THORSTEN HOLZ

# spying with bots

Thorsten Holz is a research student at the Laboratory for Dependable Distributed Systems at RWTH Aachen University. He is one of the founders of the German Honeynet Project and has extensive background in the area of honeypots/honeynets and bots/botnets.

*thorsten.holz@mmweg.rwth-aachen.de*

**DURING THE PAST FEW YEARS, WE** have seen a shift in how systems are being attacked. After a successful compromise, a *bot* (also referred to as a *zombie* or *drone*) is often installed on the system. This small program provides a remote control mechanism to command the victim. Via this remote control mechanism, the attacker is able to issue arbitrary commands and thus has complete control over the victim's computer system.

This technique is used by attackers to form networks of compromised machines (so-called *botnets*). With the help of a botnet, attackers can control several hundred or even a thousand bots in parallel, thus enhancing the effectiveness of their attack. In this article, we will discuss concepts behind bots and botnets. We focus on how bots can be used as spyware and provide several examples of this threat. We conclude with an overview of methods to defend against this kind of malware.

The results are based on information we have collected on bots and botnets during the last year as part of our research in the German Honeynet Project. We have published more results in a recent "Know Your Enemy" paper by the Honeynet Project [1].

## Bot and Botnet 101

Historically, the first bots were programs used in Internet Relay Chat (IRC, defined in RFC 2810) networks. IRC, developed in the late 1980s, allows users to talk to each other in IRC channels in real time. Bots offered services to other users, e.g., simple games or message services. But malicious behavior evolved soon and resulted in the so-called IRC wars, one of the first documented distributed denial-of-service (DDoS) attacks. A DDoS attack is a distributed attack on a computer system or network that causes a loss of service to users.

Nowadays, the term *bot* describes a remote-control program loaded ontoå a computer, usually after a successful invasion, which is often used for nefarious purposes. In 2004, bots like Agobot [2], SDBot, and many others were often used in attacks against computer systems. Moreover, several bots can be combined into a botnet, a network of compromised machines that can be remotely controlled by the attacker. Botnets in particular pose a severe threat to the Internet community, since they enable an attacker to control a large number of machines. Attackers prima-

rily use them for attacks against other systems, mass identity theft, or sending spam. A typical setup of a botnet is shown in Figure 1. A central IRC server is used for Command & Control (C&C). Normally attackers use dynamic DNS names for their servers, because it allows a botnet to be distributed across multiple servers. In addition, it allows an attacker to relocate the bots to another server in case one of the C&C servers goes down. In addition to IRC, other communication channels such as HTTP or UDP can be used for C&C.

The bots connect to the server at a predefined port and join a specific channel. The attacker can issue commands in this channel, and these commands are carried out by all bots. In this example, an attacker instructs all bots to propagate further (command advscan) by exploiting the DCOM vulnerability (Microsoft Security Bulletin MS03-026) on TCP port 135. All bots scan with 200 threads in parallel and use a delay of five seconds between their scan attempts. The parameter 0 instructs the bots to propagate forever by scanning their local Class B network (-b) [3].

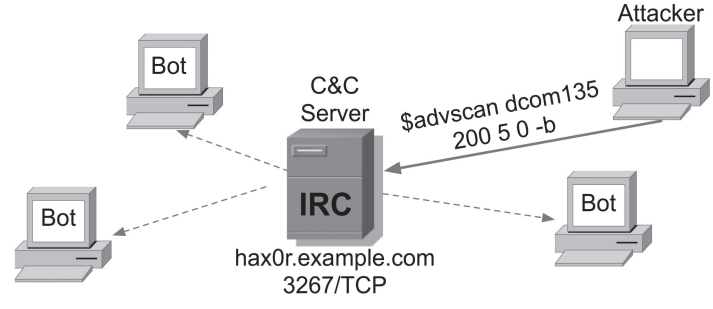


**FIGURE 1: SETUP OF BOTNET USING A CENTRAL IRC SERVER FOR COMMAND & CONTROL**

## Bot Spyware

Spyware has become a major threat in today's Internet. In May 2005, for example, an incident in Israel showed that spyware can be very dangerous. Several large companies in Israel are suspected of having used a malicious program to steal sensitive information from their rivals. In this espionage case, the malicious program was a kind of spyware that is able to retrieve sensitive data (e.g., spreadsheets or screen captures) from the victim's computer. This information is then sent to an FTP server controlled by the attacker and can be used for nefarious purposes. The incident in Israel is just one of many examples of how spyware is used today.

In the following, we will introduce several bots and show how they can be employed to spy on the users of the compromised machines. Our treatment of different bot types is, of course, incomplete, but we discuss the most prevalent usages. In addition to spying, an attacker can issue arbitrary commands, since the vast majority of bots allow an attacker to install arbitrary programs on the victim's computer.

One of the most dangerous bot features is a *keylogger.* With the help of this functionality, an attacker can observe everything the victim is doing. A keylogger can reveal very sensitive information about the victim because she does not suspect that everything she types or clicks is observable by the attacker. Figure 2 shows example output of a keylogger. The attacker can observe that the victim currently uses MSN Messenger, an instant messaging tool. In addition, he observes that the victim is using a search engine.

```
<@controller>    .keylog on
<+[UNC]68395> [KEYLOG]:  (Changed Windows: MSN Messenger)
<+[UNC]68395> [KEYLOG]:hi!(Return) (Changed Windows: Harry )
<+[UNC]68395> [KEYLOG]:  (Changed Windows: Google -Microsoft IE)
<+[UNC]68395> [KEYLOG]:nasa start(Return) (Microsoft IE)
```

**FIGURE 2: EXAMPLE OF KEYLOGGING FEATURE**

Another way to spy on the victim is to grab email addresses or other contact information from the compromised machine. For example, Agobot supports searching for email addresses or AOL contact information on the infected host. Via this spying mechanism, it is possible for an attacker to send customized spam or phishing emails to more victims. More detailed information about the mechanics behind phishing attacks can be found in a recent whitepaper published by the Honeynet Project [4].

Bots often include functions to steal CD-keys from the victim's hard disk. A CD-key is a credential to prove that a specific software has been legally purchased. For example, we found a version of Agobot that is capable of grabbing 26 different CD-keys from a compromised machine, ranging from popular games like Half-Life or Fifa to applications like Windows product IDs. Bots retrieve this information from the Windows registry. They search for characteristic keys and send this data to their controller, as shown in Figure 3. Furthermore, there are several other bots that allow the attacker to read arbitrary registry entries from the victim's computer.

```
<@controller>    .getcdkeys
<+[UNC]75211> Microsoft Windows Product ID CD Key: (XXX).
<+[UNC]75211> [CDKEYS]: Search completed.
<+[UNC]00374> Microsoft Windows Product ID CD Key: (XXX).
<+[UNC]00374> [CDKEYS]: Search completed.
```

**FIGURE 3: EXAMPLE OF AN ATTACK THAT STEALS CD-KEYS FROM COMPROMISED MACHINES**

Another basic spy-functionality is stealing information about the victim's host, such as the speed of the CPU, the uptime, and IP address. For example, SDBot provides the attacker with several facts about the compromised host. Figure 4 shows the output of the two commands sysinfo and netinfo. We see that an attacker gets an overview of the hardware configuration and the network connectivity. Similarly, 4x10m, a rather uncommon bot, implements several functions to retrieve the registered owner and company of the compromised machine. This kind of information is especially interesting if the attacker plans to sell or rent his bots to others.

```
<@controller>    .sysinfo
<DE|924621> cpu: 1200MHz. ram: 523744KB total, 139206KB free.
        os: Windows XP (5.1, build 2600). uptime: 0d 1h 17m
<@controller>    .netinfo
<DE|924621> connection type: dial-up (MSN). IP Address: X.X.X.X
                    connected from: aaa.bbb.ccc.ddd
```

**FIGURE 4: EXAMPLE OF AN ATTACK THAT RETRIEVES INFORMATION ABOUT THE VICTIM**

Many bots also include functions to search the hard drive of all victims for sensitive files, based on a regular expression. Moreover, these bots implement functions to download these files from the victim's computer. As an example, we take a look at a bot called reverb. This bot implements a function called weedfind that can be used to retrieve information. An example is the command .weedfind c:*.xls or c:*finance*. This command lists all Excel spreadsheets and all files which contain the string finance on compromised machines.

Spybot, a quite popular bot nowadays, implements several methods to retrieve sensitive information from a victim. An analysis revealed that this specific spyware implements at least 10 functions that can be used for spying purposes. Besides functions to retrieve a file listing and retrieve files, this bot also implements a function to delete files.

In addition, Spybot offers a method to log keystrokes on the victim's machine. To achieve this, two functions are implemented: startkeylogger is used to start the logging of keystrokes and stopkeylogger to stop this function. The logged keystrokes are sent directly to the attacker. Moreover, keystrokes can also be sent to the victim's computer and, thus, arbitrary key-sequences can be simulated with the help of the sendkeys [keys] command. Spybot also implements functions that return information about the running processes: with the function listprocesses, a listing of all running processes can be retrieved and killprocess [processname] can then be used to stop processes on the victim's machine, e.g., an antivirus scanner or some kind of personal firewall. Our analysis revealed two additional functions to retrieve sensitive information from the victim's machines. First, the command passwords lists the Remote Access Service (RAS) password from computers running Windows. Second, the command cachedpasswords lists all passwords that are returned by the Windows API function WNetEnumCachedPasswords(). Table 1 gives a short summary of all functions from Spybot that are spyware-related, including examples of how an attacker could use these commands to retrieve sensitive information.

| Command | Action / Example |
|---|---|
| list [path+filter] | example: list c:*.ini |
| delete [filename] | example: delete c:\windows\netstat.exe |
| get [filename] | send specified file to attacker |
| startkeylogger | starts online-keylogger |
| stopkeylogger | stops the keylogger |
| sendkeys [keys] | simulates keypresses |
| listprocesses | lists all running processes |
| killprocess [processname] | example: killprocess taskmgr.exe |
| passwords | lists the RAS passwords in Windows 9x |
| cachedpasswords | get WNetEnumCachedPasswords |

**TABLE 1: SUMMARY OF SPYWARE-RELATED OPTIONS IN SPYBOT**

## Defending Against Bots

After presenting the wide spectrum of possible usage of bots as spyware, we now want to present several ways to stop this threat. This should help to get an overview of possible methods to detect the presence of bots and also to detect the existence of communication channels used for C&C.

Currently, the most effective method to stop bots is to stop the initial establishment of a connection from a bot to the C&C server. As explained above, most bots use a central server for C&C, and, in most cases, a dynamic DNS name is used for this server. This allows us to stop a botnet effectively. Once we know this DNS name, we can contact the DNS provider and ask for help. Since many DNS providers do not tolerate abuse of their service, they are also interested in stopping the attack. The DNS provider can easily "blackhole" the dynamic DNS name, i.e., set it to an IP address in the private range as defined in RFC 1918. If

an infected machine then tries to contact the C&C server, the DNS name will resolve to a private IP address and thus the bot will not be able to contact the C&C server. This method is mostly used by CERTs and similar organizations and has proved to be quite effective; many communication channels have been disrupted in this way. Nevertheless, it requires the DNS provider's cooperation and this is not always obtainable.

There are also several methods to stop a bot within a network that can be carried out by a network administrator or security engineer. We will introduce several methods in what follows. As always, the best way to cancel a threat is to stop its root cause. In this case, this would mean eliminating the attack vectors and checking for signs of intrusions, e.g., by patching all machines and keeping AV signatures up-to-date. But this is often difficult: a zero-day exploit, i.e., an exploit that has no available patch, cannot be eliminated in all cases, and patching needs some testing since it could break important systems. In addition, AV scanners often cannot identify targeted attacks. With the recent bot Zotob, the time between a proof-of-concept exploit for a new security vulnerability and the integration of it into a bot can be as little as several hours or days, so patching cannot always help; nevertheless, it is still important to try to keep patches as up to date as possible.

One quite effective method to detect the presence of bots also exploits their rather noisy nature. Most bots try to spread by exploiting security flaws on other systems. To find such a system, they have to extensively scan the network for other machines. In addition, the communication channel often uses specific, rather unusual ports. So by looking at the state of your network, you can often detect bots. Netflow/cflow is an easy-to-use solution for this problem, in which the collected data often allows you to spot an infected machine. A typical sign is a spike in the number of outgoing connections, most often on TCP ports 445 and 135, or on ports with recent security vulnerabilities, caused by bots that try to propagate via common vulnerabilities. Another sign is a high amount of traffic on rather unusual ports. We analyzed the information about more than 11,000 botnets and found out that the vast majority of botnets use TCP port 6667 for C&C. Other commonly used ports include TCP ports 7000, 3267, 5555, 4367, and 80. TCP port 6667 is commonly used for IRC, and of course 80 for HTTP, but you should take a look at these and the others mentioned. In addition, tools like ngrep or snort can help to detect the presence of C&C channels and typical C&C messages. This can, for example, be done with the following regular expression [5]:

```
(advscan|asc|xscan|xploit|adv\.start|adv5c4n) (webdav|netbios|\
ntpass|dcom(2|135|445|1025)|mssql|lsass|optix|upnp|ndcass|imail)
```

Of course, such a method requires some human supervision, since it is not error-free and could lead to false positives. In addition, the C&C commands can change with time, and thus regular updates are necessary.

We are currently also exploring other mechanisms to stop or observe botnets; for example, we introduced a methodology to infiltrate remote control networks to learn more about them [6]. This method is based on the usage of honeypots [7]: we use these tools to actually capture a binary, and an analysis of it leads to all of the botnet's sensitive information (e.g., DNS name, port, passwords). By smuggling a fake bot into the botnet we can learn more about the actual botnet and the tactics of the attackers.

A similar approach uses specialized honeypots like mwcollect (http://mwcollect.org) or nepenthes (http://www.nepenthes.it). Both tools are capable of collecting malware in an automated way and work with the same basic principle: they simulate a known vulnerability and wait to be exploited. Once the tool detects an exploitation attempt, it triggers the incoming exploit and analyzes the incoming payload. This analysis leads to much more information, which can be com-

bined to download the malware from another computer system. Thus we are able to download malware that tries to propagate in an automated way. Once we have downloaded a binary, we can analyze it and extract more information regarding the botnet. We can use this information to stop the bot from spreading within the local network, e.g., by stopping all network connections to the C&C server or by searching for the bot on all machines. This approach is currently in development, but preliminary results look promising.

## Conclusion

Currently, bots pose a threat to individuals and corporate environments. They are often used for DDoS attacks, for sending spam, and as spyware to steal sensitive information from the victim's machine. Since an attacker can install programs of his choice on the compromised machines, his proceedings are unpredictable.

There are several ways to defend networks and computer systems against this threat. The methods either try to proactively disrupt the communication flow between bots and the C&C server or to detect signs of a successful invasion.

More research is needed in this area: current botnets are rather easy to stop due to their central C&C server. But in the future, we expect other communication channels to be deployed, especially peer-to-peer-based C&C communication. With Sinit we have seen the first bot that uses such communication channels [8], but presumably the future will bring much more of this type of malware.

### ACKNOWLEDGMENTS

This paper was a result of the research carried out by members of the Honeynet Research Alliance, especially members of the German Honeynet Project. Special thanks go to Julian Grizzard, Chris Lee, David Dittrich, and Niels Provos for helpful comments on previous versions of this paper. I would also like to thank the Deutsche Forschungsgemeinschaft (DFG), who supported my work as part of the graduate school work, "Software for Mobile Communication Systems," at RWTH Aachen University.

### REFERENCES

[1] The Honeynet Project, "Know Your Enemy: Tracking Botnets," March 2005: http://www.honeynet.org/papers/bots/.

[2] LURHQ Threat Intelligence Group, "Phatbot Trojan Analysis," 2004: http://www.lurhq.com/phatbot.html.

[3] A more detailed introduction to bots, including a classification and several examples, can be found in Thorsten Holz, "A Short Visit to the Bot Zoo," *IEEE Security & Privacy,* vol. 3, no. 3 (2005), pp. 76–79.

[4] The Honeynet Project, "Know Your Enemy: Phishing," May 2005: http://www.honeynet.org/papers/phishing/.

[5] Tom Fischer, "Botnetze," *Proceedings of 12th DFN-CERT Workshop,* March 2005.

[6] Felix Freiling, Thorsten Holz, and Georg Wicherski, "Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks," *Proceedings of the 10th European Symposium on Research in Computer Security (ESORICS05)*, Milan, Italy, September 12–14, 2005 (Springer, 2005).

[7] The Honeynet Project, "Know Your Enemy: GenII Honeynets," November 2003: http://www.honeynet.org/papers/gen2/.

[8] LURHQ Threat Intelligence Group, "Sinit P2P Trojan Analysis," 2003: http://www.lurhq.com/sinit.html.