## HotBots '07: First Workshop on Hot Topics in Understanding Botnets

*Cambridge, MA*
*April 10, 2007*
*Summarized by Rik Farrow, with help from Dan Geer*

HotBots 2007 focused on understanding the current state of Botnets, and this workshop delivered what it promised. The workshop PC read 32 papers and accepted 11 for the workshop. The papers varied from detailed analysis of dissected bots to theoretical notions of potential bots. I particularly liked the talks that focused on reality, for example, on the trend toward peer-to-peer (P2P) botnets and on the research into the sizes and frequency of botnets in the wild.

You can read the papers included in this workshop at http://www.usenix.org/events/hotbots07/.

### PEER-TO-PEER

■ *Peer-to-Peer Botnets: Overview and Case Study*

*Julian B. Grizzard, The Johns Hopkins University; Vikram Sharma, Chris Nunnery, and Brent ByungHoon Kang, University of North Carolina at Charlotte; David Dagon, Georgia Institute of Technology*

Julian Grizzard began the day with this paper about P2P botnets. Earlier botnets relied on IRC for Command and Control (C&C), a feature that made bots easier to detect. An ngrep of network traffic can turn up IRC commands,

and as IRC has declined in popularity relative to IM, this can be a dead giveaway. Moving to P2P does make bots harder to detect, but it also means that the bot-herder loses fine-grained control over his or her botnet. With IRC C&C, the bot-herder can issue commands to all the bots currently connected to an IRC channel (or some subset of those connected) and have them carried out immediately. With P2P used as C&C, issuing commands becomes a process of sharing files, which may contain commands or an entirely new executable, a process that takes time. The bot-herder no longer can launch a devastating DDoS on a moment's notice, or fire orchestrated salvos of packets from different botnets to make detection of the DDoS sources more difficult. As it turns out, these uses of botnets have declined in importance anyway, a point made several times throughout the workshop's presentations.

P2P botnets do gain something else by giving up IRC for C&C, and that is robustness. Losing control of the C&C IRC server once meant losing potentially thousands of bots. With P2P botnets, there is no centralized point of control. Not only does this protect the botnet owner's investment but it makes bots more difficult to detect and the entire botnet more resilient to countermeasures.

Grizzard and his fellow researchers analyzed captured software, Trojan.Peacomm, a P2P bot. Peacomm uses the Overnet protocol (once used for P2P filesharing), a distributed hash table based on the Kademlia algorithm. The initial trojan infection, via spam, uses a promise of displaying a movie as an incentive and results in the bot joining a P2P network and downloading subsequent versions (secondary downloads) from other members of the botnet. Encrypted URLs are distributed using Overnet, decrypted using a static key, and new executables are downloaded using HTTP. This general technique, multiple staged infection with downloads over HTTP, seems common, based on later presentations. Network traces of the captured bot revealed a list of 10,105 unique IP addresses in packets received from Overnet hash table lookups, but the bot only contacted 4,200 during the time monitored.

During the Q&A session, Dan Geer asked, "Why not poison the distribution network if the encryption key is known?" Grizzard answered that the legal panel would have an answer to that in the afternoon. Nicholas Ianelli, a member of the CERT technical staff, mentioned that the key is known, and the Overnet packet used includes a unique meta-ID field that makes these packets easier to recognize. Someone else mentioned that more sophisticated packers are being used to make reverse engineering of malware more difficult.

Grizzard ended by pointing out that the secondary injection downloaded other specific malware, including rootkits, spam relays, email address harvesters, the trojan propagator, and a DDoS agent.

***An Advanced Hybrid Peer-to-Peer Botnet***

*Ping Wang, Sherri Sparks, and Cliff C. Zou, University of Central Florida*

Cliff Zou described hypothetical hybrid botnets. In this research, his group designed an advanced P2P botnet that would be harder to shut down, monitor, or hijack. Their design has two classes of bots, servants and clients. Servants have fixed IP addresses and act as the C&C network for the botnet. Clients get their commands and new versions of malware from the servants. Any bot can act as a sensor host and have performance information sent to the sensor host from clients. The sensor host then reports to a servant. Clients contain a fixed list of servant addresses, but no client has all servant addresses.

In analysis, removing 80% of servants bots leaves 95% of client bots connected, so this type of botnet would be very resilient. Possible defenses include poisoning servants using honeypots or capturing servants early in the infection process. Someone asked whether it was wise to publish research like this, and Zou responded that they felt it was appropriate to be forward-looking about new styles of attacks.

■ ***A Distributed Content Independent Method for Spam Detection***

*Alex Brodsky, University of Winnipeg; Dmitry Brodsky, Microsoft Corporation*

Alex Brodsky presented Trinity, a distributed database designed to collect source addresses of spam relays while remaining resilient to attacks. Brodsky pointed out that using blacklists has become less effective over time, as bots provide spam relays that often route spam via an ISP's mail gateway, as well as sending a relatively small number of emails. Trinity uses a SpamAssassin plug-in to capture a spam sender's email address by parsing Received headers email envelopes that have been classified as spam and sent to Trinity servers. The servers are selected to store spam relay IP addresses using the DHT hash, so no single server holds all the addresses. Servers also replicate the addresses received, so one or more servers can be lost without the service being disabled. Trinity also uses reputation scoring by collecting answers from different peers, to guard against spammers running a server and poisoning the database.

Trinity is under development.

## MEASUREMENT

■ ***The Ghost in the Browser: Analysis of Web-based Malware***

*Niels Provos, Dean McNamee, Panayiotis Mavrommatis, Ke Wang, and Nagendra Modadugu, Google, Inc.*

Niels Provos described the results of a project he started at Google in 2006. Niels had become increasingly aware that browser vulnerabilities provide fertile ground for the in-

stallation of malware. In what he terms "drive-by downloads," Niels tells how Web sites with imperfect security wind up with modified pages. The modifications include IFRAMES or JavaScript that causes the browser to download a first-stage exploit. The first stage then downloads the real malware, which will be spyware for financial information, spam relays, bots, and tools for actively attacking systems.

Google already trawls the Web and caches pages. What the Google team has done is build a system that uses heuristics to decide whether a page may be malicious. The suspect pages get loaded into an instrumented version of Internet Explorer (IE) running within a VM-encapsulated copy of Windows. Network activity (downloading malware), changes to the browser state or registry keys, and software installs that occur after loading the suspected page in IE determine whether the page really does result in a drive-by download. Google then marks pages determined to result in the downloading of malware by returning a warning page as a search result. The current setup scans up to 500,000 pages a day, and under 0.5% of all pages scanned have been infected.

Besides compromising Web servers to install the download code, Niels said that other methods of serving up code included delegated (subsyndicated) sponsored ads and third-party widgets, such as the Preying Mantis counter that started delivering malware to visitors instead of being "just a counter." Popular sites were less likely to be vandalized as these sites are better maintained, and large sites were much faster at responding to reports of infected Web pages. Niels also mentioned that sometimes a server hosting many Web sites will be compromised, and pages for all the hosted sites will then be infected.

Fabian Monrose (Johns Hopkins) asked how Google handles infected pages. Niels explained that Google serves up a warning page. Earlier versions of this page allowed users to click through to the infected site, and 30–40% of users did so. Someone else asked whether exploiters are using Google Analytics, and Niels said he wasn't sure, but he expected that some exploiters were using it to keep track of the number of sites they had exploited. In response to Dan Geer's question whether the robots.txt file is honored during this process, Niels answered that Google follows the standard for crawling Web pages and ignores those. Fabian asked if Google is censoring the Web, and Niels replied that they are observing that some Web pages exploit your browser. You can still cut-and-paste your way past the warning.

■ ***My Botnet Is Bigger Than Yours (Maybe, Better Than Yours): Why Size Estimates Remain Challenging***

*Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis, Johns Hopkins University*

Andreas Terzis started by saying that botnet sizes vary according to how and when they are measured. Some researchers have found botnets with 350,000 members, whereas others report that botnets rarely exceed a few thousand bots. Andreas suggested several metrics for measuring, including descriptions of how the botnets were measured. For example, the infection footprint will be the largest size, as it represents all systems detected as infected, but effective size represents the number of bots available at one time. The difference between these two measurements can easily be one or more orders of magnitude, for example, 45k infected but only 3k active.

There are certainly difficulties in measuring botnets. A total of 48% of IRC servers used for bot-herding block join messages. Another counting technique involves querying DNS servers for cached copies of bot server addresses, which provides a list of domains that have at least one infection. Another issue has to do with overlap between botnets and their owners. The authors determined that 25% of 472 botnets that they tracked were associated into only 90 groups. Some of the overlap occurs because bots can be commanded to clone themselves, instantly creating a related botnet.

■ ***Toward Botnet Mesocosms***

*Paul Barford and Mike Blodgett, University of Wisconsin—Madison*

Mike Blodget described the Botnet Evaluation Environment (BEE), a testbed for experimenting in a secure and flexible fashion with botnets that is Emulab-enabled. Mike explained that the attraction of botnets to organized crime makes study of this phenomena an important area of research. Bots and botnets are also growing in complexity as well as in their resistance to dissection. It is common to find malware that is packed (obfuscated), can detect whether it is being run in debugger mode, or is run within a VM. Techniques used for determining whether a VM is present include checking the interrupt vector and looking for VMWare tools.

BEE provides support for bots and the services they require, such as DHCP, DynDNS, and IRC. The authors are building a library of OS/bot images using both bots built from source and bot binaries. For security, they block all UDP traffic from BEE, use an unroutable ten-net within BEE, and firewall all traffic between the test network and the experimenters' network.

## DETECTION, RESPONSE, AND ANALYSIS

■ ***Wide-Scale Botnet Detection and Characterization***

*Anestis Karasaridis, Brian Rexroad, and David Hoeflin, AT&T Labs*

Anestis Karasandridis described how his group detects bots by analyzing flow logs. As part of AT&T security efforts, they have been analyzing flow logs for evidence of bots and C&C servers. AT&T runs a Tier 1 network, what

I once would have called an Internet backbone, and collects *8 to 10 billion flow logs per hour*. They use triggers, such as hosts that scan, relay spam, or attack other systems, then apply heuristics, described in their paper, to whittle down the number of flows. Some heuristics are obvious, such as selecting flows to the common IRC ports. Others are much more sophisticated, for example, looking for a pattern of flows that would match the PONG response from bots to the C&C server within a particular time window.

Using this and other techniques, the group detected 376 unique C&C server IP addresses between August 2006 and February 2007. During the same period, they discovered 6 million bots, and they continue to find about 1 million bots per month. They tested the accuracy of their data by contacting other ISPs or looking within the data, measuring a false positive level of less than 2%.

Nicholas Ianelli (CERT) asked about the percentage of commands that were encrypted. Brian Rexroad answered that perhaps 5% use some channel encryption or obfuscation. Someone else asked what other means were used to verify correctness, and Brian answered that they do perform some packet capture of suspected bots. Another question concerned the high number of bots seen, and the answer was that because unique IP addresses are counted, dynamic addresses can affect this. In the paper, they mentioned that there is considerable churn, with bots changing channels or servers every 3-4 days. Another person asked about using their algorithms on Arbor Network boxes, but the authors didn't know whether this would work. Brian did make a comment that games can attempt many connections to servers and fail, making them appear like scanners, and thus bot clients. Finally, Niels Provos asked if they had looked at anyone else's Netflow data. Andreas said they are using such data to protect their customers (which are other ISPs). Niels then inquired, "If we asked politely, might we exchange info?" Brian answered that it might be possible.

■ ***Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation***

*Jan Goebel, RWTH Aachen University, Germany; Thorsten Holz, University of Mannheim, Germany*

Thorsten explained how his group had created a simple script that looks at communication channels for common IRC commands and watches for patterns in bot nicks. Nicks are used in the IRC protocol to identify a client connection to an IRC channel, and each must be unique per channel. Rishi uses ngrep to collect lines and a 1700-line Python script for analysis. Vern Paxton asked whether you could use Bro, and Thorsten answered that this is a prototype, but perhaps.

Liss asked whether botnet owners actually use nicks to partition their networks. Thorsten did notice checks for .edu hosts and for country domain, so they can group commands (.edu hosts will have more bandwidth). Nicholas Ianelli says he has seen partitioning in nicks, such as *p* for private network. Someone asked about using machine learning techniques, which had been shown in one of the slides, and Thorsten confirmed that they would work. Another person asked how the method compared to other methods. The answer was that it did better than Nepenthes (see "Advanced Honeypot-Based Intrusion Detection" in the December 2006 *;login:*). Niels Provos asked whether this assumes that IRC is still being used. Thorsten said that gamers use IRC as well as members of the whole underground economy. In response to Niels's question "If people start using dictionaries, can you use other events?" the answer was that indeed heuristics could help here.

■ ***AS-Based Accountability as a Cost-Effective DDoS Defense***

*Daniel R. Simon, Sharad Agarwal, and David A. Maltz, Microsoft Research*

Fabian Monrose, the session chair, introduced this as "the evil bit talk" and we soon discovered why. Dave Maltz provided motivation for a DDoS defense by pointing out that 4000 bots can overload a 4-Gbps link ($25,000 to $50,000 per month) for about $1,600 a month cost for the attacker. To protect against a 50,000-bot network costs $10 million over three years, to pay for 3 OC48s from a provider for $210,000/month. Other solutions include Content Distribution Networks (CDNs) such as Akamai and are very expensive.

The proposed solution is an architectural change, involving just software and not hardware, to identify trusted sources with a persistent attribute, and then receivers can blackhole sources using whatever method they want. The solution assumes pairwise trust among Autonomous Systems (AS). Each ISP modifies customer relationship software (used for billing customers), keeps track of their IP addresses, and must be willing to install filters on particular source-address pairs (you can only filter sources that target your destination). The Filter Request Server, FRS, forwards data to your router and is used to install this filter.

Border routers at the edges of accountable networks add the "evil bit" to packets (Bellovin's evil bit, or Crocker's well-maintained bit, a.k.a. the anti-evil bit). This bit indicates that marked packets came from a trusted source.

Rik Farrow leaped to his feet and provided several comments: You can set the evil bit on packets coming from any AS you don't like; you can clear all evil bits because you are opposed to this solution; trust between ASes may not (probably does not) exist (since these are often competitors); and ISPs can slap the evil bit on repeat offenders instead of installing filters on their own routers. Someone else mentioned that FRS could be used to overwhelm the ability of routers (a DoS by overusing the filtering function of routers). Dave still maintained that this is a cheap, on-the-fly reputation system. Someone else suggested that it

would be easy to frame opponents, while a final suggestion was to check Netflow logs to determine guilt at the user's ingress point.

## CASE STUDIES

■ ***Panel: Legal Issues About Botnet Tracking and Response***

*Panelists: Jon L. Praed, Internet Law Group; Jody R. Westby, Global Cyber Risk, Adjunct Distinguished Fellow to Carnegie Mellon CyLab; David Dagon, Georgia Institute of Technology; Alexander Muentz, OnSite E-Discovery*

Dagon started with a scenario where a student accidentally starts to proxy all his IP traffic from his dorm room via your honeynet, and you capture data such as his social security number and email to his doctor, and to his lawyer, who happens to be in Europe where privacy laws are stronger. Dagon pointed out that there are many laws that could be applied here that offer protection of the student's privacy: FERPA, HIPPA, and GLBA. Also, nearly every university has a policy pertaining to Human Subjects that can also affect privacy. The best approach is to have a clear policy that addresses privacy issues.

Dagon also suggested that you operationalize your research, that is, make it useful to the organization, to gain allies in operations. You should also sandbox your investigations, using a Truman network.

Alex Muentz described the applicable U.S. federal laws. The Computer Fraud and Abuse Act, 18 USC 1030, provides you with a great deal of protection when you act as a provider. Working within a sandbox does not give you that protection. The Stored Communications Act, 18 USC 2701, and Electronic Communications Privacy Act/Wiretap Act, 18 USC 2511, are similar to 18 USC 1030, so don't sandbox for maximum protection, but act as provider working to maintain the network or system.

John Praed's group actually works as private investigators, trying to track down wrong-doers using log and other data to do so. He suggested that even working as a "white hat" can get you in trouble via overly aggressive conduct. He mentioned winding up in trouble because of an ambitious local DA who wants to be governor. He also pointed out that Putin in Russia could well act aggressively to protect "businessmen" in Russia who look like "black hats" to us. Praed also suggested moving cautiously in government investigations, as illegal search can cause evidence to be tossed out. There are also cases of "black hats" suing the "good guys," as in spammers suing SpamHaus to be removed from the blackhole list.

Praed said that in the future, the primary economic driver for criminals will be cyberextortion over hard drive contents and captured keystrokes. Already, he has seen examples of blackmail backed up with physical threats.

Jody Westby spoke quickly as time was running out for this session. She discussed procedural and practical methods for combating bots. The problem exists in 243 countries, with 1.2 billion users, and lots of juridicial issues. A recent Council of Europe Cybercrime Convention agreed to by the United States will be changing the U.S. legal scene soon.

Someone asked about attacking evil servers, and the unanimous response was "Don't do it!" Don't even log in unless you have authorization.

■ ***A Case Study of the Rustock Rootkit and Spam Bot***

*Ken Chiang and Levi Lloyd, Sandia National Laboratories*

Ken Chiang told us about reverse engineering Backdoor .Rustock.B or Spam-Mailbot.c. Chiang and Levi Lloyd started by packet capture and could see that the bot uses HTTP POSTs, with encrypted payloads, for C&C. That made them interested in recovering the key used in order to learn more about the C&C channel.

Chiang told us that Rustock has three different phases of deobfuscation. The first phase deobfuscates the rootkit loader, which contains the second deobfuscation routine. The rootkit includes the third deobfuscation routine and unpacks the spam module. The rootkit then destroys the magic numbers in the PE and MZ headers to help defeat RAM forensics. The rootkit adds a value to the services registry key to restart itself upon boot, and it hides this key once the rootkit is running. The rootkit hooks several system calls to hide itself and injects the spam module threads into services.exe.

Levi took over and talked about the spam module. The C&C channel uses HTTP POST and performs a key exchange with a login.php script. The RC4 key is stored in a global sruct in memory, created by the client, encrypted using the server's public key and sent to the server. The server can decrypt the key with its private key, and then the server responds with a check. Once the client responds, the client is ready to accept commands. Rekeying and a new login occur every few minutes. The real reason for the exchange is to collect a list of mail servers, some spam content, and a list of targets to which to send the spam.

Dan Geer asked, "Is a good botnet better run than the average home system?" Levi answered that he could agree with that, as the client gets a list of processes to kill, which includes other bot software.

■ ***The Anatomy of Clickbot.A***

*Neil Daswani, Michael Stoppelman, and the Google Click Quality and Security Teams, Google, Inc.*

Neil Daswani began by explaining the business model exploited by Clickbot.A. What most of us know is that advertisers pay for click-throughs, and what has occurred to many of us is that this mechanism appears to be ripe for fraud. It turns out that this is indeed an issue for Google,

and Clickbot.A provides a stunning example of just how complex exploits can be.

To begin with, the attacker set up Web servers acting as doorway sites, using doorway.php, signed up to be subsyndicators, then signed up referral accounts to get paid for page views. Then Clickbot.A went after syndicated search engines, worked to be under the radar, sending very few clicks from each bot. Google did notice a pattern in the click-through traffic, and it marked Clickbot.A clicks invalid based on a recognizable pattern. This scheme also used redirectors, a form of proxy that strips off identifying information, such as the Referer: header line in the client's request.

Clickbot.A is an IE Browser Helper Object (BHO), likely because BHOs run within the process space of the browser and have access to the entire DOM library. In what was described as a "slow infection," desktops infected rose from 100 in May 2006 to 100,000 in mid-June 2006. Clickbot.A was distributed as a trojanned game download. The botnet server was written using PHP, and it did not initially have any form of access control, making it impossible to know how many desktops were infected. You can see screen output from the script in the paper. Neil also mentioned that only 7 out of 24 AV packages even recognized Clickbot.A as malware in June 2006.

Clickbot.A generates traffic by contacting the botmaster site, requesting a keyword and doorway site URL, then using this information to choose an ad to click on, but it does this only once every 15 minutes. Neil presented some back-of-the-envelope calculations about the amount of money a successful click-fraud campaign like this might make, obtaining a value of about $50,000. Google claims that there is less than a 10% rate of click fraud.

## WORK-IN-PROGRESS REPORTS

Steve Santorellis, Microsoft, announced conferences coming up in Sydney in July and in France in November for Law Enforcement (LE) and training of LE; some academia will be coming to these events. Contact steves@microsoft.com.

Michael Collins (CERT/NetSA), SEI, described measurements of machines that have likely been compromised (unclean machines). He expected that IP addresses of compromised systems would be randomly distributed, but he instead found that IP addresses cluster on certain netblocks rather than in any random block on the network; CERT has been monitoring a large /6 or /8 network since 2002. Looking for tightly packed addresses, Collins compared 600,000 bot addresses and found that unclean addresses tend to cluster. His data shows that uncleanliness is persistent over six months and that spamming but not phishing is closely related to unclean machines.

Dan Geer spoke about security metrics. He reminded attendees that the Metricon workshop will be at USENIX Security in Boston this summer and that CMU has an economics workshop this summer. Dan said that little data sharing is going on in financial and energy sectors about Internet-related fraud, but that insurance companies are really interested in this. In terms of presenting statistics, the best you can do now is trend analysis, and used the CSI/FBI surveys as an example. He advised that if you present statistics, be consistent about how you collect your data and the terms used to present them. Dan did point out that eBay takes down 1000 fraudulent sites per day, and eTrade reported a loss of $0.12 per share on $18 million in profits—losses based on stolen identity via keystroke logging.

Thorstein Holz of the University of Mannheim, Germany (and a frequent *;login:* contributor), described examples of HTTP-based bots. He did malware analysis using Nepenthes to collect examples. His group found that many new bots use HTTP in their analysis, with many having second- or even third-stage downloads. Recently he has seen bots using HTTP control channels and encrypted commands; he showed examples of doing ping, creating UID, getting second- or third-stage code, getting new code, and periodically querying the same HTTP server. Also, he has seen bots actually sending email for communicating, like slow-motion IRC.

D. Dagon of Georgia Tech pointed out that there are tens of thousands of new versions of malware appearing and that these are not being created by industrious individuals. Instead, queen software and code generators morph existing malware into new versions that won't be identified by AV. He suspects that there is just a small group of people doing this. His agenda is to find the people doing this. He also wants to collect more examples, so we can do analysis and learn more about obfuscation techniques.

William Zalewski of AOL claimed that bots providing SOCKS proxies are a silent but growing threat to the Internet. Based on his own analysis of traffic, he believes that there are many more relays than are active and that some provide reverse-connect proxies (where the bot goes out through a firewall or NAT, then offers to relay external connections internally). He has seen reverse-connect proxies that are totally nonrandom in behavior, connecting every 20 minutes. He has seen both v4 (a simple nine-byte setup) and v5 SOCKS proxies (a more complex four-packet exchange setup) being used.

M. AbuRajab of Johns Hopkins suggests clustering malware by activity, not by the features of the binary. Basically, he wants a feature vector for bots. Botnets apparently are purpose-built, not off-the-rack, a reflection of a clustering in the underground economy.

Masashi Eto, NICT, presented an integrated analysis of threats in large networks. His group, NICTER, monitors a

darknet of 100,000 IP addresses for real-time detection of incident candidates. They use automatic capture of malware, Nepenthes, and code analysis. Eto demonstrated some very cool 3D animated visualizations, at the very end of the WiPs and the workshop.