# An Introduction to Botnet Attacks and it's Solutions

Kalpna Midha
Research Scholar, Sri Ganganagar
Rajasthan

Kusum Rajawat
Research Scholar , Jaipur
Rajasthan

Dr. Vijay Singh Rathore
Director, Sh. Karni College
Jaipur , Rajasthan

## ABSTRACT

Usually though, when people talk about botnets, they are talking about a group of computers infected with a malicious kind of robot software, the bots, which present a security threat to the computer owner. Once the robot software (also known as malicious software or malware) has been successfully installed in a computer, the computer becomes a zombie or a drone, unable to resist the bot commander's commands. This paper studies the effectiveness of monitoring lookups to a DNS-based black hole list (DNSBL) to expose botnet membership. We perform counter-intelligence based on the insight that botmasters themselves perform DNSBL lookups to determine whether their spamming bots are blacklisted. Using heuristics to identify which DNSBL lookups are perpetrated by a botmaster performing such reconnaissance, we are able to compile a list of likely bots. This paper studies the prevalence of DNSBL reconnaissance observed at a mirror of a well-known blacklist for a 45-day period, identifies the means by which bot masters are performing reconnaissance, and suggests the possibility of using counter-intelligence to discover likely bots. We find that bots are performing reconnaissance on behalf of other bots. Based on this finding, we suggest counterintelligence techniques that may be useful for early bot detection. A botnet (also known as a zombie army) is a number of Internet computers that, although their owners are unaware of it, have been set up to forward transmissions (including spam or viruses) to other computers on the Internet. Any such computer is referred to as a zombie - ineffect, a computer "robot" or "bot" that serves the wishes of some master spam or virus originator. Most computers compromised in this way are home-based.

According to a report today from The Associated Press, Internet security company Prevx recently discovered a Web site that was being used as a storage facility for data stolen from 160K infected computers, and the discovery offers an interesting case study. The storage site was hosted in the Ukraine and its contents showed that the botnet was harvesting data. Information found included passwords, social security numbers, credit card numbers, addresses, telephone numbers and other personal information; quite a treasure chest if you're into identity theft. According to the article, both government and bank sites had also been compromised. The Associated Press contacted one bank customer whose Social Security number and other personal details were compromised during the attack, only to learn that he hadn't been notified by the bank.

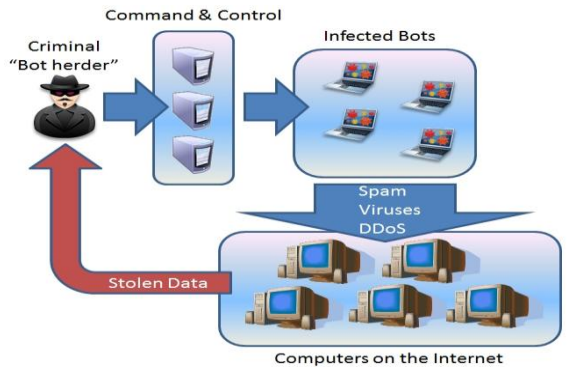
**Keywords** :

***Broadband, HTML, Honeynets, Virus***

## INTRODUCTION

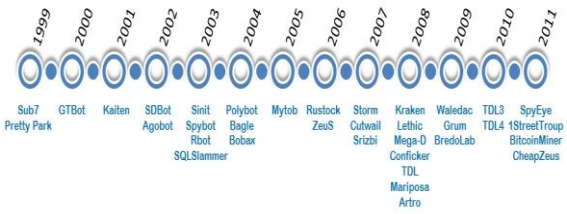
These days, home PCs are a desirable target for attackers. Most of these systems run Microsoft Windows and often are not properly patched or secured behind a firewall, leaving them vulnerable to attack. In addition to these direct attacks, indirect attacks against programs the victim uses are steadily increasing. Examples of these indirect attacks include malicious HTML-files that exploit vulnerabilities in Microsoft's Internet Explorer or attacks using malware in Peer-to-Peer networks. Especially machines with broadband connection that are always on are a valuable target for attackers. As broadband connections increase, so to do the number of potential victims of attacks. Crackers benefit from this situation and use it for their own advantage. With automated techniques they scan specific network ranges of the Internet searching for vulnerable systems with known weaknesses. Attackers often target Class B networks (/16 in CIDR notation) or smaller net-ranges. Once these attackers have compromised a machine, they install a so called IRC bot - also called zombie or drone - on it.

The term bot is short for robot. Criminals distribute malicious software (also known as malware) that can turn your computer into a bot (also known as a zombie). When this occurs, your computer can perform automated tasks over the Internet, without you knowing it. Criminals typically use bots to infect large numbers of computers. These computers form a network, or a botnet.




Criminals use botnets to send out spam email messages, spread viruses, attack computers and servers, and commit other kinds of crime and fraud. If your computer becomes part of a botnet, your computer might slow down and you might inadvertently be helping criminals.

## LITERATURE REVIEW




The first bot, "GMBot", was not malicious—it was created in the late 1980s to emulate a live person in Internet Relay sChat

(IRC) sessions. However, around 1999 bots emerged that were designed with harmful intentions; Sub7 and Pretty Park used IRC as a Command and Control channel.

Subsequent bots grew more sophisticated, and in some cases were commercialized as products; the Zeus bot of 2006 originally sold for several thousand dollars. IRC was replaced by protocols such as HTTP, ICMP, and SSL for command and control of a network of compromised systems.

In mid-2011, source code for the Zeus and SpyEye botnet kits was leaked, making these powerful botnet creators available to practically anyone that wants to establish their own botnet.

"Notable points along the botnet timeline are numerous. First up, the emergence of the Global Threat bot, or GTbot, in 2000. GTbot was based on the mIRC client, which meant that it could run custom scripts in response to IRC events and also importantly that it had access to raw TCP and UDP sockets, making it perfect for rudimentary Denial of Service attacks, some attacks went as far as scanning for Sub7 infected hosts and updating them to GTbots," writes Ferguson this week on Business Computing World.

Ferguson goes on to discuss the fact that early bots were aimed at remote control and information theft, but the move toward modularization and open sourcing lead to a huge increase in variants and the expansion of functionality As we know, Ferguson points out that malware authors gradually started to introduce encryption for ransomware as well as HTTP and SOCKS proxies, allowing them to use their victims for onward connection or FTP servers for storing illegal content.

Spybot in 2003 was an evolution of the earlier SDbot but introduced some important new functionality such as keylogging, data mining, SPIM (Instant Messaging Spam). In the same year we also saw the rise of Rbot which introduced the SOCKS proxy, and included DDoS functionality and information stealing tools.

Rbot was also the first family of bots to use compression and encryption algorithms to try to evade detection. 2003 also saw the first manifestation of a peer-to-peer botnet by the name of Sinit, later on Agobot modules were developed to incorporate this peer-to-peer functionality. The following year another Agobot derivative, known as Polybot introduced polymorphism to try to evade detection by changing its appearance as often as possible

**Botnets of Past, Present and Future**

**a) Botnets of the Past**

In the following section we list down some popular categories of Botnets which have been the major focus of studies and analysis in the past. A very good reference paper that deals with study of attack, control, distribution, deception mechanisms, compares the architecture and presents its findings is. The following Botnets have been studied in the paper in detail.

1. Sdbot

2. Agobot

3. Spybot

4. Rbot

Another Botnet that has been part of popular study was a P2P based botnet known as “Storm”. For more details on this Botnet refer .

**b) Botnets at Present**

In the present, based on various Malware samples that we receive on a daily basis and based on the readings in various blogs, tracking websites and discussion groups we have found the following types and family of botnets in the rise

i. Various IRC based bots

ii. Waledac - It uses HTTP fast-flux proxies to hide the true origin of the command&control (C&C) server. Srizbi botnet

How this botnet had been taken down can be referred here . Botnets that spreading through browsers after exploiting a webserver. This has been well documented in this paper. [“The Ghost in the Browser Analysis of Web-based Malware” .

iii. Botnets that are using DNS based fluxes to connect. Also they use several proxyies with flux based addresses. This makes it difficult to track these botnets with changing addresses.

iv. A recent outbreak reported usage of a MS Widnows vulnerability and a fast spreading Botnet using peer to peer distribution. The static analysis of the Malware payload carried by the botnet distribution phase included algorithms to generate random DNS names and registering the same with popular domain name registrars thereby exhausting the resources of DNS registry system. This bot is known as Conflicker/Downadup.

**c) Future Botnets of**

The botnets of future are envisaged to be more intelligent, using all sorts of means to evade detection and of varying architecture.

The following are some points identified which should be noted while designing botnet detection systems

i. Scalability: Botnets of future are definitely going to be huge in number. Owing to the various kinds of data networks being interconnected like Internet, mobile networks, grid networks, botnets of future will be having very high number of networks of hosts in control and hence the impact will be of a high nature accordingly.

ii. Proprietary protocol based botnets: The botnets will use proprietary protocol to evade detection.

iii. Encrypted communications: The botnets will use encrypted communications for their functionality.

iv. Targeting upcoming Protocols: Botnets will target vulnerabilities and anomalies in upcoming protocols. The same has been demonstrated as part of study performed by ERNW security group. They discuss how the PNRP protocol can be used in future for botnet development.

v. Targeting popular known protocols other than those used in present: These

include protocols like SNMP and other similar protocols which can be targeted eg

SNMP GET flood responses can be invoked to generate botnet based DDoS across number of hosts. Similarly DNS, ICMP, and other protocols can be identified and used.

vi. Botnets of 2009: A good reference paper to study for future botnets is [Spam Botnets to Watch in 2009] which provides list of several Botnets and their sizes, distribution techniques, identification strings, rootkit possibility.

**Different Types of Bots**

During our research, we found many different types of bots in the wild. In this section we present some of the more widespread and well-known bots. We introduce the basic concepts of each piece of malware and furthermore describe some of the features in more detail. In addition, we show several examples of source code from bots and list parts of their command set.

**Agobot/Phatbot/Forbot/XtremBot**

This is probably the best known bot. Currently, the AV vendor Sophos lists more than 500 known different versions of Agobot (Sophos virus analyses) and this number is steadily increasing. The bot itself is written in C++ with cross-platform capabilities and the source code is put under the GPL.

Agobot was written by Ago alias Wonk, a young German man who was arrested in May 2004 for computer crime. The latest available versions of Agobot are written in tidy C++ and show a really high abstract design. The bot is structured in a very modular way, and it is very easy to add commands or scanners for other vulnerabilities: Simply extend the CCommandHandler or CScanner class and add your feature. Agobot uses libpcap (a packet sniffing library) and Perl Compatible Regular Expressions (PCRE) to sniff and sort traffic. Agobot can use NTFS Alternate Data Stream (ADS) and offers Rootkit capabilities like file and process hiding to hide it's own presence on a compromised host.

**SDBot/RBot/UrBot/UrXBot/...**

This family of malware is at the moment the most active one: Sophos lists currently seven derivatives on the "Latest 10 virus alerts". SDBot is written in very poor C and also published under the GPL. It is the father of RBot, RxBot, UrBot, UrXBot, JrBot, .. and probably many more. The source code of this bot is not very well designed or written. Nevertheless, attackers like it, and it is very often used in the wild. It offers similar features to Agobot, although the command set is not as large, nor the implementation as sophisticated.

**mIRC-based Bots - GT-Bots**

We subsume all mIRC-based bots as GT-bots, since there are so many different versions of them that it is hard to get an overview of all forks. mIRC itself is a popular IRC client for Windows. GT is an abbreviation for Global Threat and t0his is the common name used for all mIRC-scripted bots. These bots launch an instance of the mIRC chat-client with a set of scripts and other binaries. One binary you will never miss is a HideWindow executable used to make the mIRC instance unseen by the user. The other binaries are mainly Dynamic Link Libraries (DLLs) linked to mIRC that add some new features the mIRC scripts can use. The mIRC-scripts, often having the extension ".mrc", are used to control the bot. They can access the scanners in the DLLs and take care of further spreading. GT-Bots spread by exploiting weaknesses on remote computers and uploading themselves to compromised hosts (filesize > 1 MB).

Besides these three types of bots which we find on a nearly daily basis, there are also other bots that we see more seldom. Some of these bots offer "nice" features and are worth mentioning here:

**DSNX Bots**

The Dataspy Network X (DSNX) bot is written in C++ and has a convenient plugin interface. An attacker can easily write scanners and spreaders as plugins and extend the bot's features. Again, the code is published under the GPL. This bot has one major disadvantage: the default version does not come with any spreaders. But plugins are available to overcome this gap. Furthermore, plugins that offer services like DDoS-attacks, portscan-interface or hidden HTTP-server are available.

**Q8 Bots**

Q8bot is a very small bot, consisting of only 926 lines of C-code. And it has one additional noteworthiness: It's written for Unix/Linux systems. It implements all common features of a bot: Dynamic updating via HTTP-downloads, various DDoS-attacks (e.g. SYN-flood and UDP-flood), execution of arbitrary commands, and many more. In the version we have captured, spreaders are missing. But presumably versions of this bot exist which also include spreaders.

**kaiten**

This bot lacks a spreader too, and is also written for Unix/Linux systems. The weak user authentication makes it very easy to hijack a botnet running with kaiten. The bot itself consists of just one file. Thus it is very easy to fetch the source code using wget, and compile it on a vulnerable box using a script. Kaiten offers an easy remote shell, so checking for further vulnerabilities to gain privileged access can be done via IRC.

**Perl bots -based**

There are many different version of very simple based on the programming language Perl. These bots are very small and contain in most cases only a few hundred lines of code. They offer only a rudimentary set of commands (most often DDoS-attacks) and are used on Unix-based systems.

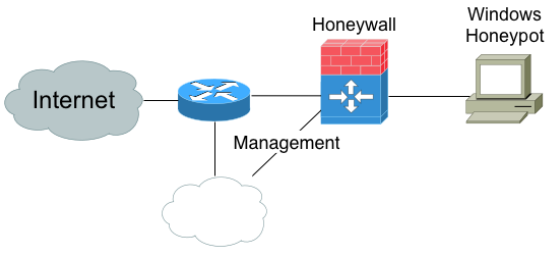
**SOLUTIONS**

Getting information with the help of honeynets

As stated before, we need some sensitive information from each botnet that enables us to place a fake bot into a botnet. The needed information include:

- DNS/IP-address of IRC server and port number
- (optional) password to connect to IRC-server
- Nickname of bot and ident structure
- Channel to join and (optional) channel-password.

Using a GenII Honeynet containing some Windows honeypots and snort_inline enables us to collect this information. We deployed a typical GenII Honeynet with some small modifications as depicted in the next figure:




The Windows honeypot is an unpatched version of Windows 2000 or Windows XP. This system is thus very vulnerable to

attacks and normally it takes only a couple of minutes before it is successfully compromised. It is located within a dial-in network of a German ISP. On average, the expected lifespan of the honeypot is less than ten minutes. After this small amount of time, the honeypot is often successfully exploited by automated malware. The shortest compromise time was only a few seconds: Once we plugged the network cable in, an SDBot compromised the machine via an exploit against TCP port 135 and installed itself on the machine.

As explained in the previous section, a bot tries to connect to an IRC server to obtain further commands once it successfully attacks one of the honeypots. This is where the Honeywall comes into play: Due to the Data Control facilities installed on the Honeywall, it is possible to control the outgoing traffic. We use snort_inline for Data Control and replace all outgoing suspicious connections. A connection is suspicious if it contains typical IRC messages like " 332 ", " TOPIC ", " PRIVMSG " or " NOTICE ". Thus we are able to inhibit the bot from accepting valid commands from the master channel. It can therefore cause no harm to others - we have caught a bot inside our Honeynet. As a side effect, we can also derive all necessary sensitive information for a botnet from the data we have obtained up to that point in time: The Data Capture capability of the Honeywall allows us to determine the DNS/IP-address the bot wants to connect to and also the corresponding port number. In addition, we can derive from the Data Capture logs the nickname and ident information. Also, the server's password, channel name as well as the channel password can be obtained this way. So we have collected all necessary information and the honeypot can catch further malware. Since we do not care about the captured malware for now, we rebuild the honeypots every 24 hours so that we have "clean" systems every day. The German Honeynet Project is also working on another project - to capture the incoming malware and analyzing the payload - but more on this in a later section.

**Botnet Detection and Mitigation**

Botnets use multiple attack vectors; no single technology can provide protection against them. For instance, the goal of a DDoS attack is to cripple a server. The goal of a phishing attack is to lure users to a spoofed Website and get them to reveal personal data. The goal of malware can range from collecting personal data on an infected PC to showing ads on it or sending spam from it. A defense-in-depth approach is essential to detect and mitigate the effects of botnets.

Traditional packet filtering, port-based, and signature-based techniques do not effectively mitigate botnets that dynamically and rapidly modify the exploit code and control channel, resort to "port-hopping" (or using standard HTTP/S ports such as 80 and 443), and shuffle the use of zombie hosts.

A variety of open source and commercial tools are currently used for botnet detection. Many of them analyze traffic flow data reported by routers, such as Cisco® NetFlow. Others use behavioral techniques; for example, building a baseline of a network or system under "normal" conditions and using it to flag abnormal traffic patterns that might indicate a DDoS attack. DNS log analysis and "honeypots" are also used to detect botnets, but these technique are not always scalable.

The most common detection and mitigation techniques include:

• Flow data monitoring: This technique uses flow-based protocols to get summary network and transport-layer information from network devices. Cisco NetFlow is often used by service providers and enterprises to identify command-and-control traffic for compromised workstations or servers that have been subverted and are being remotely controlled as members of botnets used to launch DDoS attacks, perform keystroke logging, and other forms of illicit activity.

• Anomaly detection: While signature-based approaches try to have a signature for every vulnerability, anomaly detection (or behavioral approaches) try to do the opposite. They characterize what normal traffic is like, and then look for deviations. Any burst of scanning activity on the network from zombie machines can be detected and blocked. Anomaly detection can be effectively used on the network as well as on endpoints (such as servers and laptops). On endpoints, suspicious activity and policy violations can be identified and infections prevented.

• DNS log analysis: Botnets often rely on free DNS hosting services to point a subdomain to IRC servers that have been hijacked by the botmaster, and that host the bots and associated exploits. Botnet code often contains hard-coded references to a DNS server, which can be spotted by any DNS log analysis tool. If such services are identified, the entire botnet can be crippled by the DNS server administrator by directing offending subdomains to a dead IP address (a technique known as "null-routing"). While this technique is effective, it is also the hardest to implement since it requires cooperation from third-party hosting providers and name registrars.

• Honeypots: A honeypot is a trap that mimics a legitimate network, resource, or service, but is in fact a self-contained, secure, and monitored area. Its primary goal is to lure and detect malicious attacks and intrusions. Effective more as a surveillance and early warning system, it can also help security researchers understand emerging threats. Due to the difficulty in setup and the active analysis required, the value of honeypots on large-scale networks is rather limited.

**CONCLUSION**

In this paper we have attempted to demonstrate how honeynets can help us understand how botnets work, the threat they pose, and how attackers control them. Our research shows that some attackers are highly skilled and organized, potentially belonging to well organized crime structures. Leveraging the power of several thousand bots, it is viable to take down almost any website or network instantly. Even in unskilled hands, it should be obvious that botnets are a loaded and powerful weapon. Since botnets pose such a powerful threat, we need a variety of mechanisms to counter it.

Decentralized providers like Akamai can offer some redundancy here, but very large botnets can also pose a severe threat even against this redundancy. Taking down of Akamai would impact very large organizations and companies, a presumably high value target for certain organizations or individuals. We are currently not aware of any botnet usage to harm military or government institutions, but time will tell if this persists. In the future, we hope to develop more advanced honeypots that help us to gather information about threats such as botnets. Examples include Client honeypots that actively participate in networks (e.g. by crawling the web, idling in IRC channels, or using P2P-networks) or modify honeypots so that they capture malware and send it to anti-virus vendors for further analysis. Since our current approach focuses on bots that use IRC for C&C, we focused in the paper on IRC-based bots. We have also observed other bots,

but these are rare and currently under development. In a few months/years more and more bots will use non-IRC C&C, potentially decentralized p2p-communication. So more research in this area is needed, attackers don't sleep. As these threats continue to adapt and change, so to must the security community.

**REFERENCES**

[1] Botnets – Wikipedia . (n.d.). Retrieved June 15, 2009, from Wikipedia:http://en.wikipedia.org/wiki/BotnetDunham, K., & Melnick, J. (2008).

[2] Ramneek, Puri (2003-08-08). "Bots &; Botnet: An Overview" (PDF). SANS Institute. http://www.sans.org/reading_room/whitepapers/malicious/bots-botnet-overview_1299. Retrieved 2011-06-21.

[3] "what is a Botnet trojan?". DSL Reports. http://www.dslreports.com/faq/14158. Retrieved 7 April 2011.

[4] Chuck Miller (2008-07-25). "The Rustock botnet spams again". SC Magazine US. http://www.scmagazineus.com/the-rustock-botnet-spams-again/article/112940/. Retrieved 2010-07-30.

[5] "Spam Botnets to Watch in 2009 | Dell SecureWorks". Secureworks.com. http://www.secureworks.com/research/threats/botnets2009/. Retrieved 2012-01-16.

[6] "New Massive Botnet Twice the Size of Storm — Security/Perimeter". DarkReading. http://www.darkreading.com/security/perimeter/showArticle.jhtml?articleID=211201307. Retrieved 2010-07-30.

[7] "Botnet sics zombie soldiers on gimpy websites". The Register. 2008-05-14. http://www.theregister.co.uk/2008/05/14/asprox_attacks_websites/. Retrieved 2011-04-23.

[8] "New Massive Botnet Twice the Size of Storm — Security/Perimeter". DarkReading. http://www.darkreading.com/security/perimeter/showArticle.jhtml?articleID=211201307. Retrieved 2010-07-30.