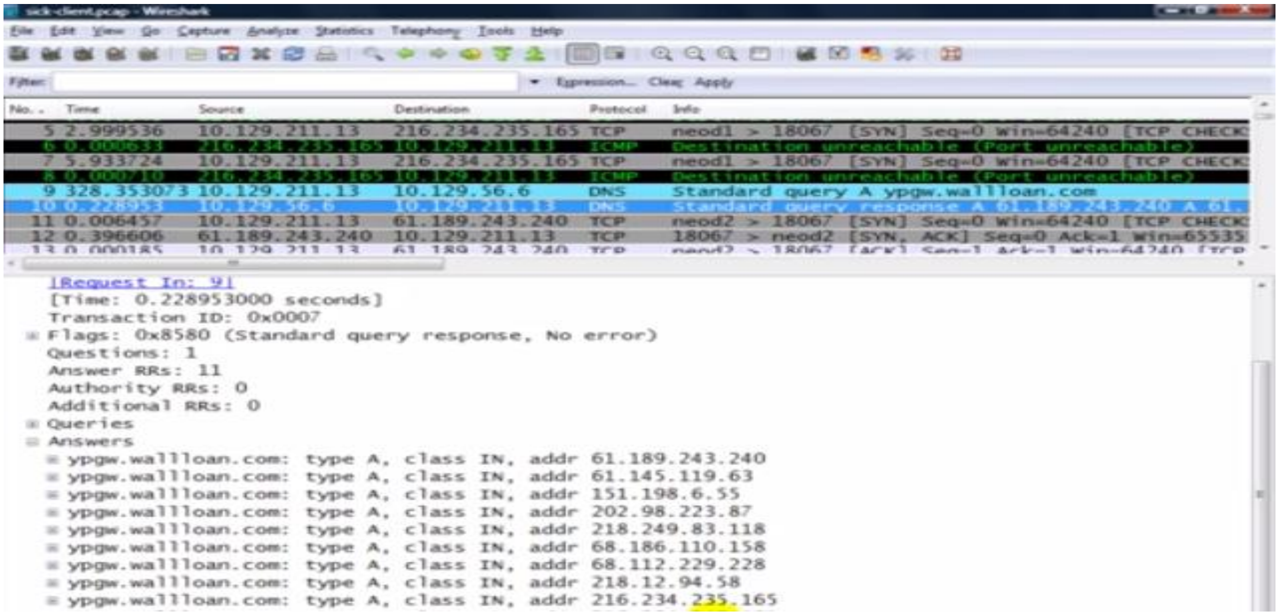

Figure 24 DNS response for ypgw.wallloan.com

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.

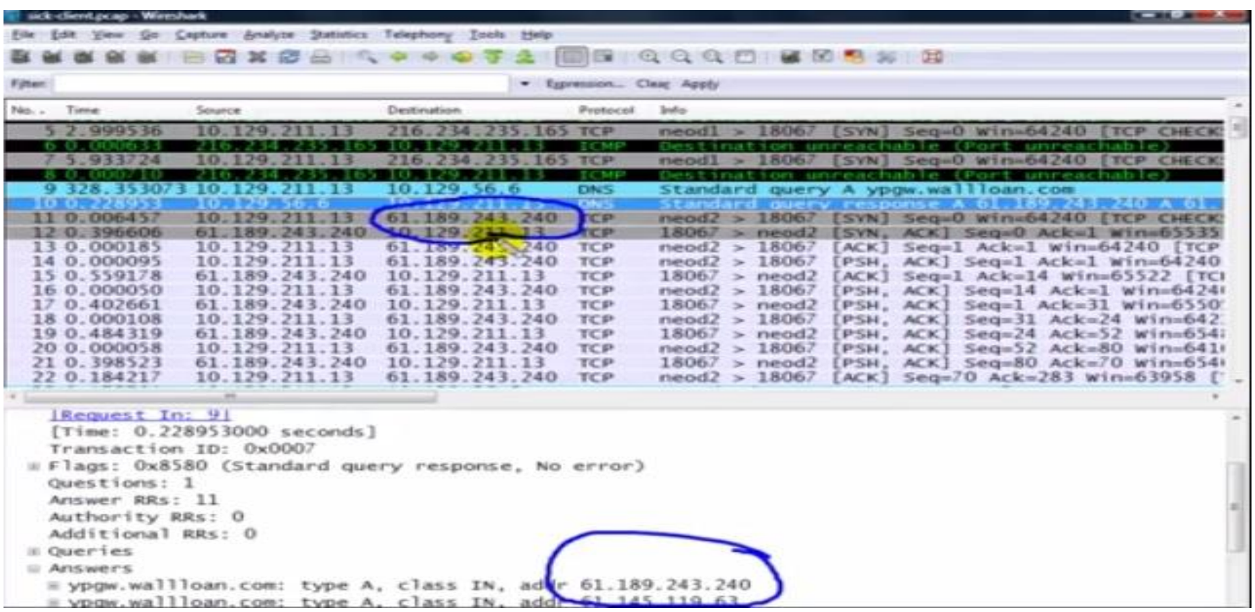

Figure 25 TCP Handshake between the client and the target system

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.

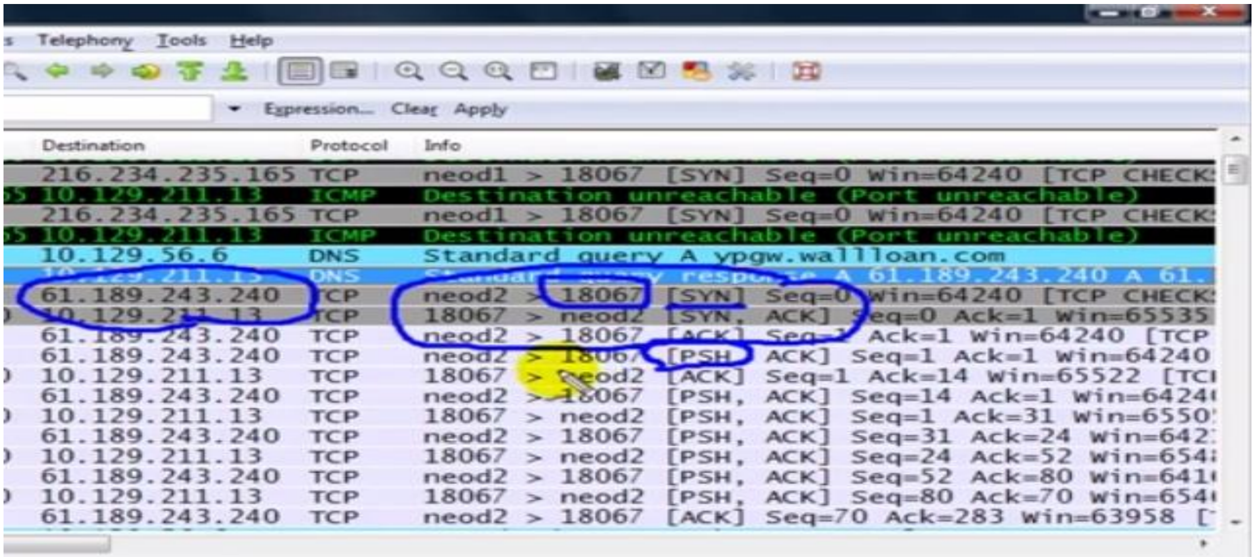

Figure 26 Push flag sent with the ACK after the successful TCP connection

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.

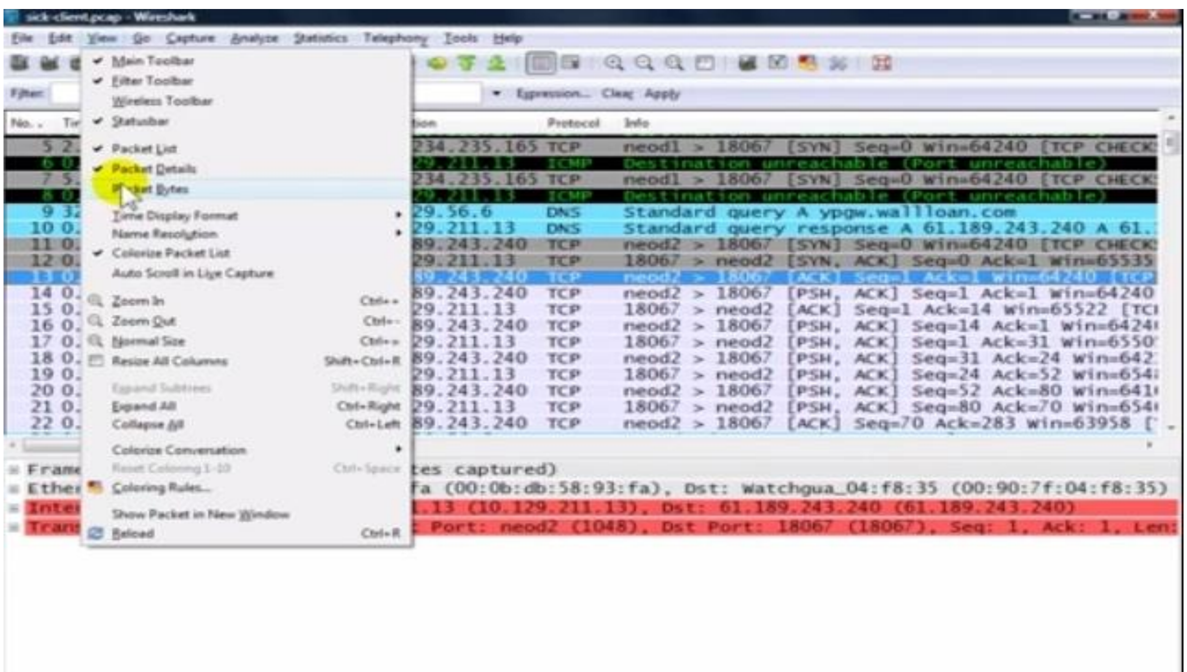

Figure 27 Opening the Packet Bytes section

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

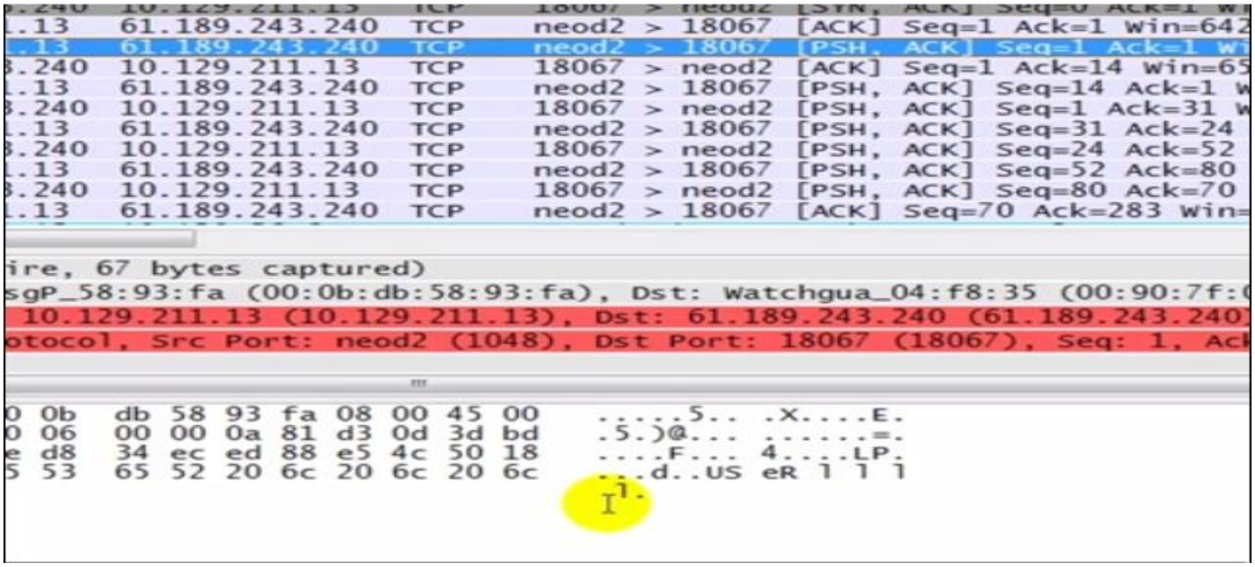

Figure 28 Data sent to the destination “61.189.243.240”

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.

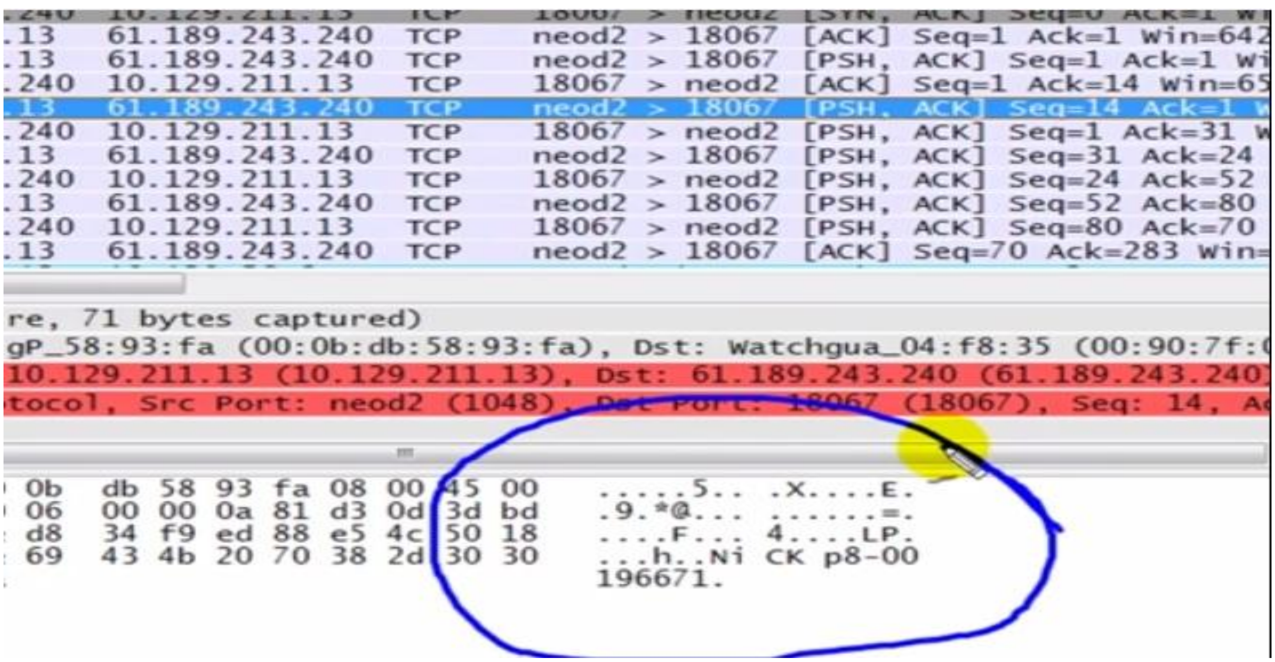

Figure 29 Additional information is sent to the destination “61.189.243.240”

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.

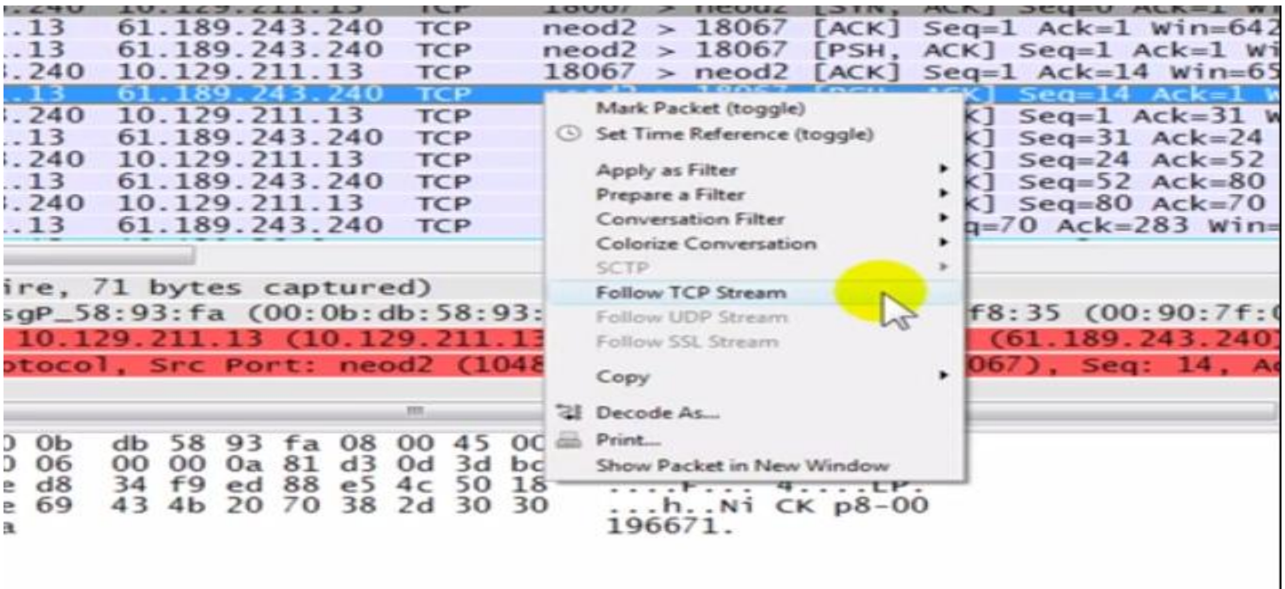

Figure 30 Following the TCP stream

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.

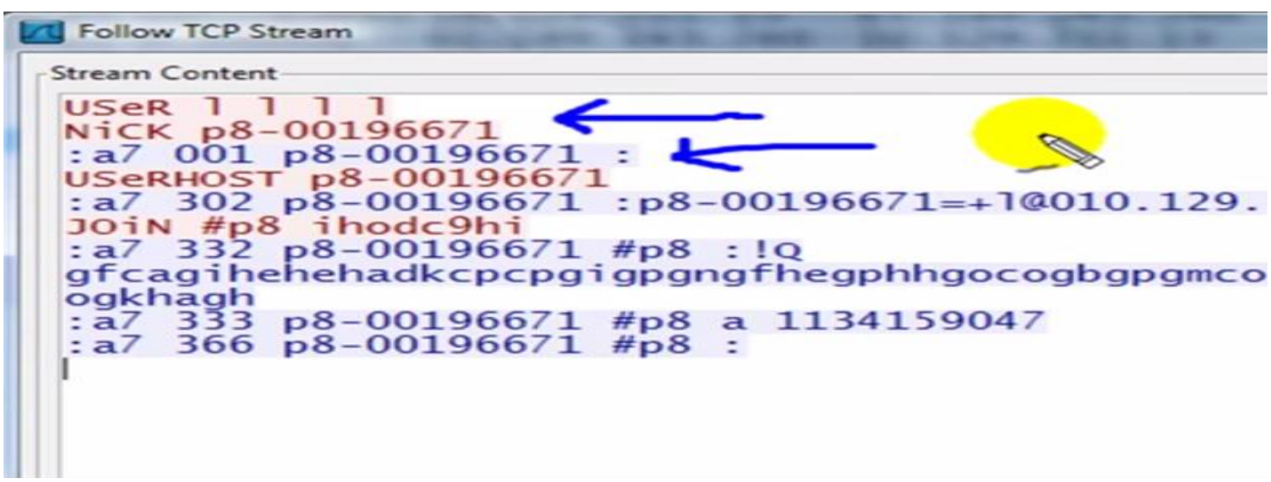

Figure 31 IRC communication

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.

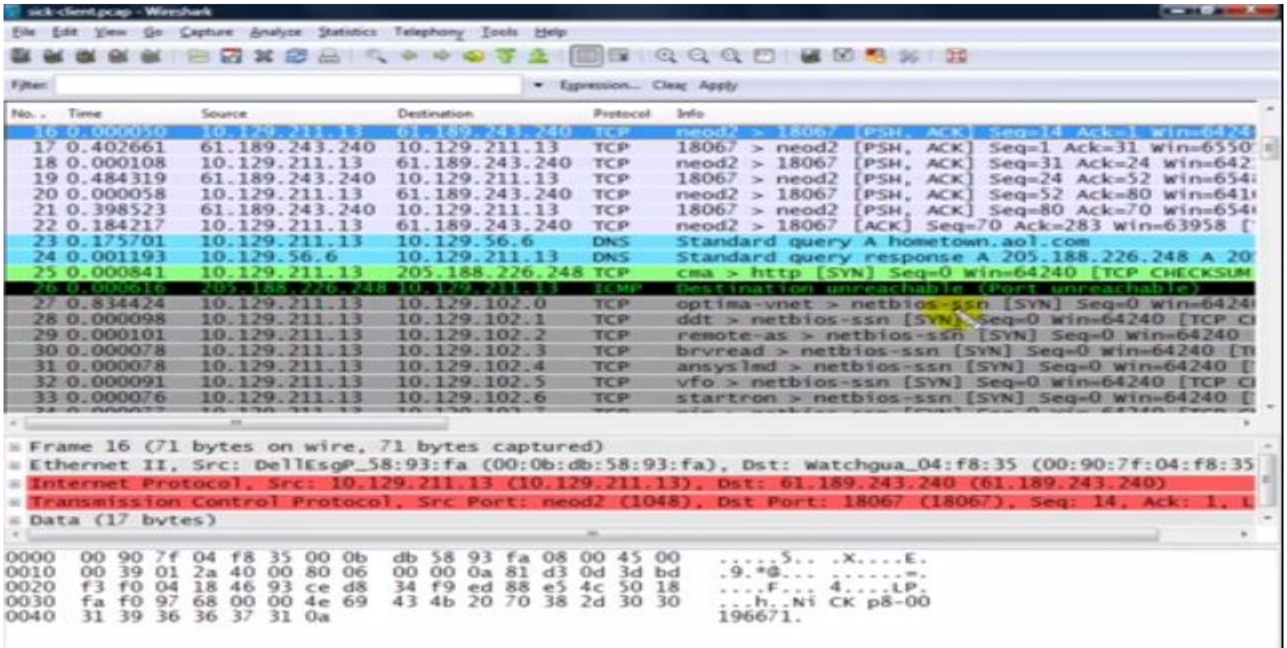

Figure 32 Client does a query for "hometown.com" and gets back a DNS response

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).

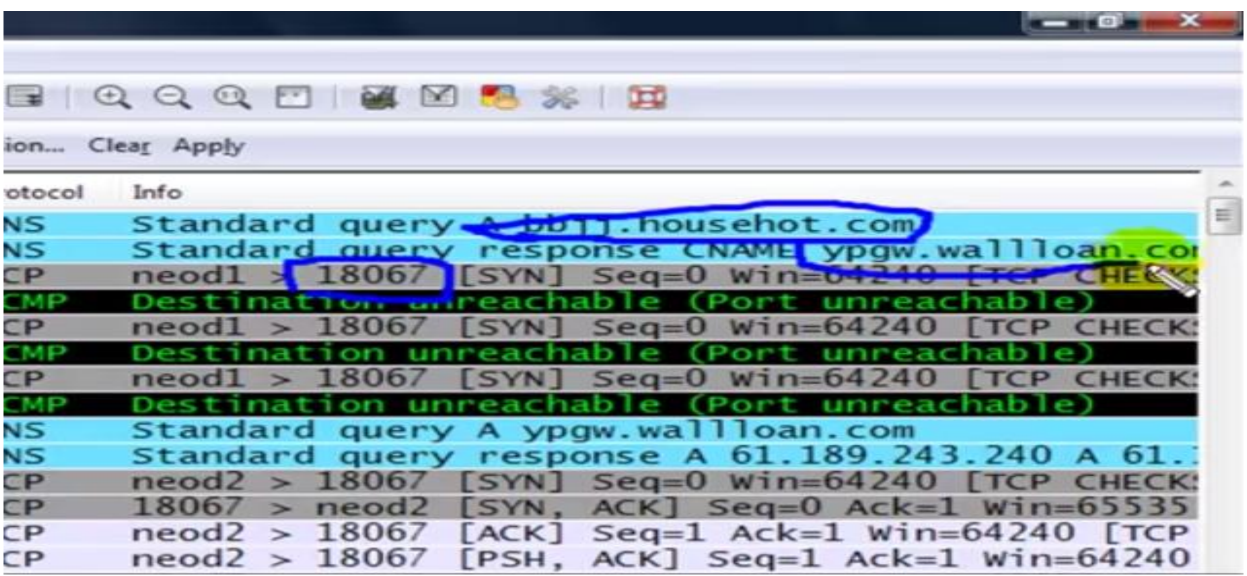

Figure 33 Signatures that indicate the abnormal activity

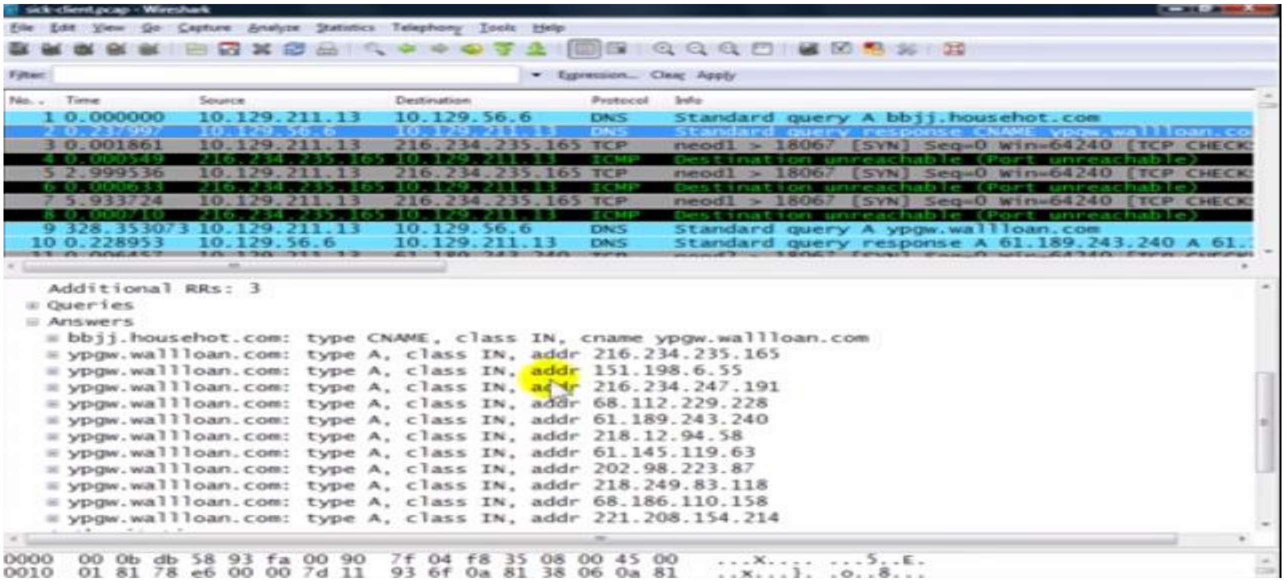

Figure 34 List of IP addresses that came in DNS response

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.

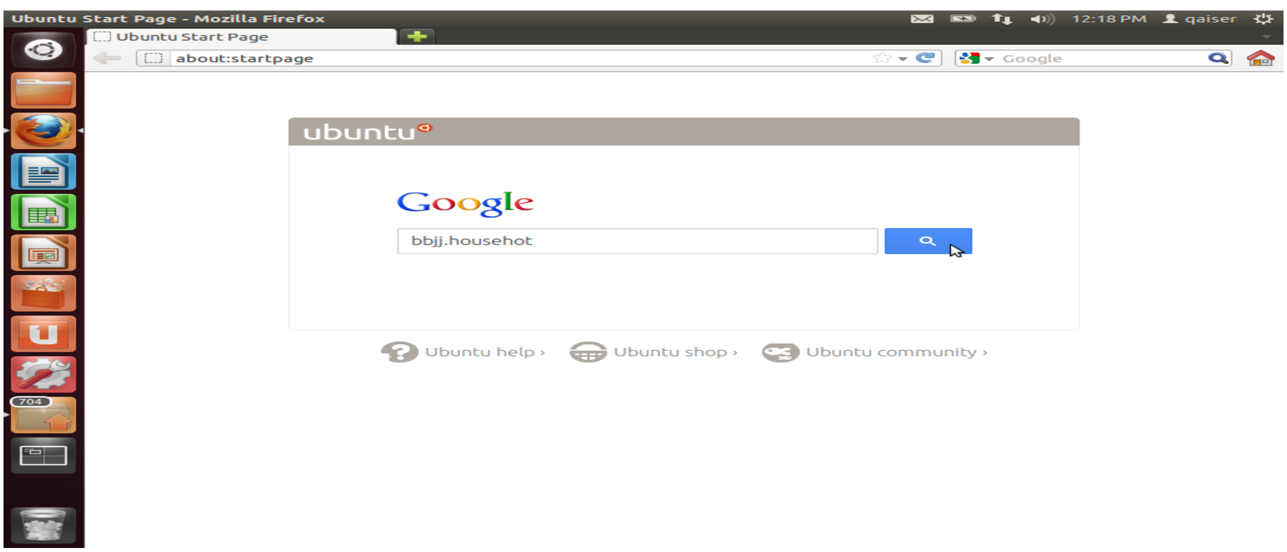

Figure 35 Search for “bbjj.househot.com”.

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.

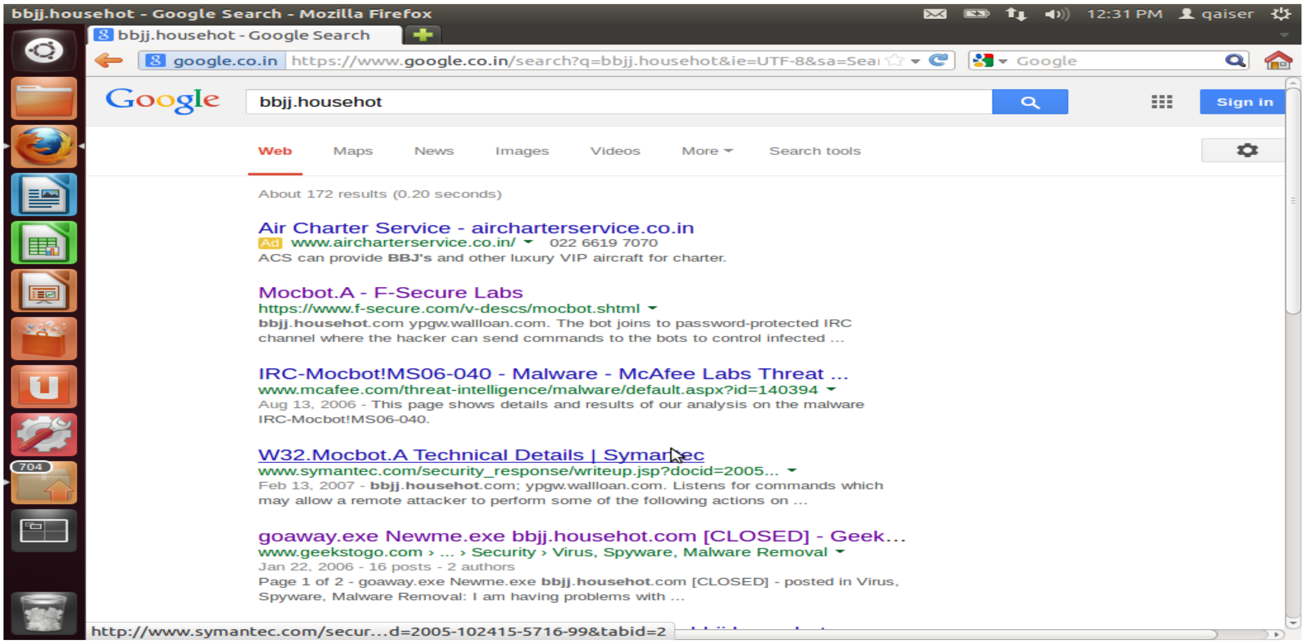

Figure 36 Shows the result for “bbjj.househot.com”

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”.

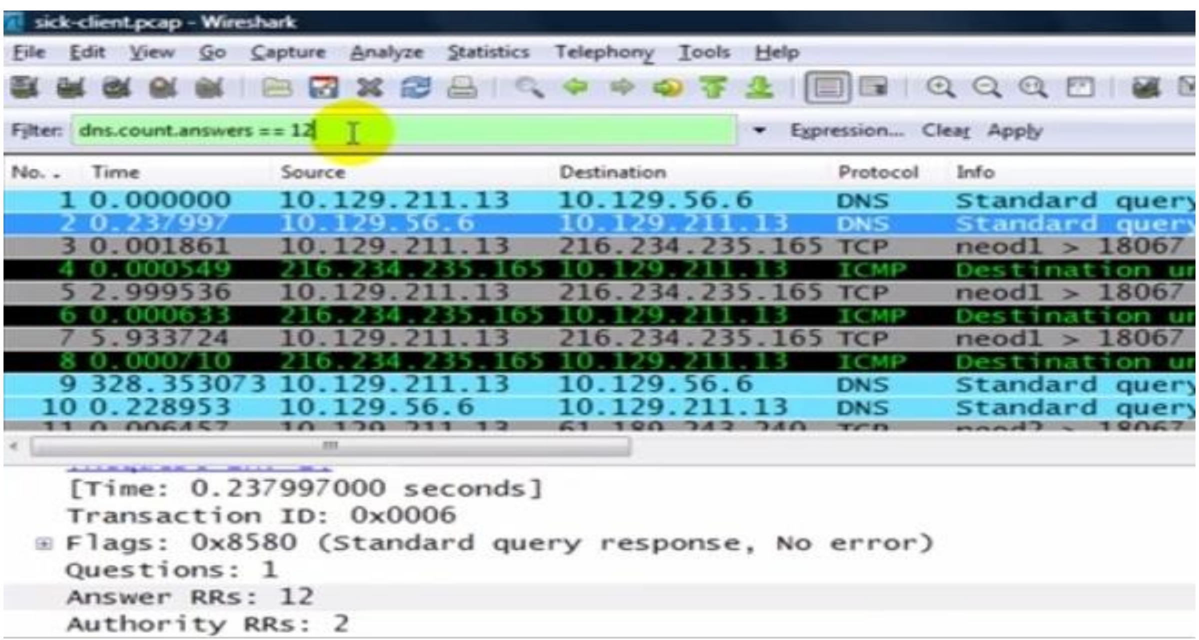

Figure 37 Filter is used to get DNS responses having Answers Resource Records greater than 12.

We did Right Click on this field and prepare a filter based on the selected value as shown in figure 38.

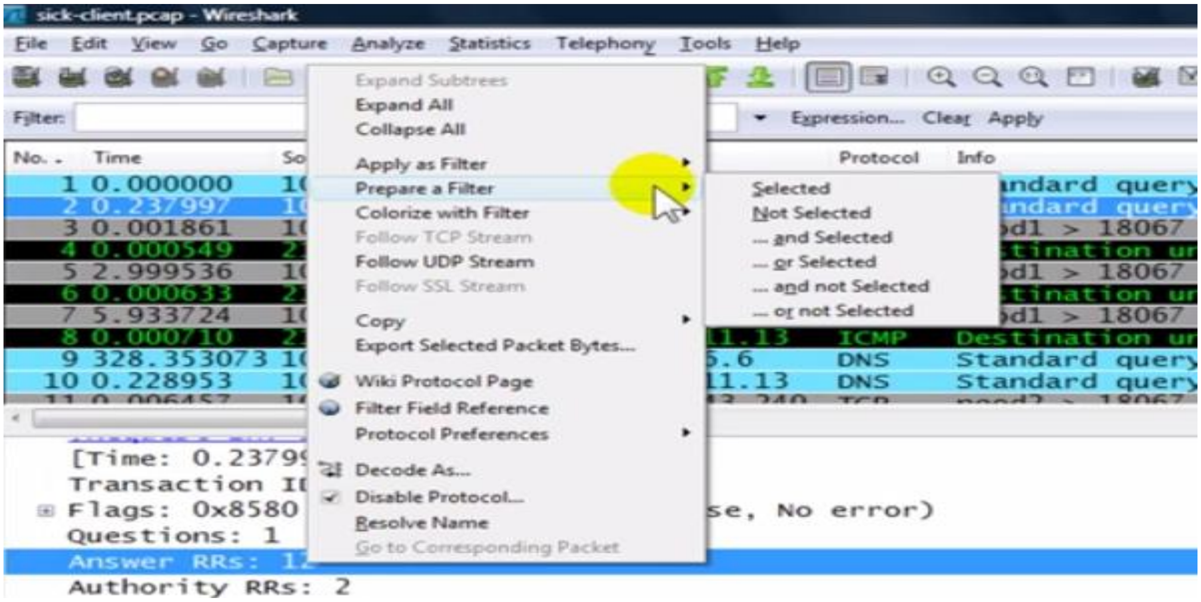

Figure 38 Prepare Filter based on selected value

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.

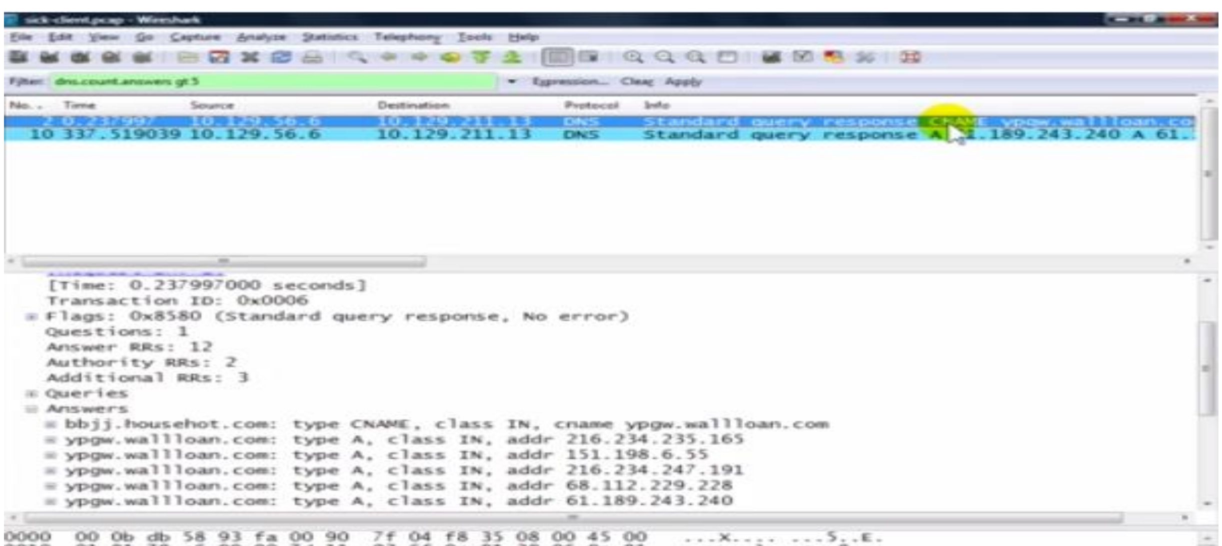

Figure 39 Filter is used to get DNS responses having Answers Resource Records greater than 5

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

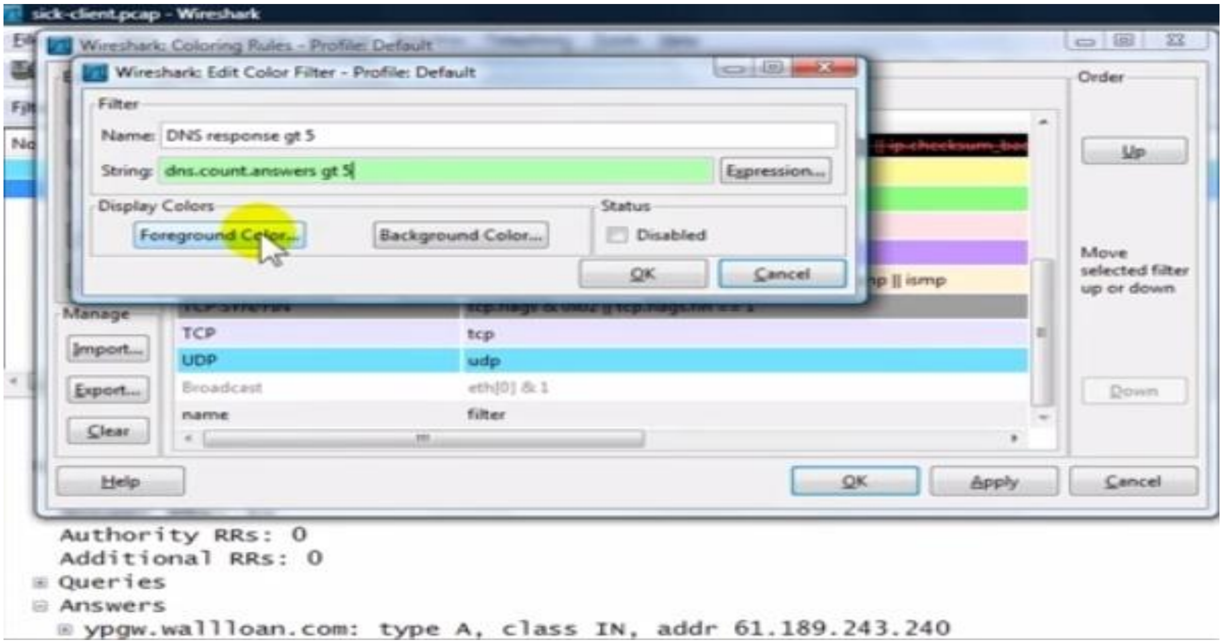

Figure 40 Select the color for foreground area.

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).

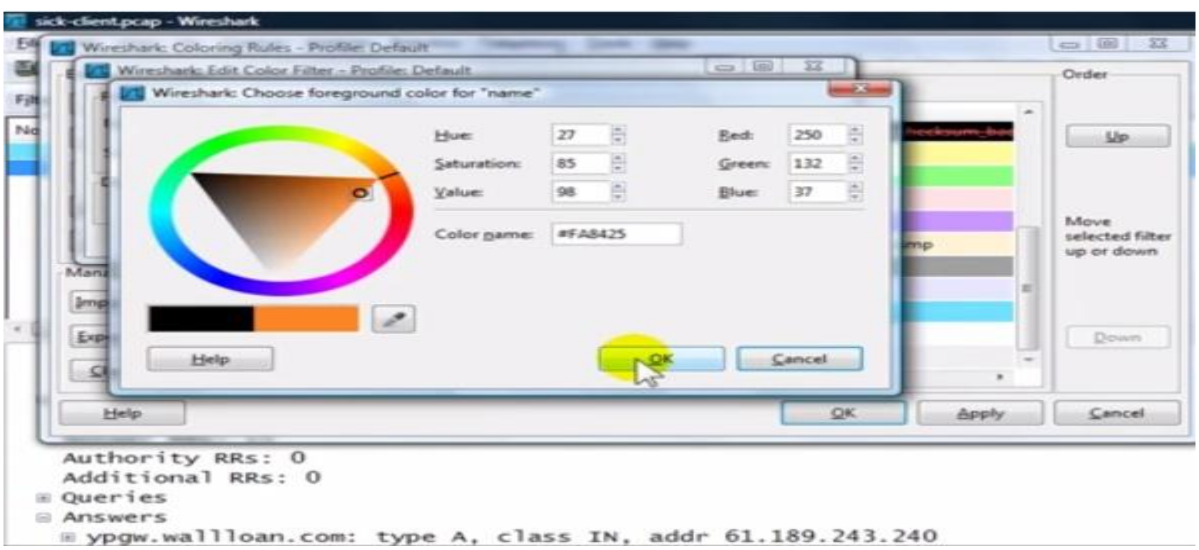

Figure 41 Orange is selected as foreground color

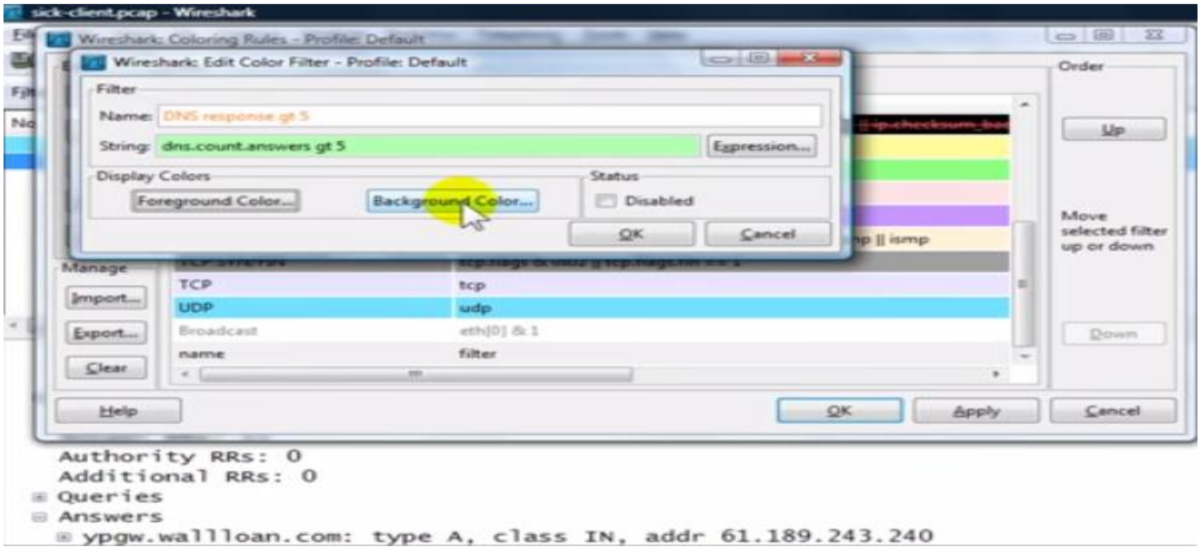

Figure 42 Select the color for background area

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.

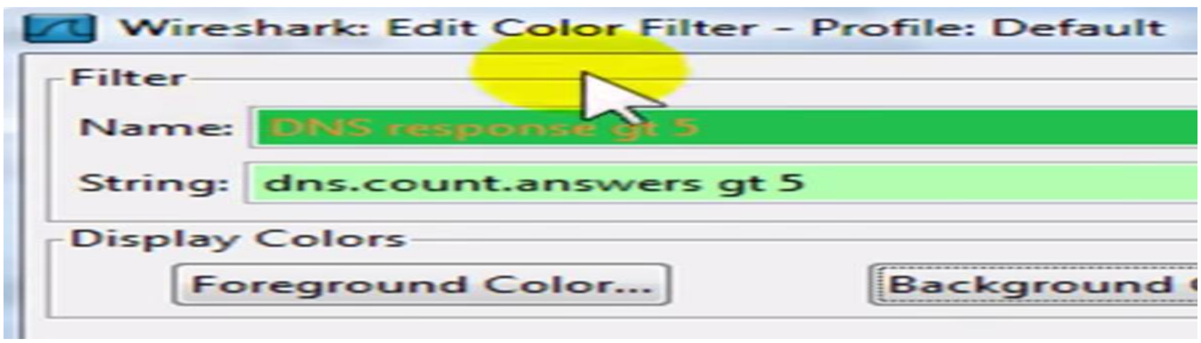

Figure 43 Edit Color Filter shows the colored foreground and background area

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

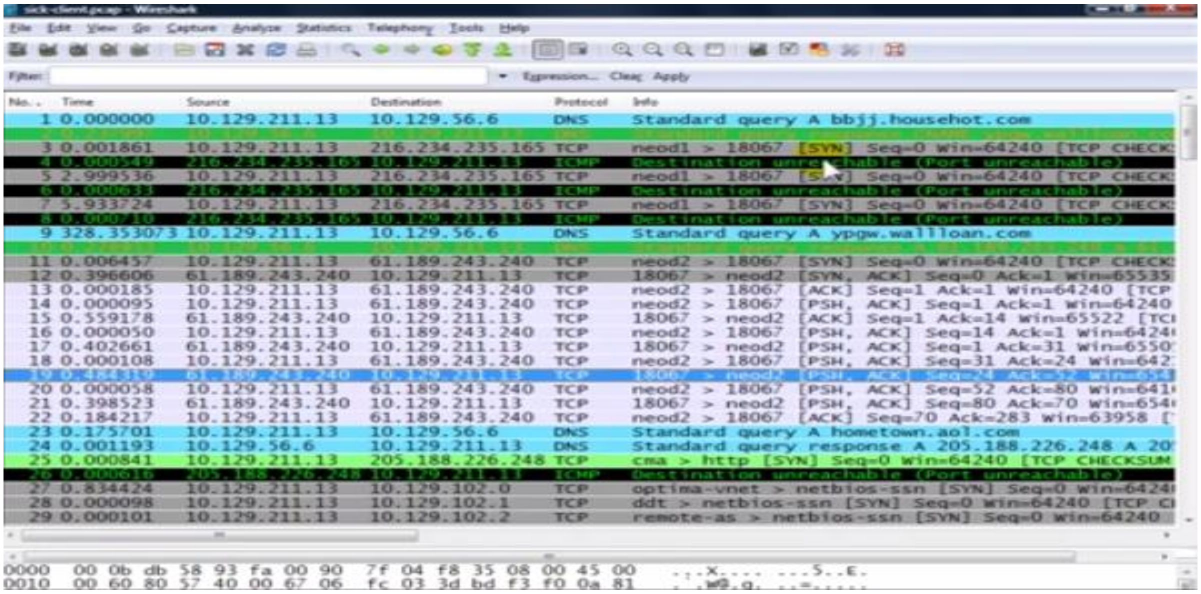

Figure 44 Results after applying the Butt-Ugly Filter

As we analyze botnet effected system we see that there is a similarities in the packets they request, the replies that come and also the data of the configuration file downloaded by the bot program used to launch the attacks. Thus we need to stop our system from becoming a bot in a botnet. This can be done in two steps.

Analyzing the traffic: This is done by seeing the DNS replies and if the answer field has more than few entities then we can just discard and quarantine such packet till the user or system-administrator looks into the contents of the DNS request and reply and decide if they are genuine or generated by the malicious program (botnet) that might have infiltrated our system. Discarding such packets will stop the bot program running on our computer from communicating with the C&C server making it unable to download the configuration file and thus stop the bot from performing the attack.

Machine Based learning system: This technique is based on a filtering program, which needs to be trained by using a training set, comprising of the similarities in a botnet communication steps or the file downloaded. If any of the communication steps or file downloaded matches the filter of the filter program it quarantines it and thus stopping the bot to perform its attack.

4.2 Generic Architecture for detecting the Botnet from the network traffic

In order to detect the botnet, we need to follow an effective way so that we can detect the bots as early as possible. We have designed a generic Architecture for effectively detecting the bots by monitoring the network traffic over the internet. The internet is widely used by people all over the world, having both legal and illegal users. The applications used on internet can be many like LinkedIn, Google +, Skype, Instagram, Twitter, Facebook, YouTube and much more. All these applications will provide a number of benefits but only if they are used in a responsible way. At the same time the attackers are also present on the internet to perform the illegal activities. All the activities going on the internet will generate the network traffic. The incoming and outgoing network traffic is first sent to the network traffic assembler containing the repository where the network packets are stored for the future use. There are number of tools used for assembling the network packets. We have used Wireshark (Network Protocol Analyzer) for capturing the network flow. The captured packets are then passed through the Filter that helps in reducing the traffic burden. There are two methods commonly used for filtering the network flow, they are White Page and Black Page filtering techniques. The legitimate packets like antivirus updates are filtered by White page filtering technique and the malicious packets like viruses, Trojans are filtered by Black Paper filtering technique.

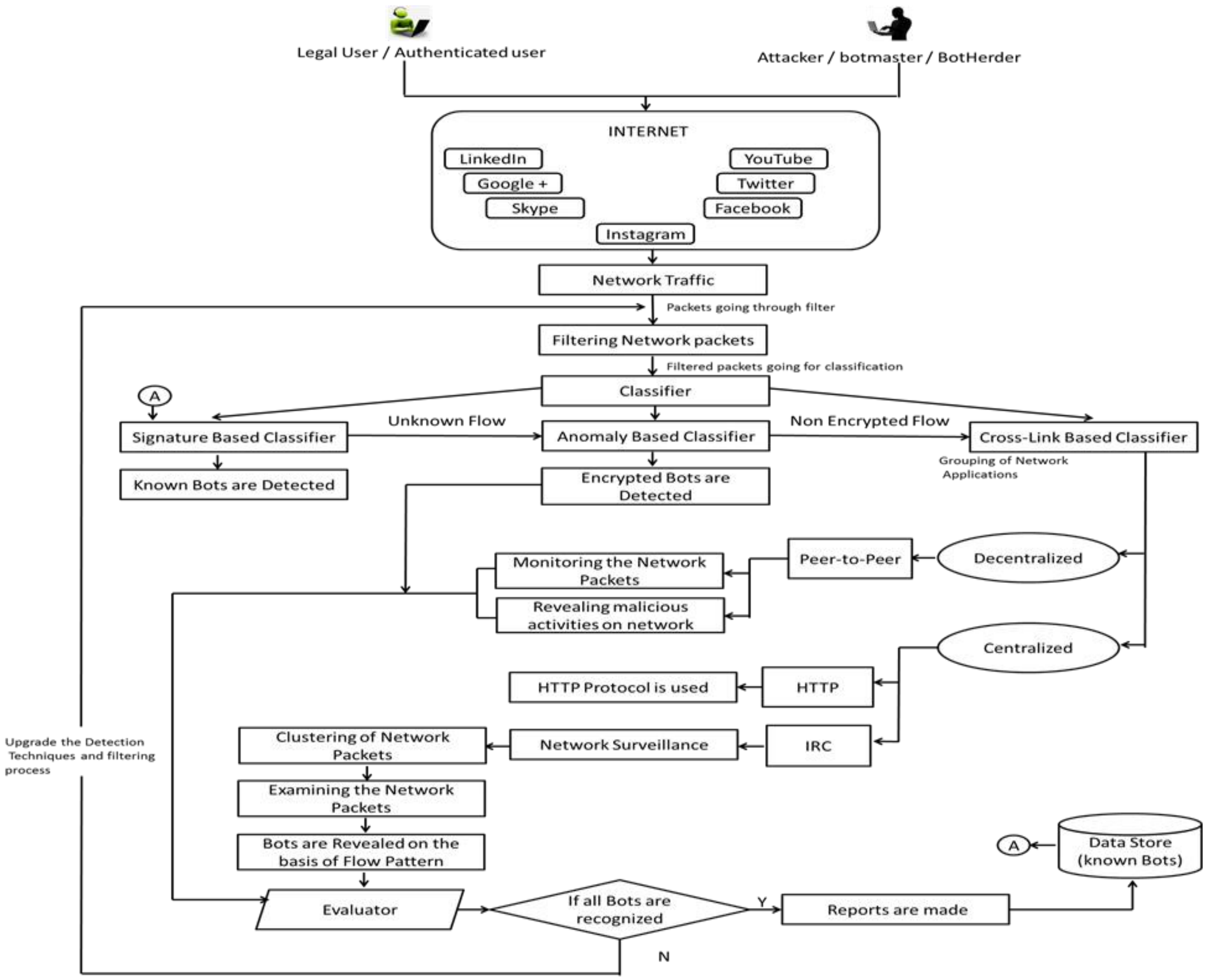

Figure 45 Generic Architecture for detection of botnet from the network traffic

The filtered network flow is passed through the Classifier. Three types of Classifiers have been used namely Signature Based Classifier, Anomaly Based Classifier and Cross Link Based Classifier (figure 45). The known bots are detected by Signature Based Classifier. It helps in minimizing the false positive rate as this technique only detects the known bots. The rest of the network traffic is left with unknown flow, which is passed through the Anomaly Based Classifier. It detects the encrypted bots only, leaving behind the non-encrypted network traffic. The encrypted traffic detected is then passed through the Evaluator or the Analyzer. The non-encrypted network flow is passed through the Cross-Link Based Classifier. It classifies the non-encrypted network flow into the different network applications.

We have grouped the network flow into the two applications i.e.; Centralized and Decentralized applications. The P2P (Peer-to-Peer) network traffic is a type of decentralized application where no single unit is accountable for providing or issuing C&C (Command and Control) to bots. Here the bots are either distributed among the multiple servers or there is no obvious master-slave relationship between C&C server and bots. The P2P traffic is monitored by using the Traffic monitoring module, in order to discover the group of hosts having same behavior and communication pattern. The possible malicious activities that are related to the P2P based packets are detected by the malicious activity detector. The IRC and HTTP network traffic is a type of Centralized applications (having single C&C Server).

The Centralized network traffic (IRC) is sent for the surveillance or monitoring. The monitored network traffic is then clustered or grouped and then examined. After the close examination of the network packets, the bots are detected on the basis of flow pattern and are passed through the network packet evaluator, which analyzes the unknown packets so that no information is lost. If all the bots are discovered, the reports are generated and are updated into the data store. Else, the Detection techniques and the Filtering process are upgraded and the filtering of the network flow is restarted.

V. RESEARCH CHALLENGES

5.1 Detection: Detecting the botnet in a system or the network is a major task. A botnet is considered to be a group of the compromised systems also known as zombies, which are under the control and command of the single botmaster. These bots keep on forming again and again with the help of the different types of the network architecture and various applications and using topologies and the digital signatures also [134]. The Firewall and IDS system are used to detect and identifying the attacks from the botnets and also a Honeypot is used to detect any malicious program and mitigating the attacks. But if there is a continuous attack going on, then detecting the botnet with the help of these systems will be difficult. So it requires some advanced techniques or systems.

5.2 Botnet size: The size of the botnet depends on the number of the bots attacking a system. Generally, the size of the botnet expands greatly and moreover, there are various botnets which consists of the million bots which can be used to launch large and powerful attack. For example, botnet Zeus has more than million of bots and botnet Waladac have the strength of sending 1.5 billion spams per day. Therefore the size of the botnet is a major challenge [135].

5.3 AnalysisS: As the botnets are both reactive and proactive in nature therefore, analysis can be done in both the active as well as the passive mode. An example of the active analysis is the honeypot, but due to its difficult setup its use is restricted for the large scale networks. And the passive analysis is performed on the network data traffic collected and can also identify many botnets at a time but it is limited to some specific types of botnets only.

5.4 Investigation: For detecting the botnet attack and collecting the data about the botnet, various types of the detection techniques are used to perform an investigation. To present our evidence and fulfill our criteria in a court of law, an acknowledgement of the attack is being used to precede the investigation process and thus generating the required result. Thus, investigating a botnet is also a one of the major challenge.

5.5 Server failure: It is one of the biggest challenges while detecting the botnets. If the server failed during the process or while collecting the packets or required information, then it is possible that all the data captured or detected is lost anyway and then there will be no proof. Server failure can relate to the DNS failures or the failures related to the name servers [137].

5.6 Cryptography: One of the important parts of the botnet is to maintain the integrity and authentication of the system or the entire network, which can be violated by an attacker through any means. Thus in keeping it all confidential throughout the process is a difficult task.

VI. CONCLUSION

Botnet is a very distinctive technology used by attacker which is very extensive in nature, thus due to this, the botnet research is still in inception. The botnet discriminates itself from other malware in the ability of its compromised machines to establish command and control with remote server controlled by human misfeasor. Every stage of the life cycle of botnet must be successfully completed if the botnet is to succeed. Therefore, even if the execution of just one stage is interrupted, it will render the whole botnet detection. This paper surveys state-of-art botnet research that can be categorized into the areas namely, (1) Botnet review and sum up. (2) Botnet revelation and botnet revelation techniques. (3) Classification of botnets based on its architecture or topology. In this paper, various botnet detection techniques have been discussed, among them only Signature- based technique is the only one that can't detect the unknown botnet. Most of the Botnet detection techniques based on DNS and Data mining can detect real –world botnets regardless of the botnet protocol and structure with a very low false positive rate. Only Mining–based botnets have the capability to detect the encrypted botnet. Data mining and machine learning techniques are well suited on flow information. Botnet detection techniques gather this information from bots to interpret their behavior and revelation mechanism. However, a large number of challenges still persist in the area of Botnet Detection.

A number of research works have been done for P2P and IRC botnets, but the motivations for using the HTTP protocol are multiple. For IRC –based botnets, the problem is that we can't get the source code of the most of the bots. The main issues related to P2P botnets are – hiding the botnet topology while some bots are apprehended by protector, changing the traffic patterns more often and making it harder for detection. Detecting the compromised hosts in the botnet will continue to be a challenging task. Anomaly detection is a feasible approach for detecting botnets. The interesting issue about this approach is time efficiency .If the attack occurs and we can capture the anomaly in the first place and fix the relevant problems before it is used for performing the abnormal activities, we say anomaly detection is time effective. We need to work on this time efficiency in future.

The botnets are turning to cloud computing to expand their potentials. The cloud platform is used by the botnets in two ways – host the C&C server on the cloud or create bots on the cloud instead of infecting user machine. The cloud security is still in a transient stage and most of the existing detection techniques do not scale to clouds, therefore, clouds provide a nice cover to botnets for carrying out their malicious activities. The mobile phones can utilize a number of communications like 3G, 4G which multiplies the possibilities for C&C and malware propagation.

REFFERENCES

[1] 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/articleDetails.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/getacceptedmsfile.aspx? msid=692654&vnum=2&ftype=manuscript [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/11390.

[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 network-based defense mechanisms countering the DoS and DDoS problems. ACM Computing Surveys (CSUR), 2007.