The botnet business

May 13 2008 | comment

Vitaly Kamluk

This article discusses zombie networks or botnets: how they are created, who uses them to make money on them and how this is done. Readers who are already familiar with the subject will find the information on the latest trends in botnet development of interest.

The article is the first in a series of publications on the botnet problem.

Botnets have been in existence for about 10 years; experts have been warning the public about the threat posed by botnets for more or less the same period. Nevertheless, the scale of the problem caused by botnets is still underrated and many users have little understanding of the real threat posed by zombie networks (that is, until their ISP disconnects them from the Internet, or money is stolen from their credit cards, or their email or IM account is hijacked).

What is a botnet?

First of all, we need to understand what a botnet or zombie network is.

A botnet is a network of computers made up of machines infected with a malicious backdoor program. The backdoor enables cybercriminals to remotely control the infected computers (which may mean controlling an individual machine, some of the computers making up the network or the entire network).

Malicious backdoor programs that are specifically designed for use in creating botnets are called bots.

Botnets have vast computing power. They are used as a powerful cyber weapon and are an effective tool for making money illegally. The owner of a botnet can control the computers which form the network from anywhere in the world – from another city, country or even another continent. Importantly, the Internet is structured in such a way that a botnet can be controlled anonymously.

Computers infected with a bot can be controlled either directly or indirectly. When bots are controlled directly, the cybercriminal establishes a connection with an infected computer and manages it by using commands built into the bot program. In the case of indirect control, the bot connects to the control center or other machines on the network, sends a request and then performs the command which is returned.

The owner of an infected machine usually does not even suspect that the computer is being used by cybercriminals. This is why computers infected with bot malware and which are surreptitiously controlled by cybercriminals are also called zombies. The networks formed from infected machines can be called zombie networks. Most zombie machines are home users' PCs.

How are botnets used?

Botnets can be used by cybercriminals to conduct a wide range of criminal activity, from sending spam to attacking government networks.

Sending spam. This is the most common use for botnets, and is also one of the simplest. Experts estimate that over 80% of spam is sent from zombie computers. It should be noted that spam is not always sent by botnet owners: botnets are often rented by spammers.

It's the spammers who understand the real value of botnets. According to our data, an average spammer makes $50,000 – $100,000 a year. Botnets made up of thousands of computers allow spammers to send millions of messages from infected machines within a very short space of time. In addition to speed and the sheer volume of spam that can be sent, botnets provide spammers with one more advantage. Addresses used to send spam are often blacklisted, and messages coming from these addresses will be blocked or automatically flagged as spam by mail servers. Using hundreds of thousands of email addresses ('borrowed' from the owners of zombie computers) helps spammers overcome this problem.

Another botnet 'bonus' for spammers is the opportunity to harvest email addresses from infected computers. Stolen addresses are sold to spammers or used by the botnet owners themselves to send spam. A growing botnet will add more and more new addresses to the harvest.

Blackmail. The second most popular method of making money via botnets is to use tens or even hundreds of thousands of computers to conduct DDoS (Distributed Denial of Service) attacks. This involves sending a stream of false requests from bot-infected machines to the web server under attack. As a result, the server will be overloaded and consequently unavailable. As a rule, cybercriminals demand payment from the server's owner in return for stopping the attack.

Today, many companies work exclusively on the Internet. Downed servers bring business to a halt,

Archive

<< 2008

Jan Feb Mar
Apr May

Most Popular Analysis

- Online Scanner Top Twenty for April 2008
- Malware evolution: January – March 2008
- Spam evolution: January - March 2008
- Virus Top 20 for April 2008
- Changing threats, changing solutions: A history of viruses and antivirus

For Potential Authors

Want to become one of our authors and see your work published on Viruslist.com? **Contact us**!

resulting in financial losses. To return stability to servers as soon as possible, such companies are more likely to give in to blackmail than ask the police for help. This is exactly what cybercriminals are counting on, and DDoS attacks are becoming increasingly common.

DDoS attacks can also be used as a political tool. In such cases, attacks usually target servers belonging to government organizations. What makes such attacks particularly dangerous is that they can be used as provocation, with a cyber attack on one country being conducted from servers in another country and controlled from a third country.

Anonymous Internet access. Cybercriminals can access web servers using zombie machines and commit cybercrimes such as hacking websites or transferring stolen money. This activity, of course, appears to come from the infected machines.

Selling and leasing botnets. One option for making money illegally using botnets is based on leasing them or selling entire networks. Creating botnets for sale is also a lucrative criminal business.

Phishing. Addresses of phishing pages are often blacklisted soon after they appear. A botnet allows phishers to change the addresses of phishing pages frequently, using infected computers as proxy servers. This helps conceal the real address of the phishers' web server.

Theft of confidential data. This type of criminal activity will probably never lose its attraction for cybercriminals. Botnets help increase the haul of passwords (passwords to email and ICQ accounts, FTP resources, web services etc.) and other confidential user data by a factor of a thousand. A bot used to create a zombie network can download another malicious program, e.g., a password stealing (PSW) Trojan, and infect all the computers on the botnet with it, providing cybercriminals with passwords from all the infected computers. Stolen passwords are sold or used for mass infections of web pages (in the case of FTP account passwords) in order to further spread the bot program and expand the zombie network.

Bot commands

Bots can carry out a wide range of commands, but the most common ones are listed below. Command names can vary from one bot implementation to another, but the functions performed remain the same.

Update: download and launch a designated executable file or module from a specific server. This is a basic command and is the first to be executed. It is used to update a bot's executable file at the command of the zombie network owner if the owner wants to install a new version of the bot program. It can also be used to infect the computer with other malicious programs (such as viruses or worms) and install other bots on the computer. Using this command, PSW Trojans can be installed on all computers that make up the botnet at the same time in order to find all the passwords ever entered on each computer and stored in its memory. The passwords will be sent to a server on the Internet.

Flood: start creating a stream of false requests to a specific Internet server in order to make it fail or to overload channels in a specific segment of the Internet. Such streams can cause servers to malfunction, making them inaccessible to ordinary users. Such attacks using botnets are called DDoS (distributed denial of service). Although there are numerous methods that can be used to create false network requests, describing them in detail is beyond the scope of this article.

Spam: download a spam message template and begin sending spam to designated addresses (each bot is assigned a set of addresses).

Proxy: use the computer as a proxy server. This function is often included in a bot's core functionality rather than being implemented as a separate command. This feature makes it possible to use any computer which is part of a botnet as a proxy server in order to conceal the real address of the cybercriminal controlling the botnet.

Other commands, which are not as popular as those described above, are only implemented in some bots. These additional commands include making screenshots, logging keystrokes, requesting the user's network activity log file (used for stealing accounts and confidential data), sending this file from the user's computer, identifying serial numbers for the software installed on the user's computer, obtaining detailed information about the user's system and network environment, requesting a list of computers included in the botnet, etc.

Types of botnet

Today's botnet classification is relatively simple, and uses botnet architecture the protocols used to control bots as a basis.

Classification of botnets according to architecture

There are currently only two known types of botnet architecture.

1. **Centralized botnets**. In this type of botnet, all computers are connected to a single command-and-control center or C&C. The C&C waits for new bots to connect, registers them in its database, tracks their status and sends them commands selected by the botnet owner from a list of bot commands. All zombie computers in the botnet are visible to the C&C. The zombie network owner needs access to the command and control center to be able to manage a centralized botnet.

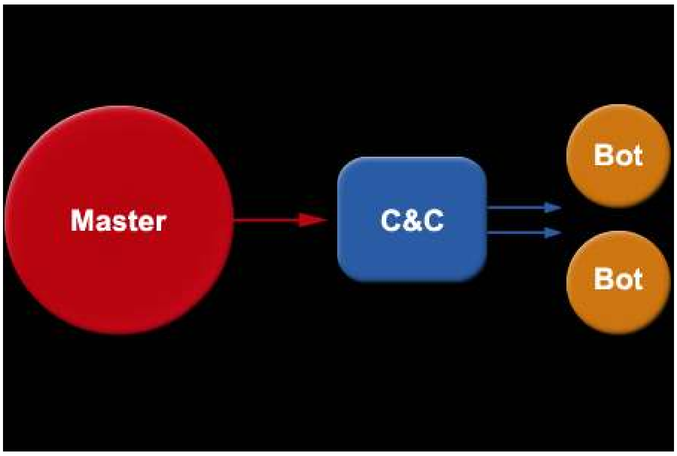


Figure 1. Centralized topology (C&C)

.

Centralized botnets are the most widespread type of zombie network. Such botnets are easier to create, easier to manage and they respond to commands faster. However, it is also easier to combat centralized botnets, since the entire zombie network is neutralized if the C&C is put out of commission.

2. **Decentralized or P2P (peer-to-peer) botnets**. In a decentralized botnet, bots connect to several infected machines on a bot network rather than to a command and control center. Commands are transferred from bot to bot: each bot has a list of several 'neighbors', and any command received by a bot from one of its neighbors will be sent on to the others, further distributing it across the zombie network. In this case, a cybercriminal needs to have access to at least one computer on the zombie network to be able to control the entire botnet.

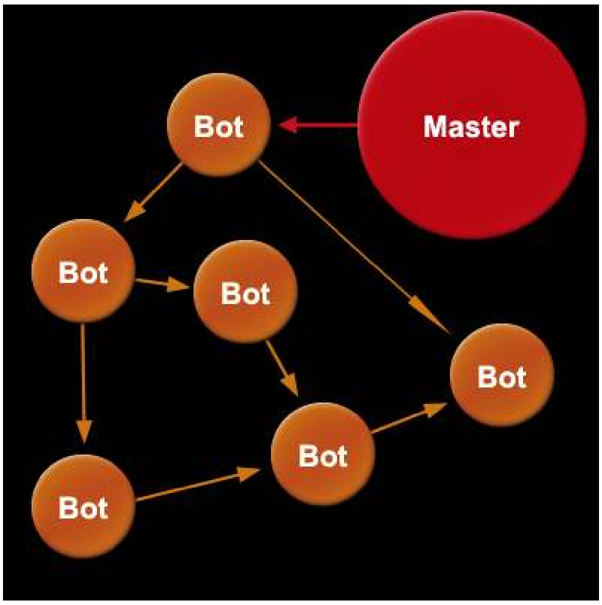


Figure 2. Decentralized topology (P2P)

In practice, building decentralized botnets is not an easy task, since each newly infected computer needs to be provided with a list of bots to which it will connect on the zombie network. It is much easier to direct a bot to a central server first, where it will receive a list of 'neighbor' bots, and only then switch it to P2P connections. This mixed topology is also categorized as P2P, although at a certain stage the bots will use a C&C. Combating decentralized botnets is a much more difficult task than that of combating centralized networks as an active P2P botnet has no control center.

Classification of botnets according to network protocols

For a botnet owner to be able to send commands to a bot, it is essential that a network connection be established between the zombie machine and the computer transmitting commands to it. All network connections are based on protocols that define rules for the interaction between computers on the network. Therefore, botnets can be classified based on the network protocols used.

Botnets can be divided into the following classes when classified according to network protocols:

1. **IRC-oriented**. This is one of the very first types of botnet: bots were controlled via IRC (Internet Relay Chat) channels. Each infected computer connected to the IRC server indicated in the body of the bot program, and waited for commands from its master on a certain channel.
2. **IM-oriented**. This type of botnet is not particularly common. It differs from IRC-oriented botnets only in that it uses communication channels provided by IM (instant messaging) services such as AOL, MSN, ICQ etc. The reason for the relatively low popularity of such botnets lies in the difficulty of creating individual IM accounts for each bot. Bots should be connected to the network and remain online all the time. Since most IM services do not permit logging on to the system from more than one computer at a time while using the same account, each bot needs its own IM account. However, IM services try hard to prevent any

kind of automatic account registration. As a result, owners of IM-oriented botnets only have a limited number of registered IM accounts at their disposal, which limits the number of bots that can be online at any one time. Of course, they can arrange for different bots to share the same account, come online at predefined times, send data to the owner's number and wait for a reply for a limited period of time, but this is inefficient: it takes such networks too long to respond to their masters' commands.

3. **Web-oriented**. This is a relatively new and rapidly evolving type of botnet designed to controlling zombie networks over the World Wide Web. A bot connects to a predefined web server, receives commands from it and transfers data to it in response. Such zombie networks are popular because they are relatively easy to create, there is no shortage of web servers on the Internet and a web interface can be used for easy management.
4. **Other**. In addition to the botnet types listed above, there are other types of botnets that communicate via their own protocol that is only based on the TCP/IP stack, i.e., they only use transport-layer protocols such as TCP, ICMP and UDP.

Botnet evolution

The history of botnets began in 1998 - 1999, when the first backdoor programs – the notorious NetBus and BackOrifice2000 – appeared. These were proof-of-concept Trojans, i.e. programs that implemented completely new technologies. NetBus and BackOrifice2000 were the first to include a complete set of functions that made it possible to remotely administer infected computers, enabling cybercriminals to perform file operations on remote machines, launch new programs, make screenshots, open or close CD-ROM drives, etc.

The backdoors, which are Trojan programs by nature, were designed to work without the user's knowledge or consent. To control an infected computer, a cybercriminal had to establish a connection with each infected machine individually. The first backdoors worked on local area networks based on the TCP/IP protocol stack and demonstrated, in essence, the possibilities to exploit the Windows API in order to control a remote machine.

Even in the early 2000s, remote administration client programs were already able to control several machines at the same time. However, unlike today's backdoors, NetBus and BackOrifice2000 operated as network servers: they opened a predefined port and passively waited for the master to connect (the contemporary backdoors which are used to create botnets establish a connection on their own).

A malicious user then came up with the idea that computers infected with backdoors should establish connections themselves and that they should always be visible online (on the condition that the machine is switched on and working). This user must almost certainly have been a hacker, because new-generation bots employed a communication channel traditionally used by hackers – IRC (Internet Relay Chat). It is also likely that the development of new bots was made easier by the fact that bots working in the IRC system were open source (even though these bots were not designed for remote administration purposes but to respond to user requests such as questions about the weather or when another user had last appeared in chat).

When infecting a computer, the new bots connected to IRC servers on a predefined IRC channel as visitors and waited for messages from the botnet owner. The owner could come online at any time, view the list of bots, send commands to all infected computers at once or send a private message to one infected machine. This was the original mechanism for implementing a centralized botnet, which was later christened C&C (Command & Control Center).

Developing such bots was not difficult because the IRC protocol has simple syntax. A specialized client program is not required to use an IRC server – a universal network client, such as Netcat or Telnet, can be used.

Information about the new IRC botnets spread rapidly. As soon as articles about them began to come out in hacker magazines, a new breed of malicious users appeared: botnet 'hijackers'. These people probably knew as much as botnet owners, but they were after easier money. They looked for IRC channels that had suspiciously large numbers of visitors, entered these channels, studied the botnet and 'hijacked' it. This was done by seizing control of the network, redirecting bots to other, password-protected, IRC channels and the result was full control over somebody else's network of infected machines.

The next stage in the evolution of botnets was marked by moving control centers onto the World Wide Web. First, hackers developed tools for remotely controlling servers based on such popular script engines as Perl and PHP or, more rarely, ASP, JSP and a few others. Then somebody developed a method by which a computer on a local area network could connect to a server on the Internet; this made it possible to control the computer from anywhere in the world. Descriptions of the method for remotely controlling computers on local area networks which bypassed such protection as proxy servers and NAT were published online and it soon became popular in certain circles. Remote administration was based on establishing an HTTP connection with the administration server using the client computer's local settings. If the user configured an address, port, login and password for a proxy server, an authorization mechanism was automatically activated in a dynamic-link library providing HTTP support (Wininet.dll). From a programmer's viewpoint, this was a simple and accessible solution.

The development of semi-legitimate remote administration tools that could be used to evade protection on machines in local area networks and to gain remote access to such computers paved the way for web-oriented botnets. A little later, a simple script was developed for controlling small computer networks and cybercriminals found a way of using such controlled networks for making money.

Web-oriented botnets proved a very convenient solution, which remains popular to this day. A large number of computers can be managed using any device that has Internet access, including a mobile phone that supports WAP/GPRS. And even a child can learn to use a web interface. The further development of the Internet and improved web development technologies were also conducive to the use of web-oriented botnets.

There were also attempts to create botnets controlled via IM services. However, IM botnets never became very widespread, particularly because they require creating IM accounts. It is difficult to register a large number of accounts automatically as systems which protect against automated registrations are constantly modified.

This was not the end of botnet evolution: after trying all available protocols, botnet developers turned their attention to network architecture. It turned out that botnets with classic architecture (i.e. a large number of bots with one command and control center) are very vulnerable, since they depend on a critical node – the command and control center. If this is disabled, control over the entire network will be lost. Models based on simultaneously infecting computers with different bots connecting to different command and control centers sometimes work, but such botnets are much harder to maintain, since two or three C&Cs need to be controlled at the same time.

Experts believe that P2P botnets, which do not have a C&C, could become both highly effective and pose a serious threat. All that the zombie network's owner needs to do is send a command to one of the computers on the network, and the bots will spread the command to other computers in the botnet automatically. In principle, each computer on a botnet can connect to any other computer in the same zombie network. Experiments related to creating such networks have been conducted for quite some time, but the first large botnet using P2P architecture did not appear until 2007. It is P2P botnets that have attracted the most attention, and are the IT security industry's greatest cause for concern.

P2P botnets

The Storm Botnet

In 2007, the attention of security researchers was attracted by a P2P botnet created using a malicious program known as the Storm Worm. Authors of the Storm Worm were spreading their creation so rapidly that it seems as though they had set up a conveyor belt to create new versions of the malicious program. From January 2007 onwards, we have detected between three and five new Storm Worm (Kaspersky Lab classifies it as Email-Worm.Win32.Zhelatin) variants a day.

Some experts believe that the Storm Worm is a malicious program designed to build new-generation zombie networks. Clearly, the bot is being developed and distributed by professionals, and both the zombie network architecture and its defense are well-designed. The following facts bear this out:

- The bot code mutates, making it similar to polymorphic viruses. However, the Strom Worm is different in that the code that conducts the mutation operates on a dedicated computer on the Internet rather than within the program itself (as is the case with polymorphic viruses). This mechanism is called server-side polymorphism.
- Mutation takes place at a relatively high rate (hourly mutation has been recorded) and, importantly, the mutation takes place on the server side, making antivirus database updates ineffective for many users.
- The Storm botnet is protected against the curiosity of security analysts. Many antivirus companies regularly download new copies of the worm from the servers used to spread the malicious program. When frequent requests from the same address are detected, bots receive the command to launch a DDoS attack on that address.
- The bot malware attempts to remain as inconspicuous as possible on infected machines. Obviously, malicious programs that are constantly launching attacks are more easily detected by users and system administrators. Therefore, controlled activity that does not use a large amount of system resources is the safest from a malicious program's point of view.
- Instead of communicating with a central server, the Storm Worm only connects to a small number of 'neighbor' computers on the infected network. This makes identifying all zombie machines on a P2P network practically impossible. The same principle can be used for setting up resistance groups: every member of a group knows only a few other members and the failure of one agent does not mean the failure of the entire group.
- The authors of the worm are constantly changing the methods used to spread it. Originally, the malicious program was distributed as an attachment to spam messages (specifically, in an attachment that looked like PDF files). Later, attachments were replaced with links to infected files inserted into spam messages. There were also attempts to automatically post comments containing links to infected web pages to blogs. Whatever the method used to spread the malicious program, its authors employed sophisticated social engineering techniques.

The Storm botnet has caused numerous problems. Apart from mass-mailing of spam, it is suspected that the botnet was used in a number of large-scale DDoS attacks across the globe. According to some researchers, the Storm botnet was implicated in the cyber attack on Estonia in 2007. The damage such a network could potentially cause makes ISPs and Internet hosting providers uneasy. Adding to the uncertainty is that fact that the size of the Storm botnet remains unknown. While other zombie networks that are fully or partially dependent on a C&C can be seen in their entirety (because the C&C sees each zombie computer connected to the botnet), no expert has seen a list of infected machines which make up the Storm Botnet. Estimates vary, putting the size of the botnet at between 50,000 to 10,000,000 zombie computers.

By the end of 2007, the Storm botnet seemed to have melted away, although we still detect several new versions of the bot every day. Some experts believe that the zombie network was broken up into parts and sold, while others think that the botnet proved unprofitable; the considerable costs of development and support could not be covered by the income it generated.

Mayday

Mayday is another interesting botnet and it technically differs slightly from its forerunners. The bot (Kaspersky Lab classifies it as Backdoor.Win32.Mayday) and the zombie network it creates bear this name as it the word was part of a domain name used by one variant of the malicious program.

Mayday is a botnet based on P2P architecture. After launching, a bot connects to the web server specified in the program's body, registers itself in the server database and receives a list of all bots on the infected computer network (in the case of the Storm Worm, each bot received only a partial list). Then the bot establishes peer-to-peer connections with other bots in the zombie network.

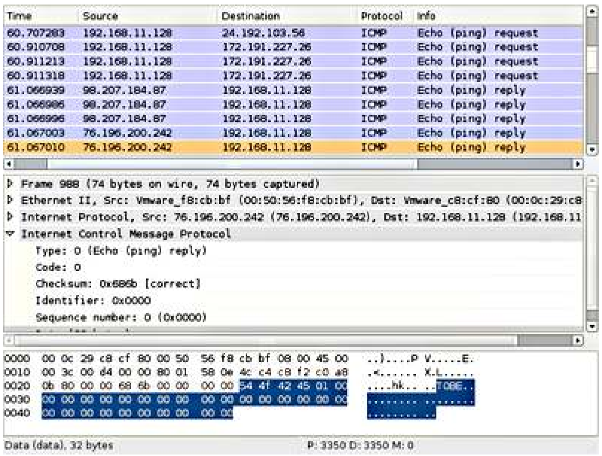
We found six different servers around the world (in the UK, the US, the Netherlands and Germany) which bots connected to when creating the botnet. By early March 2008, only one server was still operational, with about 3,000 bots registered on it (compare this to the Storm botnet, which at the most conservative estimates included tens of thousands of infected computers). Network size is not the only criterion in which Mayday is inferior to its 'big brother' Storm: the Mayday botnet uses a non-encrypted network communication protocol, the malicious code has not been tweaked to hinder analysis by antivirus software and, most importantly, new bot variants are not released with anything nearing the frequency we saw with new variants of the Storm Worm. Backdoor.Win32.Mayday was first detected by Kaspersky Lab in late November 2007, and since then just over 20 different variants of the malicious program have made it into our collection.

As regards new technologies, it is worth noting two non-standard approaches implemented in the botnet.

First of all, the Mayday network uses peer-to-peer (P2P) communication based on ICMP messages with a 32-byte payload.

Most users are familiar with ICMP (Internet Control Message Protocol) because it is used by the PING utility to check whether a network host is accessible. However, the protocol offers a much more extensive range of functions than this. Wikipedia gives the following definition of ICMP: “The Internet Control Message Protocol (ICMP) is one of the core protocols of the Internet protocol suite. It is chiefly used by networked computers' operating systems to send error messages—indicating, for instance, that a requested service is not available or that a host or router could not be reached.”

Figure 3 shows a screenshot of a packet sniffer program that has registered the transmission of ICMP packets from a Mayday bot. None of the bots previously known to us used ICMP to transmit data.



```
Time        Source            Destination       Protocol  Info
60.707283   192.168.11.128    24.192.103.56     ICMP      Echo (ping) request
60.910708   192.168.11.128    172.191.227.26    ICMP      Echo (ping) request
60.911213   192.168.11.128    172.191.227.26    ICMP      Echo (ping) request
60.911318   192.168.11.128    172.191.227.26    ICMP      Echo (ping) request
61.066939   98.207.184.87     192.168.11.128    ICMP      Echo (ping) reply
61.066986   98.207.184.87     192.168.11.128    ICMP      Echo (ping) reply
61.066996   98.207.184.87     192.168.11.128    ICMP      Echo (ping) reply
61.067003   76.196.200.242    192.168.11.128    ICMP      Echo (ping) reply
61.067010   76.196.200.242    192.168.11.128    ICMP      Echo (ping) reply

▷ Frame 988 (74 bytes on wire, 74 bytes captured)
▷ Ethernet II, Src: Vmware_f8:cb:bf (00:50:56:f8:cb:bf), Dst: Vmware_c8:cf:80 (00:0c:29:c8
▷ Internet Protocol, Src: 76.196.200.242 (76.196.200.242), Dst: 192.168.11.128 (192.168.11
▽ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x686b [correct]
    Identifier: 0x0000
    Sequence number: 0 (0x0000)

0000  00 0c 29 c8 cf 80 00 50  56 f8 cb bf 08 00 45 00   ..)....P V.....E.
0010  00 3c 00 d4 00 00 80 01  58 0e 4c c4 c8 f2 c0 a8   .<...... X.L.....
0020  0b 80 00 00 68 6b 00 00  00 00 54 4f 42 45 01 00   ....hk.. ..TOBE..
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0040  00 00 00 00 00 00 00 00  00 00                     ........ ..

Data (data), 32 bytes                         P: 3350 D: 3350 M: 0
```

Figure 3. ICMP packets sent by a Mayday bot

ICMP is used to check bot accessibility on an infected computer network and for bot identification. Since Mayday bots are designed to work under Windows XP SP2, once launched they modify the Windows firewall rules in order to receive ICMP packets.

The second and perhaps most important thing that is different about a Mayday botnet is its command and control center.

Command and control centers of web-oriented botnets use a mechanism known as CGI (Common Gateway Interface). By design, web server technology allows for the use of executable files as an implementation of CGI. Later, a variety of script engines appeared as well. A CGI application generates the content of a web page requested by a user in real time, ensuring execution of the program and displaying the results of its operation instead of static data from the web server. A CGI script works in a similar way, but it needs an interpreter (script engine) to output the results of its operation. As a rule, command and control centers for web-oriented botnets are based on script engines.

In cooperation with law-enforcement agencies, we managed to obtain a copy of the program used in a Mayday C&C. The server-side software of Mayday is a 1.2 megabyte standalone ELF file (the Linux executable file equivalent to Windows EXE files) without any modules. It does not require the system to have a script engine. At first glance, there is nothing strange about the fact that authors of Mayday developed a CGI application instead of a CGI script. However, it does raise a number of questions.

It is far more difficult to develop a CGI application than it is to write a CGI script, because it requires special effort to make the code stable and reliable. Currently, 99% of web development uses script engines, while monolithic CGI executables are developed only when it is absolutely necessary to optimize everything down to the smallest detail. As a rule, this approach is taken by large corporations when developing projects that have to be able to function under huge loads. For example, monolithic CGI programs are used in such web systems as eBay, Paypal, Yahoo! etc.

But why was it necessary to create a monolithic executable file for the Mayday botnet? One possible reason is that the developers wished to make it harder for 'outsiders' to edit, reconfigure and resell a command and control center. Whatever the case, our analysis of the structure of server software used by the Mayday botnet shows that this was a serious development project (the code is tidy, a universal system of classes was devised for the application) that required a well-organized developer team. Moreover, to create software for the Mayday botnet, cybercriminals must have had to work on two projects rather than one, developing software both for Windows and for Linux.

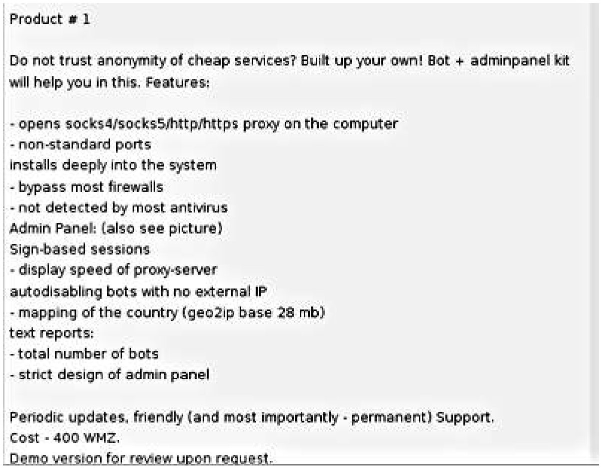
Kaspersky Lab did not detect any new variants of the Mayday bot in spring 2008. Perhaps the malicious program's authors have taken a timeout and the Mayday botnet will resurface in the near future.

The botnet business

The answer to the question why botnets keep evolving and why they are coming to pose an increasingly serious threat lies in the underground market that has sprung up around them. Today, cybercriminals need neither specialized knowledge nor large amounts of money to get access to a botnet. The underground botnet industry provides everyone who wants to use a botnet with everything they need, including software, ready-to-use zombie networks and anonymous hosting services, at low prices.

Let's take a look at the 'dark side' of the Internet and see how the botnet industry works to benefit zombie network owners.

The first thing needed to create a botnet is a bot, i.e. a program that can remotely perform certain actions on a user's computer without the user's knowledge. Software for creating botnets can be easily purchased on the Internet by simply finding a appropriate advertisement and contacting the advertiser.



Product # 1

Do not trust anonymity of cheap services? Built up your own! Bot + adminpanel kit will help you in this. Features:

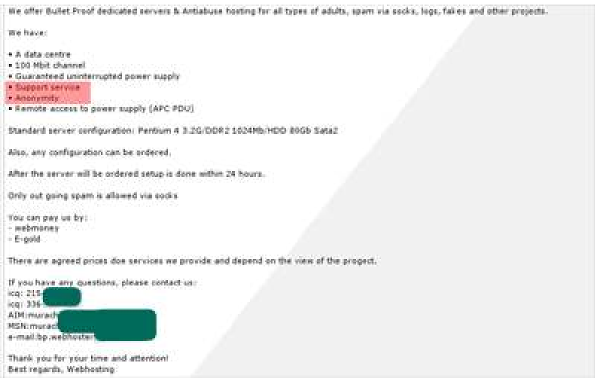
- opens socks4/socks5/http/https proxy on the computer
- non-standard ports
installs deeply into the system
- bypass most firewalls
- not detected by most antivirus
Admin Panel: (also see picture)
Sign-based sessions
- display speed of proxy-server
autodisabling bots with no external IP
- mapping of the country (geo2ip base 28 mb)
text reports:
- total number of bots
- strict design of admin panel

Periodic updates, friendly (and most importantly - permanent) Support.
Cost - 400 WMZ.
Demo version for review upon request.

Figure 4. Advertisement for the sale of a bot and administration panel (translated from Russian)

Bot prices vary from $5 to $1000, depending on how widespread a bot is, whether it is detected by antivirus products, what commands it supports, etc.

A simple web-oriented botnet requires a hosting site where a command and control center can be located. Such sites are readily available, and come complete with support and anonymous access to the server (providers of anonymous hosting services usually guarantee that log files will not be accessible to anybody, including law enforcement agencies). Advertisements like the one shown below are abundant on the Internet.



We offer Bullet Proof dedicated servers & Antiabuse hosting for all types of adults, spam via socks, logs, fakes and other projects.

We have:

• A data centre
• 100 Mbit channel
• Guaranteed uninterrupted power supply
• Support service
• Anonymity
• Remote access to power supply (APC PDU)

Standard server configuration: Pentium 4 3.2G/DDR2 1024Mb/HDD 80Gb Sata2

Also, any configuration can be ordered.

After the server will be ordered setup is done within 24 hours.

Only out going spam is allowed via socks

You can pay us by:
- webmoney
- E-gold

There are agreed prices doe services we provide and depend on the view of the progect.

If you have any questions, please contact us:
icq: 215[redacted]
icq: 336
AIM:murad[redacted]
MSN:murad
e-mail:bp.webhoster

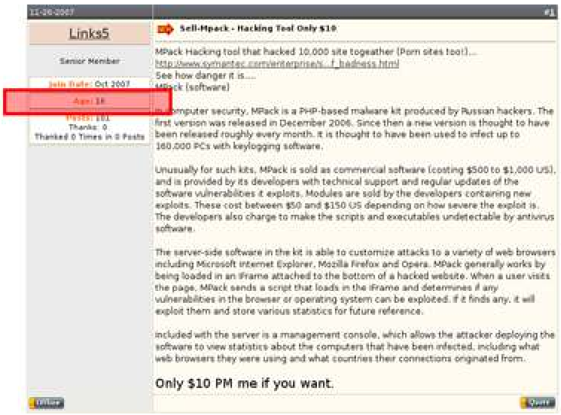
Thank you for your time and attention!
Best regards, Webhosting

Figure 5. Hosting services advertisement for would-be botnet owners

When a C&C site has been created, what's needed next are computers infected by a bot. One option is to buy a ready-made network with somebody else's bot installed. Since stealing botnets is a common practice, most buyers prefer to replace both the malicious programs and the command and control centers with their own, thereby gaining guaranteed control over the botnet. A command will be sent to the bot on the newly-purchased network to download and install a new bot (with a new C&C address) and then self-destruct. This replaces the 'wrong' bots and the botnet begins communicating with the new C&C center. This 'reloading' of botnets is also helpful for protecting them and ensuring anonymity, since IT security experts may already be aware of the 'old' C&C and the 'old' bot.

Unfortunately, building a new botnet is not very hard, either: tools that simplify this task are available. The most popular among them are software packages known as Mpack, Icepack and WebAttacker. They infect the systems of users who visit a malicious web page by exploiting vulnerabilities in browsers or browser plugins. Such software packages are called mass web infection systems or simply ExploitPacks. After an exploit has performed its function, the browser downloads an executable file from the Internet and runs it. The file is a bot program, which adds a new zombie computer to the botnet and gives control over it to the cybercriminal.

Sadly, these tools are so accessible that even adolescents can easily find them and they even try to make money by reselling them.



Links5
Senior Member
Join Date: Oct 2007
Age: 16
Posts: 101
Thanks: 0
Thanked 0 Times in 0 Posts

Sell-Mpack - Hacking Tool Only $10

MPack Hacking tool that hacked 10.000 site togeather (Porn sites too!)...
http://www.symantec.com/enterprise/s...f_badness.html
See how danger it is....
MPack (software)

In computer security, MPack is a PHP-based malware kit produced by Russian hackers. The first version was released in December 2006. Since then a new version is thought to have been released roughly every month. It is thought to have been used to infect up to 160,000 PCs with keylogging software.

Unusually for such kits, MPack is sold as commercial software (costing $500 to $1,000 US), and is provided by its developers with technical support and regular updates of the software vulnerabilities it exploits. Modules are sold by the developers containing new exploits. These cost between $50 and $150 US depending on how severe the exploit is. The developers also charge to make the scripts and executables undetectable by antivirus software.

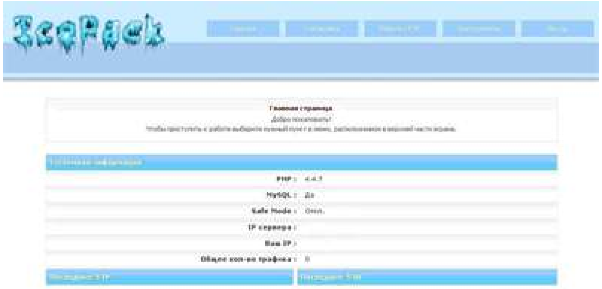
The server-side software in the kit is able to customize attacks to a variety of web browsers including Microsoft Internet Explorer, Mozilla Firefox and Opera. MPack generally works by being loaded in an IFrame attached to the bottom of a hacked website. When a user visits the page, MPack sends a script that loads in the IFrame and determines if any vulnerabilities in the browser or operating system can be exploited. If it finds any, it will exploit them and store various statistics for future reference.

Included with the server is a management console, which allows the attacker deploying the software to view statistics about the computers that have been infected, including what web browsers they were using and what countries their connections originated from.

Only $10 PM me if you want.

Figure 6. A 16-year-old selling Mpack

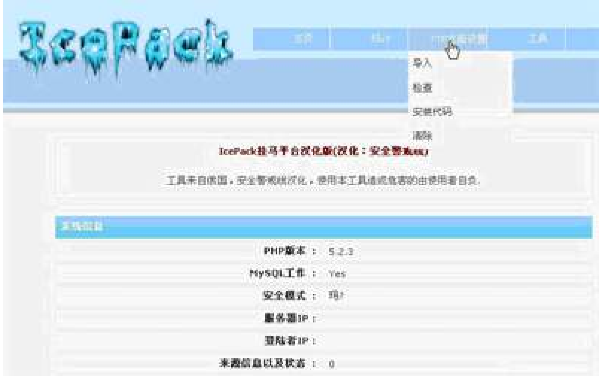
Interestingly, ExploitPacks were originally developed by Russian hackers but later they found an audience in other countries as well. These malicious programs have been localized (showing that they were commercially successful on the black market) and are now actively used in China, among other places.



Главная страница
Добро пожаловать!
Чтобы приступить к работе выберите нужный пункт в меню, расположенном в верхней части экрана.

PHP : 4.4.7
MySQL : Да
Safe Mode : Откл.
IP сервера :
Ваш IP :
Общее кол-во трафика : 0

Figure 7. The original Russian version of IcePack



IcePack挂马平台汉化版(汉化：安全警戒线)

工具来自俄国，安全警戒线汉化，使用本工具造成危害的由使用者自负.

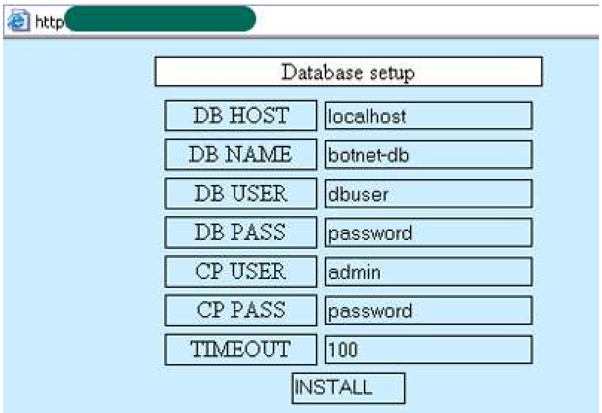
系统信息

PHP版本 : 5.2.3
MySQL工作 : Yes
安全模式 : 玛?
服务器IP :
登陆者IP :
来源信息以及状态 : 0

Figure 8. A localized Chinese version of Icepack

The easier it is to use a system the more popular and successful it is with cybercriminals. Developers of such systems as C&C software or ExploitPacks realize this and develop user-friendly installation and configuration mechanisms for their products in order to make them more popular and increase demand.

For example, installation of a command and control center usually involves copying files onto a web server and using the browser to launch an install.php script. A web interface makes installation much easier: all a cybercriminal needs to do to configure and launch a command and control center is fill in all the web form fields correctly.

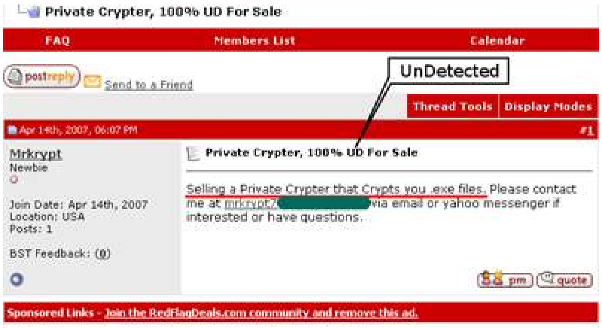

http[redacted]

Database setup

DB HOST	localhost
DB NAME	botnet-db
DB USER	dbuser
DB PASS	password
CP USER	admin
CP PASS	password
TIMEOUT	100

Figure 9. C&C web installer

It is well known in the cybercriminal world that sooner or later antivirus products will start detecting any bot program. When this happens, the infected machines on which an antivirus product is installed are lost to the cybercriminals, while the rate of new infections significantly deteriorates. Botnet owners use a number of methods to retain control of their networks, the most effective of which is protecting malicious programs from detection by processing the malicious code. The black market offers a broad range of services related to the encryption, packing and obfuscation of malicious code.



Private Crypter, 100% UD For Sale

Apr 14th, 2007, 06:07 PM #1

Mrkrypt
Newbie

Join Date: Apr 14th, 2007
Location: USA
Posts: 1

BST Feedback: (0)

Private Crypter, 100% UD For Sale

Selling a Private Crypter that Crypts you .exe files. Please contact me at mrkrypt2[redacted]via email or yahoo messenger if interested or have questions.

Sponsored Links - Join the RedFlagDeals.com community and remove this ad.

Figure 10. An advertisement offering help concealing code from antivirus products

Everything that is needed to successfully run a botnet is available on the Internet. At the moment, it's impossible to arrest the development of the botnet industry.

Conclusion

Today, botnets are among the main sources of illegal income on the Internet and they are powerful weapons in the hands of cybercriminals. It is totally unrealistic to expect that criminals will relinquish such an effective tool. Security experts view the future with some trepidation as they anticipate the continued development of botnet technologies.

What makes botnets increasingly dangerous is that they are becoming easier and easier to use. In the near future, even children will be able to manage them. The ability to gain access to a network of infected computers is determined by the amount of money cybercriminals have at their disposal rather than whether they have specialized knowledge. Additionally, the prices in the well-developed and structured botnet market are relatively low.

It may not only be cybercriminals who have an interest in creating international botnets. Such botnets can be used by governments or individuals to exert political pressure in tense situations. In addition, anonymous control of infected machines that does not depend on their geographic location could be used to provoke cyber conflicts. All this takes is organizing a cyber attack on one country's servers from computers located in another country.

Networks which unite the resources of tens or hundreds of thousands or even millions of infected computers, have the potential to be extremely dangerous – a potential which (luckily!) has not yet been fully exploited. Virtually all this cyber power stems from infected home computers, which make up the overwhelming majority of zombie machines exploited by cybercriminals.

Think of ten friends or acquaintances who have computers – out of the ten, one of them is likely to own a machine that is part of a zombie network. Could that person be you?

Source:
Kaspersky Lab

Email: webmaster@viruslist.com