Your Botnet is My Botnet: Analysis of a Botnet Takeover

Brett Stone-Gross, Marco Cova, Lorenzo Cavallaro, Bob Gilbert, Martin Szydłowski, Richard Kemmerer, Chris Kruegel, Giovanni Vigna
Security Group
Department of Computer Science
University of California, Santa Barbara
[bsstone,marco,sullivan,rgilbert,msz,kemm,chrism,vigna]@cs.ucsb.edu

ABSTRACT
Botnets, networks of malware-infected machines that are controlled by an adversary, are the root cause of a large number of security threats on the Internet. A particularly sophisticated and insidious type of bot is Torpig, a malware program that is designed to harvest sensitive information (such as bank account and credit card data) from its victims. In this paper, we report on our efforts to take control of the Torpig botnet for ten days. Over this period, we observed more than 180 thousand infections and recorded more than 70 GB of data that the bots collected. While botnets have been “hijacked” before, the Torpig botnet exhibits certain properties that make the analysis of the data particularly interesting. First, it is possible (with reasonable accuracy) to identify unique bot infections and relate that number to the more than 1.2 million IP addresses that contacted our command and control server. This shows that botnet estimates that are based on IP addresses are likely to report inflated numbers. Second, the Torpig botnet is large, targets a variety of applications, and gathers a rich and diverse set of information from the infected victims. This opens the possibility to perform interesting data analysis that goes well beyond simply counting the number of stolen credit cards.

1. INTRODUCTION
Malicious code (or malware) has become one of the most pressing security problems on the Internet. In particular, this is true for bots [3], a type of malware that is written with the intent of taking control over hosts on the Internet. Once infected with a bot, the victim host will join a botnet, which is a network of compromised machines that are under the control of a malicious entity, typically referred to as the botmaster. Botnets are the primary means for cyber criminals to carry out their nefarious tasks, such as sending spam mails [30], launching denial-of-service attacks [24], or stealing personal data such as mail accounts or bank credentials [14,32]. To take over the Torpig (a.k.a. Sinowal, Anserin) botnet for ten days, Torpig, which has been described in [34] as “one of the most advanced pieces of crimeware ever created,” is a type of malware that is typically associated with bank account and credit card theft. However, as we will see, it also steals a variety of other personal information. To take control of the Torpig botnet, we exploited a weakness in the way that bots try to locate their C&C servers. More precisely, Torpig uses an increasingly popular technique to increase the reliability of its C&C infrastructure, which we term domain flux. With domain flux, each bot periodically (and independently) generates a list of domains that it contacts. The first host that sends a reply that identifies it as a valid C&C server is considered genuine, until the next period of
domain generation is started. We leveraged information about the domain generation algorithm and Torpig’s C&C protocol to register domains that the infected hosts would contact. By providing a valid response, the bots accepted our server as genuine. While “hijacking” a botnet can provide valuable information about the infected host population, Torpig has certain properties that make the analysis particularly insightful. First, Torpig bots transmit identifiers that allow us to distinguish between individual infections. When comparing the number of infections with the number of IP addresses that contacted our C&C server, we found a significant discrepancy. This calls into question previous botnet size estimates that are based on IP address counts. Second, Torpig is a data harvesting bot that targets a wide variety of applications and extracts a wealth of information from the infected victims. Together with the large size of the botnet (we observed more than 180 thousand infections), we have access to a rich data set that sheds light on the quantity and nature of the data that cyber criminals can harvest, the financial profits that they can make, and the threats to the security and privacy of bot victims.

To summarize, the contributions of this paper are the following:

- We introduce domain flux, a technique that recently gained popularity among botnet authors as a way to increase the robustness of their networks. Moreover, we explain how we exploited this technique to take control of the Torpig botnet for ten days.
- We identify a mechanism to distinguish between individual bot infections, and relate this finding to the IP addresses that contacted our C&C server. This revealed a problem in the common approach used to estimate botnet size based on IP counts.
- We analyze the data that the Torpig bots have collected and relayed to our C&C server. This provides insights into the dimension of the financial and privacy threats that botnet victims are exposed to.

2. BACKGROUND

Torpig is a malware that has drawn much attention recently from the security community. On the surface, it is one of the many Trojan horses infesting today’s Internet that, once installed on the victim’s machine, steals sensitive information and relays it back to its controllers. However, the sophisticated techniques it uses to steal data from its victims, the complex network infrastructure it relies on, and the vast financial damage that it causes set it apart from other threats.

So far, Torpig has been distributed to its victims as part of Mebroot. Mebroot is a rootkit that takes control of a machine by replacing the system’s Master Boot Record (MBR). This allows Mebroot to be executed at boot time, before the operating system is loaded, and to remain undetected by most anti-virus tools. More details on Mebroot can be found in [6,9,21]. In this paper, we will focus on Torpig, introducing Mebroot only when necessary to understand Torpig’s behavior. In particular, hereinafter, we present the lifecycle of Torpig and the organization of the Torpig botnet, as we observed it during the course of our analysis. We will use Figure 1 as a reference.

Victims are infected through drive-by-download attacks [27]. In these attacks, web pages on legitimate but vulnerable web sites (1) are modified with the inclusion of HTML tags that cause the victim’s browser to request JavaScript code (2) from a web site (the drive-by-download server in the figure) under control of the attackers (3). This JavaScript code launches a number of exploits against the browser or some of its components, such as ActiveX controls and plugins. If any exploit is successful, an executable is downloaded from the drive-by-download server to the victim machine, and it is executed (4).

The downloaded executable acts as an installer for Mebroot. The installer injects a DLL into the file manager process (explorer.exe), and execution continues in the file manager’s context. This makes all subsequent actions appear as if they were performed by a legitimate system process. The installer then loads a kernel driver that wraps the original disk driver (disk.sys). At this point, the installer has raw disk access on the infected machine. The installer can then overwrite the MBR of the machine with Mebroot. After a few minutes, the machine automatically reboots, and Mebroot is loaded from the MBR.

Mebroot has no malicious capability per se. Instead, it provides a generic platform that other modules can leverage to perform their malicious actions. In particular, Mebroot provides functionality to manage (install, uninstall, and activate) such additional modules. Immediately after the initial reboot, Mebroot contacts the Mebroot C&C server to obtain malicious modules (5). These modules are saved in encrypted form in the system32 directory, so that, if the user reboots the machine, they can be immediately reused without having to contact the C&C server again. The saved modules are timestamped and named after existing files in the same directory (they are given a different, random extension), to avoid raising suspicion. After the initial update, Mebroot contacts its C&C server periodically, in two-hour intervals, to report its current configuration (i.e., the type and version number of the currently installed modules) and to potentially receive updates. All communication with the C&C server occurs via HTTP requests and responses and is encrypted using a sophisticated, custom encryption algorithm [6]. Currently, no publicly available tool exists to circumvent this encryption scheme.

During our monitoring, the C&C server distributed three modules, which comprise the Torpig malware. Mebroot injects these modules (i.e., DLLs) into a number of applications. These applications include the Service Control Manager (services.exe), the file manager, and 29 other popular applications, such as web browsers (e.g., Internet Explorer, Firefox, Opera), FTP clients (LeechFTP, CuteFTP), email clients (e.g., Thunderbird, Outlook, Eudora), instant messengers (e.g., Skype, ICQ), and system programs (e.g., the command line interpreter cmd.exe). After the injection, Torpig can inspect all the data handled by these programs and identify and store interesting pieces of information, such as credentials for online accounts and stored passwords.

Periodically (every twenty minutes, during the time we monitored the botnet), Torpig contacts the Torpig C&C server to upload the data stolen since the previous reporting time (6). This communication with the server is also over HTTP and is protected by a simple obfuscation mechanism, based on XORing the clear text with an 8-byte key and base64 encoding. This scheme was broken by security researchers at the end of 2008, and tools are available to automate the decryption [18]. The C&C server can reply to a bot in one of several ways. The server can simply acknowledge the data. We call this reply an ok response, from the string contained in the server’s reply. In addition, the C&C server can send a configuration file to the bot (we call this reply an ok response). The configuration file is obfuscated using a simple XOR-11 encoding. This makes it difficult for security researchers to reverse-engineer the code. However, the bot will only perform “man-in-the-browser” phishing attacks [12].

Torpig uses phishing attacks to actively elicit additional, sensitive information from its victims, which, otherwise, may not be ob-
served during the passive monitoring it normally performs. These attacks occur in two steps. First, whenever the infected machine visits one of the domains specified in the configuration file (typically, a banking web site), Torpig issues a request to an injection server. The server’s response specifies a page on the target domain where the attack should be triggered (we call this page the trigger page and it is typically set to the login page of a site), a URL on the injection server that contains the phishing content (the injection URL), and a number of parameters that are used to fine tune the attack (e.g., whether the attack is active and the maximum number of times it can be launched). The second step occurs when the user visits the trigger page. At that time, Torpig requests the injection URL from the injection server and injects the returned content into the user’s browser (7). This content typically consists of an HTML form that asks the user sensitive information such as credit card numbers and social security numbers.

These phishing attacks are very difficult to detect, even for attentive users. In fact, the injected content carefully reproduces the style and look-and-feel of the target web site. Furthermore, the injection mechanism defies all phishing indicators included in modern browsers. For example, the SSL configuration appears correct and so does the URL displayed in the address bar. Appendix A contains an example phishing page for Wells Fargo Bank.

Communication with the injection server is protected using the standard HTTPS protocol. However, since Torpig does not check the validity of the server’s certificate and blindly accepts any self-signed certificate, it is possible to mount a man-in-the-middle attack and recover the data exchanged with the injection server.

In summary, Torpig relies on a fairly complex network infrastructure to infect machines, retrieve updates, perform active phishing attacks, and send the stolen information to its C&C server. We observed that the schemes used to protect the communication in the Torpig botnet (except those used by the Mebroot C&C) are insufficient to guarantee basic security properties (confidentiality, integrity, and authenticity). This was a weakness that enabled us to seize control of the botnet.

3. DOMAIN FLUX

A fundamental aspect of any botnet is that of coordination; i.e., how the bots identify and communicate with their C&C servers. Traditionally, C&C hosts have been located by their bots using their IP address, DNS name, or their node ID in peer-to-peer overlays. In the recent past, botnet authors have identified several ways to make these schemes more flexible and robust against take-down actions, e.g., by using IP fast-flux techniques [15]. With fast-flux, the bots would query a certain domain that is mapped onto a set of IP addresses, which change frequently. This makes it more difficult to take down or block a specific C&C server. However, fast-flux uses only a single domain name, which constitutes a single point of failure.

Torpig solves this issue by using a different technique for locating its C&C servers, which we refer to as domain flux. With domain flux, each bot uses a domain generation algorithm (DGA) to compute a list of domain names. This list is computed independently by each bot and is regenerated periodically. Then, the bot attempts to contact the hosts in the domain list in order until one succeeds, i.e., the domain resolves to an IP address and the corresponding server provides a response that is valid in the botnet’s protocol. If a domain is blocked (for example, the registrar suspends it to comply with a take-down request), the bot simply rolls over to the following domain in the list. Domain flux is also used to contact the Mebroot C&C servers and the drive-by-download servers. Domain flux is increasingly popular among botnet authors. In fact, similar mechanisms were used before by the Kraken/Bobax and the Srizbi bots [33, 40], and, more recently, by the Conficker worm [26].

In Torpig, the DGA is seeded with the current date and a numerical parameter. The algorithm first computes a “weekly” domain name, say dw, which depends on the current week and year, but is independent of the current day (i.e., remains constant for the entire week). Using the generated domain name dw, a bot appends a number of TLDs: in order, dw.com, dw.net, and dw.biz. It then

```python
def generate_daily_domain(t, p):
    t = GetLocalTime()
    return generate_domain(t, p)

def scramble_date(t, p):
    if t.year < 2007:
        t.year = 2007
    s = scramble_date(t, p)
    c1 = ((t.year >> 2) + 0x3fc0) + s % 25 + 'a'
    c2 = (t.month + s) % 10 + 'a'
    c3 = ((t.year & 0xff) + s) % 25 + 'a'
    c4 = t.day % 10 + '1'
    return c1 + c2 + c3 + c4 + suffix[t.month - 1]

def generate_domain(t, p):
    suffix = ['anj', 'eef', 'arm', 'pra', 'aym', 'unj', 'ulj', 'uaq', 'esp', 'kot', 'onv', 'educ']
    t = GetLocalTime()
    return (((t.month ^ t.day) + t.day) + t.day! + p) +
    t.day + t.year
```

Listing 1: Torpig daily domain generation algorithm.
resolves each domain and attempts to connect to its C&C server. If all three connections fail, Torpig computes a “daily” domain, say dd, which in addition depends on the current day (i.e., a new domain dd is generated each day). Again, dd.com is tried first, with fallbacks to dd.net and dd.biz. If these domains also fail, Torpig attempts to contact the domains hardcoded in its configuration file (e.g., rikora.com, pinakola.com, and flippibi.com). Listing 1 shows the pseudo-code of the routines used to generate the daily domains dd. The DGA used in Torpig is completely deterministic; i.e., once the current date is determined, all bots generate the same list of domains, in the same order.

From a practical standpoint, domain flux generates a list of “rendezvous points” that may be used by the botmasters to control their bots. Not all the domains generated by a DGA need to be valid for the botnet to be operative. However, there are two requirements that the botmasters must satisfy to maintain their grip on the botnet. First, they must control at least one of the domains that will be contacted by the bots. Second, they must use mechanisms to prevent other groups from seizing domains that will be contacted by bots before the domains under their control.

In practice, the Torpig controllers registered the weekly .com domain and, in a few cases, the corresponding .net domain, for backup purposes. However, they did not register all the weekly domains in advance, which was a critical factor in enabling our hijacking.

### 3.1 Threats and Countermeasures

The use of domain flux in botnets has important consequences in the arms race between botmasters and defenders. From the attacker’s point of view, domain flux is yet another technique to potentially improve the resilience of the botnet against take-down attempts. More precisely, in the event that the current rendezvous point is taken down, the botmasters simply have to register the next domain in the domain list to regain control of their botnet. On the contrary, to the defender’s advantage, domain flux opens up the possibility of sinkholing (or “hijacking”) a botnet, by registering an available domain that is generated by the botnet’s DGAs and returning an answer that is a valid C&C response (to keep bots from switching over to the next domain in the domain list). As we mentioned, Torpig allowed both of these actions: C&C domain names were available for registration and it was possible to forge valid C&C responses.

The feasibility of these sinkholing attacks depends not only on technical means (e.g., the ability to reverse engineer the botnet protocol and to forge a valid C&C server’s response), but also on economic factors, in particular the cost of registering a number of domains sufficient to make the sinkholing effective. Since domain registration comes at a price (currently, from about $5 to $10 per year per .com and .net domain name), botmasters could prevent attacks against domain flux by making them economically infeasible, for example, by forcing defenders to register a disproportionate number of names. Unfortunately, this is a countermeasure that is already in use. Newer variants of Conficker generate 50,000 domains per day and introduce non-determinism in their generation algorithm [26].

Taking over all the domains generated by Conficker at market prices would cost between $91.3 million and $182.5 million per year. Furthermore, the domain flux arms race is clearly in favor of the malware authors. Generating thousands more domains requires an inexpensive modification to the bot code base, while registering them costs time and money.

In short, the idea of combating domain flux by simply acquiring more domains is clearly not scalable in the long term, and new approaches are needed to tilt the balance away from the botmasters. In particular, the security community should build a stronger relationship with registrars. Registrars, in fact, are the entity best positioned to mitigate malware that relies on DNS (including domain flux), but, with few exceptions, they often lack the resources, incentives, or culture to deal with the security issues associated with their roles. In addition, rogue registrars (those known to be a safe haven for the activity of cyber criminals) should lose their accreditation. While processes exist to terminate registrar accreditation agreements (a recent case involved the infamous EstDomains registrar [1]), they should be streamlined and used more promptly.

### 4. TAKING CONTROL OF THE BOTNET

In this section, we describe in more detail how we obtained control over the Torpig botnet. We registered domains that bots would resolve and setup a server to which bots would connect to find their C&C. Moreover, we present our data collection and hosting infrastructure and review a timeline of events during our period of control.

The behavior of the botmasters was to not register many of the future Torpig C&C domains in advance. Therefore, we were able to register the .com and .net domains that were to be used by the botnet for three consecutive weeks from January 25th, 2009 to February 15th, 2009. However, on February 4th, 2009, the Mebroot controllers distributed a new Torpig binary that updated the domain algorithm. This ended our control prematurely after ten days. Mebroot domains, in fact, allow botmasters to upgrade, remove, and install new malware components at any time, and are tightly controlled by the criminals. It is unclear why the controllers of the Mebroot botnet did not update the Torpig domain algorithm sooner to thwart our sinkholing.

#### 4.1 Sinkholing Preparation

We purchased service from two different hosting providers that are well-known to be unresponsive to abuse complaints, and we registered our .com and .net domains with two different registrars. This provided redundancy so that if one domain registrar or hosting provider suspended our account, we would be able to maintain control of the botnet. This proved to be useful when our .com domains were suspended on January 31, 2009 due to an abuse complaint. Fortunately, we owned the backup .net domain and were able to continue our collection unabated during this period until we could get our primary domain reinstated.

On our machines, we set up an Apache web server to receive and log bot requests, and we recorded all network traffic. We then automated the process of downloading the data from our hosting providers. Once a datafile was downloaded, we removed it from the server on the hosting provider. Therefore, if our servers were compromised, an attacker would not have access to any historical data. During the ten days that we controlled the botnet, we collected over 8.7GB of Apache log files and 69GB of pcap data.

We expected infected machines to connect to us on January 25th, which was the day when bots were supposed to switch to the first weekly domain name that we owned. However, on January 19th, when we started our collection, we instantly received HTTP requests from 359 infected machines. This was almost a week before the expected time. After analyzing the geographical distribution of these machines and the data they were sending, we concluded that these were probably systems that had their clock set incorrectly.

---

1All the collected traffic was encrypted using 256-bit AES.
4.2 Data Collection Principles

During our collection process, we were very careful with the information that we gathered and with the commands that we provided to infected hosts. We operated our C&C servers based on previously established legal and ethical principles [2]. In particular, we protected the victims according to the following:

**Principle 1.** The sinkholed botnet should be operated so that any harm and/or damage to victims and targets of attacks would be minimized.

**Principle 2.** The sinkholed botnet should collect enough information to enable notification and remediation of affected parties.

There were several preventative measures that were taken to ensure Principle 1. In particular, when a bot contacted our server, we always replied with an ok message and never sent it a new configuration file. By responding with okn, the bots remained in contact only with our servers. If we had not replied with a valid Torpig response, the bots would have switched over to the .biz domains, which had already been registered by the criminals. Although we could have sent a blank configuration file to potentially remove the web sites currently targeted by Torpig, we did not do so to avoid unforeseen consequences (e.g., changing the behavior of the malware on critical computer systems, such as a server in a hospital). We also did not send a configuration file with a different HTML injection server IP address for the same reasons. To notify the affected institutions and victims, we stored all the data that was sent to us, in accordance with Principle 2, and worked with ISPs and law enforcement agencies, including the United States Department of Defense (DoD) and FBI Cybercrime units, to assist us with this effort. This cooperation also led to the suspension of the current Torpig domains owned by the cyber criminals.

5. BOTNET ANALYSIS

As mentioned previously, we have collected almost 70GB of data over a period of ten days. The wealth of information that is contained in this dataset is remarkable. In this section, we present the results of our data analysis and important insights into the size of botnets and their victims.

5.1 Data Collection and Format

All bots communicate with the Torpig C&C through HTTP POST requests. The URL used for this request contains the hexadecimal representation of the bot identifier and a submission header. The body of the request contains the data stolen from the victim’s machine, if any. The submission header and the body are encrypted using the Torpig encryption algorithm (base64 and XOR). The bot identifier (a token that is computed on the basis of hardware and software characteristics of the infected machine) is used as the symmetric key and is sent in the clear.

After decryption, the submission header consists of a number of key-value pairs that provide basic information about the bot. More precisely, the header contains the timestamp when the configuration file was last updated (tsa), the IP address of the bot or a list of IPs in case of a multi-homed machine (ip), the port numbers of the HTTP and SOCKS proxies that Torpig opens on the infected machine (hport and aport), the operating system version and locale (os and cn), the bot identifier (nid), and the build and version number of Torpig (bid and ver).

**Table 1: Data items sent to our C&C server by Torpig bots.**

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Data Items (#)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mailbox account</td>
<td>54,090</td>
</tr>
<tr>
<td>Email</td>
<td>1,258,862</td>
</tr>
<tr>
<td>Form data</td>
<td>11,966,532</td>
</tr>
<tr>
<td>HTTP account</td>
<td>411,039</td>
</tr>
<tr>
<td>FTP account</td>
<td>12,307</td>
</tr>
<tr>
<td>POP account</td>
<td>415,206</td>
</tr>
<tr>
<td>SMTP account</td>
<td>100,472</td>
</tr>
<tr>
<td>Windows password</td>
<td>1,235,122</td>
</tr>
</tbody>
</table>

The request body consists of zero or more data items of different types, depending on the information that was stolen. Table 1 shows the different data types that we observed during our monitoring. In particular, mailbox account items contain the configuration information for email accounts, i.e., the email address associated with the mailbox and the credentials required to access the mailbox and send emails from it. Torpig obtains this information from email clients, such as Outlook, Thunderbird, and Eudora. Email items consist of email addresses, which can presumably be used for spam purposes. According to [39], Torpig initially used spam emails to propagate, which may give another explanation for the botmasters’ interest in email addresses. Form data items contain the content of HTML forms submitted via POST requests by the victim’s browser. More precisely, Torpig collects the URL hosting the form, the URL that the form is submitted to, and the name, value, and type of all form fields. These data items frequently contain the usernames and passwords required to authenticate with web sites. Notice that credentials transmitted over HTTPS are not safe from Torpig, since Torpig can access them before they are encrypted by the SSL layer (by hooking appropriate library functions). HTTP account, FTP account, POP account, and SMTP account data types contain the credentials used to access web sites, FTP, POP, and SMTP accounts, respectively. Torpig obtains this information by exploiting the password manager functionality provided by most web and email clients. SMTP account items also contain the source and destination addresses of emails sent via SMTP. Finally, the Windows password data type is used to transmit Windows passwords and other uncategorized data elements. Figure 3 shows a sample of the data items sent by a Torpig bot.

5.2 Botnet Size

The first question we want to answer about the Torpig botnet is “What is its size?” By size, here we mean the number of distinct infections that we observed during our control of the botnet. The problem of determining the size of a botnet is known to be difficult [20, 28]. In particular, researchers have often based their estimations on the number of unique IP addresses observed connecting to a C&C server. Unfortunately, this method has several problems. For example, it discounts the effect of widely used network techniques and protocols that may hide multiple machines behind a single address (NAT) or assign multiple addresses over time to the same machine (DHCP and dial-up). Interestingly, the Torpig botnet has a feature that makes an accurate size estimation possible, because each infected machine reports distinct host-based identifiers to the C&C server. In the next section we discuss the characteristics of the botnet that enabled us to determine an overall range for the number of infected machines.

2Note that even though bots are identified by a unique id, a bot host that is cleaned and then re-infected would receive a new id, and, therefore, it might be counted twice.
Counting Bots by nid

As a starting point to estimate the total number of victims, we analyzed the nid field that Torpig sends in the submission header. Our hypothesis was that this value was unique for each machine and remained constant over time, and that, therefore, it would provide an accurate method to uniquely identify each bot.

By reverse engineering the Torpig binary, we were able to reconstruct the algorithm used to compute this 8-byte value. In particular, the algorithm first queries the primary SCSI hard disk for its model and serial numbers. If no SCSI hard disk is present, or retrieving the disk information is unsuccessful, it will then try to extract the same information from the physical hard disk drive (i.e., IDE or SATA). The disk information is then used as input to a hashing function that produces the final nid value. If retrieving hardware information fails, the nid value is obtained by concatenating the hard-coded value of 0xBAD1D222 with the Windows volume serial number.

In all cases, the nid depends on (software or hardware) characteristics of the infected machine’s hard disk. Therefore, it does not change, unless the hard disk is replaced (in which case the machine would no longer be infected), or the user manually changes the system’s volume serial number (which requires special tools and is not likely to be done by casual users). This gave us confidence that the nid remains constant throughout the life of an infected machine.

We then attempted to validate whether the nid is unique for each bot. Therefore, we correlated this value with the other information provided in the submission header and bot connection patterns to our server. In particular, we were expecting that all submissions with a specific nid would report the same values for the os, cn, bld, and ver fields. Unfortunately, we found 2,079 cases for which this assumption did not hold.

Therefore, we conclude that counting unique nids understimates the size of the botnet. As a reference point, between Jan 25, 2009 and February 4, 2009, 180,835 nid values were observed.

Counting Bots by Submission Header Fields

As a more accurate method to identify infected machines, we used the nid, os, cn, bld, and ver values from the submission header that Torpig bots send. As we have seen, the nid value is mostly unique among bots, and the other fields help distinguishing different machines that have the same nid. In particular, the os (OS version number) and cn (locale information) fields are determined by using the system calls GetVersionEx and GetLocaleInfo, respectively, and do not change unless the user modifies the locale information on her computer or changes her OS. The values of the bld and ver fields are hard-coded into the Torpig binary.

We decided not to use the ts field (timestamp of the configuration file), since its value is determined by the Torpig C&C that distributed the configuration file and not by characteristics of the bot. Also, we discarded the ip field, since it could change depending on DHCP and other network configurations, and the sport and hport fields, which specify the proxy ports that Torpig opens on the local machine, because they could change after a reboot.

By counting unique tuples from the Torpig headers consisting of (nid, os, cn, bld, ver), we estimate that the total size of the botnet for the ten days of our monitoring consisted of 182,914 machines.

Identifying Probers and Researchers

Finally, we wanted to identify security researchers and other curious individuals who probed our botnet servers. These do not correspond to actual victims of the botnet and, therefore, we would like to identify them and subtract them from the total botnet size.

We used two heuristics to identify probers and (likely) security researchers. First, we observed that the nid values generated by infected clients running in virtual machines is constant. This is because the nid depends essentially on physical characteristics of the hard disk, and, by default, virtual machines provide virtual devices with a fixed model and serial number. Since virtual machines are often used by researchers to study malware in a contained environment, we assume that these bots in reality correspond to researchers studying the Torpig malware. In particular, we were able to determine the nid values generated on a standard configuration of the VMware and QEMU virtual machines and we found 40 hosts using these values. Second, we identified hosts that send invalid requests to our C&C server (i.e., requests that cannot be generated by Torpig bots). For example, these bots used the GET HTTP method in requests where a real Torpig bot would use the POST method. Using this approach, we discounted another 74 hosts. We further ignored background noise, such as scanning of our web server and traffic from search engine bots. After subtracting probers and researchers, our final estimate of the botnet size is 182,800 hosts.
Figure 4 displays the number of unique IP addresses observed during the ten days that we were in control of the Torpig C&C. After the initial spike when the bots started to contact our server, there was a consistent diurnal pattern of unique IP addresses with an average of 4,690 new IPs per hour. In contrast, the average number of new bots observed was 705 per hour, with a very rapid drop-off after the first peak, as shown in Figure 5. Therefore, the number of cumulative new IP addresses that we saw over time increased linearly, as shown in Figure 6. On the other hand, the aggregate number of new bots observed decayed quickly. Figure 7 shows that more than 75% of all new Torpig bots during the ten-day interval were observed in the first 48 hours. As a result, the number of unique IP addresses considerably overestimated the number of infections, and, provided enough time, could have distorted the number of infections even more. We believe this result casts doubt on previous studies that have estimated the number of infected hosts for particular botnets based solely on the number of unique IP addresses that were seen.

The difference between IP count and the actual bot count can be attributed to DHCP and NAT effects. In networks using the DHCP protocol (or connecting through dial-up lines), clients (machines on the network) are allocated an address from a pool of available IP addresses. The allocation is often dynamic, that is, a client is not guaranteed to always be assigned the same IP address. This can inflate the number of observed IP addresses at the botnet C&C server. Short leases (the length of time for which the allocation is valid) can further magnify this effect. This phenomenon was very common during our monitoring. In fact, we identified the presence of ISPs that rotate IP addresses so frequently that almost every time that an infected host on their network connected to us, it had a new IP address. In one instance, a single host had changed IP addresses 694 times in just ten days! In some cases, the same host was associated with different IP addresses on the same autonomous systems, but different class B /16 subnets. We observed this DHCP churn on several different networks with the most common being, in descending order: Deutsche Telekom, Verizon, and BellSouth. Overall, there were 706 different machines that were seen with more than one hundred unique IP addresses. At this point, we can only speculate why these ISPs recycle IP addresses so frequently.

Furthermore, by comparing the number of bots we observed and their IP addresses, we can determine the effect of DHCP churn at a country level. Interestingly, the IP address count significantly overestimates the infection count in some countries, because the ISPs in those regions recycle IP addresses more often in comparison to others as shown in Table 2. For instance, a naïve estimate per country would consider Italy and Germany to have the largest number of infections. However, the ISPs in those countries assign IP addresses much more frequently than their U.S. counterparts. In fact, Germany had less than half the number of infected hosts, yet double the number of IP address connections. Furthermore, the ratio of IPs to hosts in Germany was four times higher than that of the United States. Because Torpig spreads through drive-by-download web sites, we believe the clustering by country reflects that most of the malicious sites use English, Italian, or German, since these are the top affected countries.

The information provided in the Torpig headers also allows us to estimate the impact of NAT. NAT is commonly used to enable shared Internet access for an entire private network through a single public access (masquerading). This technique reduces the number of IPs observed at the C&C server, since all the infected ma-
machines in the masqueraded network would count as one. By looking at the IP addresses in the Torpig headers we are able to determine that 144,236 (78.9%) of the infected machines were behind a NAT, VPN, proxy, or firewall. We identified these hosts by using the non-publicly routable IP addresses listed in RFC 1918: 10/8, 192.168/16, and 172.16-172.31/16. We observed 9,336 distinct bots for 2,753 IP addresses from these infected machines on private networks. Therefore, if the IP address count was used to determine the number of hosts it would underestimate the infection count by a factor of more than 3 times.

**New Infections**

The Torpig submission header provides the timestamp of the most recently received configuration file. We leveraged this fact to approximate the number of machines newly infected during the period of observation by counting the number of distinct victims whose initial submission header contains a timestamp of 0. Figure 8 shows the new infections over time. In total, we estimate that there were 49,294 new infections while the C&C was under our control. New infections peaked on the 25th and the 27th of January. There were 4,073 new infections on the 27th of January. We can only speculate that, on those days, a popular web site was compromised and redirected its visitors to the drive-by-download servers controlled by the botmasters.

**5.3 Botnet as a Service**

An interesting aspect of the Torpig botnet is that there are indications that different groups would be dividing (and profiting from) the data it steals. Torpig DLLs are marked with a build type represented by the bld field in the header. This value is set during the drive-by-download (the build type is included in the URL that triggers the download) and remains the same during the entire lifecycle of an infection. The build type does not seem to indicate different feature sets, since different Torpig builds behave in the same way. However, Torpig transmits its build type in all communications with the C&C server, and, in particular, includes it in both the submission header (as the bld parameter) and in each data item contained in a submission body (for example, in Figure 2 the build type was gnh5). Therefore, the most convincing explanation of the build type is that it denotes different “customers” of the Torpig botnet, who, presumably, get access to their data in exchange for a fee. If correct, this interpretation would mean that Torpig is actually used as a “malware service”, accessible to third parties who do not want or cannot build their own botnet infrastructure.

During our study, we observed 12 different values for the bld parameter: dxtzrbc, eagle, gnh1, gnh2, gnh3, gnh4, gnh5, grey, grobin, grobin1, mentat, and zipp. Not all builds contribute equally to the amount of data stolen. The most active versions are dxtzrbc (5,432,528 submissions), gnh5 (2,836,198), and mentat (1,582,547).

**6. THREATS AND DATA ANALYSIS**

In this section, we will discuss the threats that Torpig poses and will turn our attention to the actual data that infected machines sent to our C&C server. We will see that Torpig creates a considerable potential for damage due not only to the sheer volume of data it collects, but also to the amount of computing resources the botnet makes available.

**6.1 Financial Data Stealing**

Consistent with the past few years’ shift of malware from a fun (or notoriety) activity to a for-profit enterprise [7, 13], Torpig is specifically crafted to obtain information that can be readily monetized in the underground market. Financial information, such as bank accounts and credit card numbers, is particularly sought after. For example, the typical Torpig configuration file lists roughly 300 domains belonging to banks and other financial institutions that will be the target of the “man-in-the-browser” phishing attacks described in Section 2.

Table 3 reports the number of accounts at financial institutions at 410 different institutions. The top targeted institutions were PayPal (1,770 accounts), Poste Italiane (765), Capital One (314), E*Trade (304), and Chase (217). On the other end of the spectrum, a large number of companies had only a handful of compromised accounts (e.g., 310 had ten or less). The large number of institutions that had been breached made notifying all of the interested parties a monumental effort. It is also interesting to observe that 38% of the credentials stolen by Torpig were obtained from the password manager of browsers, rather than by intercepting an actual login session.

Another target for collection by Torpig is credit card data. Using a credit card validation heuristic that includes the Luhn algorithm for validation, the botnet was able to steal 1,257,642 credit card numbers, 182,800 of which were valid. Table 3: Accounts at financial institutions stolen by Torpig.

**Table 2: Top 10 Infected Hosts by Country.**

<table>
<thead>
<tr>
<th>Country</th>
<th>IP Addresses</th>
<th>Bot IDs</th>
<th>DHCP Churns</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>158,209</td>
<td>54,627</td>
<td>1000</td>
<td>2.90</td>
</tr>
<tr>
<td>IT</td>
<td>383,077</td>
<td>46,508</td>
<td>1200</td>
<td>8.24</td>
</tr>
<tr>
<td>DE</td>
<td>325,816</td>
<td>24,413</td>
<td>1400</td>
<td>13.35</td>
</tr>
<tr>
<td>PL</td>
<td>44,117</td>
<td>6,365</td>
<td>1600</td>
<td>6.93</td>
</tr>
<tr>
<td>GR</td>
<td>31,745</td>
<td>5,733</td>
<td>1800</td>
<td>5.54</td>
</tr>
<tr>
<td>CH</td>
<td>45,809</td>
<td>5,402</td>
<td>2000</td>
<td>8.48</td>
</tr>
<tr>
<td>UK</td>
<td>30,706</td>
<td>4,826</td>
<td>2200</td>
<td>6.36</td>
</tr>
<tr>
<td>BG</td>
<td>21,465</td>
<td>4,792</td>
<td>2400</td>
<td>4.48</td>
</tr>
<tr>
<td>NL</td>
<td>11,240</td>
<td>3,037</td>
<td>2600</td>
<td>3.70</td>
</tr>
<tr>
<td>Other</td>
<td>190,070</td>
<td>2,331</td>
<td>2800</td>
<td>1.75</td>
</tr>
<tr>
<td>Totals</td>
<td>1,247,044</td>
<td>182,800</td>
<td>1,000,000</td>
<td>6.83</td>
</tr>
</tbody>
</table>

**Table 3: Accounts at financial institutions stolen by Torpig.**

<table>
<thead>
<tr>
<th>Country</th>
<th>Institutions</th>
<th>Accounts</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>60</td>
<td>4,287</td>
</tr>
<tr>
<td>IT</td>
<td>34</td>
<td>1,549</td>
</tr>
<tr>
<td>DE</td>
<td>122</td>
<td>641</td>
</tr>
<tr>
<td>ES</td>
<td>18</td>
<td>228</td>
</tr>
<tr>
<td>PL</td>
<td>14</td>
<td>102</td>
</tr>
<tr>
<td>Other</td>
<td>162</td>
<td>1,593</td>
</tr>
<tr>
<td>Total</td>
<td>410</td>
<td>8,310</td>
</tr>
</tbody>
</table>
and matching against the correct number of digits and numeric prefixes of card numbers from the most popular credit card companies, we extracted 1,660 unique credit and debit card numbers from our collected data. Through IP address geolocation, we surmise that 49% of the card numbers came from victims in the US, 12% from Italy, and 8% from Spain, with 40 other countries making up the balance. The most common cards include Visa (1,056), MasterCard (447), American Express (81), Maestro (36), and Discover (24).

While 86% of the victims contributed only a single card number, others offered a few more. Of particular interest is the case of a single victim from whom 30 credit card numbers were extracted. Upon manual examination, we discovered that the victim was an agent for an at-home, distributed call center. It seems that the card numbers were those of customers of the company that the agent was working for, and they were being entered into the call center’s central database for order processing.

Quantifying the value of the financial information stolen by Torpig is an uncertain process because of the characteristics of the underground markets where it may end up being traded. A report by Symantec [37] indicated (loose) ranges of prices for common goods and, in particular, priced credit cards between $0.10–$25 and bank accounts from $10–$1,000. If these figures are accurate, in ten days of activity, the Torpig controllers may have profitied anywhere between $83k and $8.3M.

Furthermore, we wanted to determine the rate at which the botnet produces new financial information for its controllers. Clearly, a botnet that generates all of its value in a few days and later only recycles stale information is less valuable than one where fresh data is steadily produced. Figure 9 shows the rate at which new bank accounts and credit card numbers were obtained during our monitoring period. In the ten days when we had control of the botnet, new data was continuously stolen and reported by Torpig bots.

### 6.2 Proxies

As we mentioned previously, Torpig opens two ports on the local machine, one to be used as a SOCKS proxy, the other as an HTTP proxy. 20.2% of the machines we observed were publicly accessible. Their proxies, therefore, could be easily leveraged by miscreants to, for example, send spam or navigate anonymously. In particular, we wanted to verify if spam was sent through machines in the Torpig botnet. We focused on the 10,000 IPs that contacted us most frequently. These, arguably, correspond to machines that are available for longer times and that are, thus, more likely to be used by the botmasters. We matched these IPs against the ZEN blocklist, a well-known and accurate list of IP addresses linked to spamming, which is compiled by the Spamhaus project [38]. We found that one IP was marked as a verified spam source or spam operation and 244 (2.45%) were flagged as having open proxies that are used for spam purposes or being infected with spam-related malware. While we have no evidence that the presence of these IPs on the ZEN blocklist is a consequence of the Torpig infection, it is clear that Torpig has the potential to drag its victims into a variety of malicious activities. Furthermore, since most IPs are "clean," they can be used for spamming, anonymous navigation, or other dubious enterprises.

### 6.3 Denial-of-Service

To approximate the amount of aggregate bandwidth among infected hosts, we mapped the IP addresses to their network speed, using the ip2location database. This information is summarized in Table 4. Unfortunately the database does not contain records for about two-thirds of the IP addresses, but from the information that it provides, we can see that cable and DSL lines account for 65% of the infected hosts. If we assume the same distribution of network speed for the unknown IP addresses, there is a tremendous amount of bandwidth in the hands of the botmaster, considering that there were more than 60,000 active hosts at any given time. In 2008, the median upstream bandwidth in the United States was 435 kbps for DSL connections [36]. Since the United States ranks as one of the slowest in terms of broadband speeds, we will use 435 kbps as a conservative estimate for each bot’s upstream bandwidth. Thus, the aggregate bandwidth for the DSL/Cable connections is roughly 17 Gbps. If we further add in corporate networks, which account for 22% of infected hosts, and consider that they typically have significantly larger upstream connections, the aggregate bandwidth is likely to be considerably higher. Hence, a botnet of this size could cause a massive distributed denial-of-service (DDoS) attack.

### 6.4 Password Analysis

Torpig bots stole 297,962 unique credentials (i.e., username and password pairs), sent by 52,540 different Torpig-infected machines over the ten days we controlled the botnet. This provided us with the opportunity to study the quality of passwords that are used online, based on a large data corpus.

The Torpig-stolen credentials were discovered as follows. For each infected host \( H \), we retrieved all the unique username and password pairs \( C \) submitted by \( H \). Next, we obtained the number, \( W_C \), of distinct web services where a credential \( C \) was used. Finally, if \( W_C \geq 2 \), we concluded that \( C \) had been reused across \( W_C \) different web services.

Our analysis found that almost 28% of the victims reused their credentials for accessing 368,501 web sites. According to a recent survey conducted by Sophos in March 2009 [35], one third of 676 Internet users neglect the importance of using strong passwords. Thus, our results confirm those of the Sophos poll.

In addition to checking for credential reuse, we also conducted an experiment to assess the strength of the 173,686 unique passwords discovered in the experiment above. To this end, we created a UNIX-like password file to feed John the Ripper, a popular password cracker tool [25]. The results are presented in Figure 11 in Appendix B. About 56,000 passwords were recovered in less than

![Figure 9: The arrival rate of financial data.](http://www.ip2location.com)
65 minutes by using permutation, substitution, and other simple replacement rules used by the password cracker (the "single" mode). Another 14,000 passwords were recovered in the next 10 minutes when the password cracker switched modes to use a large wordlist. Thus, in less than 75 minutes, more than 40% of the passwords were recovered. 30,000 additional passwords were recovered in the next 24 hours by brute force (the "incremental" mode).

### 6.5 Privacy and Other Threats

Although the data collected by Torpig is focused on financial information, it also consists of a (sparse) trace of the online activity of the users of infected machines. This data can be readily mined by the botmaster to determine all kinds of sensitive information that pose far-reaching threats to the privacy of the victims. We were curious about how extensive the breach of the victim’s privacy was, and, therefore, we further analyzed the data. We present some of the results of the analysis plus some interesting anecdotes in this section.

We wondered about the interests, habits, and concerns of the victims, so we analyzed the messages that users of infected machines send, for example, through webcam systems, forums, and chats. Since the full content of these messages is captured by Torpig, they often contain detailed (and private) descriptions of the lives of their authors. We focused on messages written in English and longer than a given threshold (250 characters). We obtained 6,542 messages. We classified each message in one or more categories, by defining a set of keywords for each category and matching the message’s contents against these keywords. The result is what one could call the *zeitgeist* of the Torpig botnet. The victims of Torpig prepare for exams and worry about grades (5% of the messages), look for professional advice from doctors and lawyers (1%), play video games (2%), seek jobs and submit resumes (14%), are sport fans (6%), discuss money (7%), trade goods online (4%), exchange insults (0.1%), and look for sex or partners online (4%).

We noted that online security is a concern of the infected population (almost 10% of the messages mention phishing, viruses, and spyware), but only a few people seem to suspect that they are using an infected machine. On the contrary, there is one who unwittingly announces that his computer has been fixed and the virus is gone. We looked for reactions to the phishing mechanism used by Torpig. We found it mostly among eBay and PayPal customers. In We looked for reactions to the phishing mechanism used by Torpig.

### Table 5: Top web account credentials sent by Torpig victims.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Account Credentials Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>google.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>facebook.com</td>
<td>Social Networking</td>
</tr>
<tr>
<td>myspace.com</td>
<td>Social Networking</td>
</tr>
<tr>
<td>netlog.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>libero.it</td>
<td>Search/Email/ISP</td>
</tr>
<tr>
<td>yahoo.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>nasza-klasa.pl</td>
<td>Social Networking</td>
</tr>
<tr>
<td>alice.it</td>
<td>Search/Email/ISP</td>
</tr>
<tr>
<td>live.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>hi5.com</td>
<td>Social Networking</td>
</tr>
</tbody>
</table>

Table 5: Top web account credentials sent by Torpig victims.

65 minutes by using permutation, substitution, and other simple replacement rules used by the password cracker (the "single" mode). Another 14,000 passwords were recovered in the next 10 minutes when the password cracker switched modes to use a large wordlist. Thus, in less than 75 minutes, more than 40% of the passwords were recovered. 30,000 additional passwords were recovered in the next 24 hours by brute force (the "incremental" mode).

### 6.5 Privacy and Other Threats

Although the data collected by Torpig is focused on financial information, it also consists of a (sparse) trace of the online activity of the users of infected machines. This data can be readily mined by the botmaster to determine all kinds of sensitive information that pose far-reaching threats to the privacy of the victims. We were curious about how extensive the breach of the victim’s privacy was, and, therefore, we further analyzed the data. We present some of the results of the analysis plus some interesting anecdotes in this section.

We wondered about the interests, habits, and concerns of the victims, so we analyzed the messages that users of infected machines send, for example, through webcam systems, forums, and chats. Since the full content of these messages is captured by Torpig, they often contain detailed (and private) descriptions of the lives of their authors. We focused on messages written in English and longer than a given threshold (250 characters). We obtained 6,542 messages. We classified each message in one or more categories, by defining a set of keywords for each category and matching the message’s contents against these keywords. The result is what one could call the *zeitgeist* of the Torpig botnet. The victims of Torpig prepare for exams and worry about grades (5% of the messages), look for professional advice from doctors and lawyers (1%), play video games (2%), seek jobs and submit resumes (14%), are sport fans (6%), discuss money (7%), trade goods online (4%), exchange insults (0.1%), and look for sex or partners online (4%).

We noted that online security is a concern of the infected population (almost 10% of the messages mention phishing, viruses, and spyware), but only a few people seem to suspect that they are using an infected machine. On the contrary, there is one who unwittingly announces that his computer has been fixed and the virus is gone. We looked for reactions to the phishing mechanism used by Torpig. We found it mostly among eBay and PayPal customers. In We looked for reactions to the phishing mechanism used by Torpig.

### Table 5: Top web account credentials sent by Torpig victims.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Account Credentials Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>google.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>facebook.com</td>
<td>Social Networking</td>
</tr>
<tr>
<td>myspace.com</td>
<td>Social Networking</td>
</tr>
<tr>
<td>netlog.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>libero.it</td>
<td>Search/Email/ISP</td>
</tr>
<tr>
<td>yahoo.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>nasza-klasa.pl</td>
<td>Social Networking</td>
</tr>
<tr>
<td>alice.it</td>
<td>Search/Email/ISP</td>
</tr>
<tr>
<td>live.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>hi5.com</td>
<td>Social Networking</td>
</tr>
</tbody>
</table>

Table 5: Top web account credentials sent by Torpig victims.

65 minutes by using permutation, substitution, and other simple replacement rules used by the password cracker (the "single" mode). Another 14,000 passwords were recovered in the next 10 minutes when the password cracker switched modes to use a large wordlist. Thus, in less than 75 minutes, more than 40% of the passwords were recovered. 30,000 additional passwords were recovered in the next 24 hours by brute force (the "incremental" mode).

### 6.5 Privacy and Other Threats

Although the data collected by Torpig is focused on financial information, it also consists of a (sparse) trace of the online activity of the users of infected machines. This data can be readily mined by the botmaster to determine all kinds of sensitive information that pose far-reaching threats to the privacy of the victims. We were curious about how extensive the breach of the victim’s privacy was, and, therefore, we further analyzed the data. We present some of the results of the analysis plus some interesting anecdotes in this section.

We wondered about the interests, habits, and concerns of the victims, so we analyzed the messages that users of infected machines send, for example, through webcam systems, forums, and chats. Since the full content of these messages is captured by Torpig, they often contain detailed (and private) descriptions of the lives of their authors. We focused on messages written in English and longer than a given threshold (250 characters). We obtained 6,542 messages. We classified each message in one or more categories, by defining a set of keywords for each category and matching the message’s contents against these keywords. The result is what one could call the *zeitgeist* of the Torpig botnet. The victims of Torpig prepare for exams and worry about grades (5% of the messages), look for professional advice from doctors and lawyers (1%), play video games (2%), seek jobs and submit resumes (14%), are sport fans (6%), discuss money (7%), trade goods online (4%), exchange insults (0.1%), and look for sex or partners online (4%).

We noted that online security is a concern of the infected population (almost 10% of the messages mention phishing, viruses, and spyware), but only a few people seem to suspect that they are using an infected machine. On the contrary, there is one who unwittingly announces that his computer has been fixed and the virus is gone. We looked for reactions to the phishing mechanism used by Torpig. We found it mostly among eBay and PayPal customers. In We looked for reactions to the phishing mechanism used by Torpig.

### Table 5: Top web account credentials sent by Torpig victims.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Account Credentials Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>google.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>facebook.com</td>
<td>Social Networking</td>
</tr>
<tr>
<td>myspace.com</td>
<td>Social Networking</td>
</tr>
<tr>
<td>netlog.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>libero.it</td>
<td>Search/Email/ISP</td>
</tr>
<tr>
<td>yahoo.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>nasza-klasa.pl</td>
<td>Social Networking</td>
</tr>
<tr>
<td>alice.it</td>
<td>Search/Email/ISP</td>
</tr>
<tr>
<td>live.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>hi5.com</td>
<td>Social Networking</td>
</tr>
</tbody>
</table>

Table 5: Top web account credentials sent by Torpig victims.

65 minutes by using permutation, substitution, and other simple replacement rules used by the password cracker (the "single" mode). Another 14,000 passwords were recovered in the next 10 minutes when the password cracker switched modes to use a large wordlist. Thus, in less than 75 minutes, more than 40% of the passwords were recovered. 30,000 additional passwords were recovered in the next 24 hours by brute force (the "incremental" mode).

### 6.5 Privacy and Other Threats

Although the data collected by Torpig is focused on financial information, it also consists of a (sparse) trace of the online activity of the users of infected machines. This data can be readily mined by the botmaster to determine all kinds of sensitive information that pose far-reaching threats to the privacy of the victims. We were curious about how extensive the breach of the victim’s privacy was, and, therefore, we further analyzed the data. We present some of the results of the analysis plus some interesting anecdotes in this section.

We wondered about the interests, habits, and concerns of the victims, so we analyzed the messages that users of infected machines send, for example, through webcam systems, forums, and chats. Since the full content of these messages is captured by Torpig, they often contain detailed (and private) descriptions of the lives of their authors. We focused on messages written in English and longer than a given threshold (250 characters). We obtained 6,542 messages. We classified each message in one or more categories, by defining a set of keywords for each category and matching the message’s contents against these keywords. The result is what one could call the *zeitgeist* of the Torpig botnet. The victims of Torpig prepare for exams and worry about grades (5% of the messages), look for professional advice from doctors and lawyers (1%), play video games (2%), seek jobs and submit resumes (14%), are sport fans (6%), discuss money (7%), trade goods online (4%), exchange insults (0.1%), and look for sex or partners online (4%).

We noted that online security is a concern of the infected population (almost 10% of the messages mention phishing, viruses, and spyware), but only a few people seem to suspect that they are using an infected machine. On the contrary, there is one who unwittingly announces that his computer has been fixed and the virus is gone. We looked for reactions to the phishing mechanism used by Torpig. We found it mostly among eBay and PayPal customers. In We looked for reactions to the phishing mechanism used by Torpig.

### Table 5: Top web account credentials sent by Torpig victims.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Account Credentials Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>google.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>facebook.com</td>
<td>Social Networking</td>
</tr>
<tr>
<td>myspace.com</td>
<td>Social Networking</td>
</tr>
<tr>
<td>netlog.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>libero.it</td>
<td>Search/Email/ISP</td>
</tr>
<tr>
<td>yahoo.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>nasza-klasa.pl</td>
<td>Social Networking</td>
</tr>
<tr>
<td>alice.it</td>
<td>Search/Email/ISP</td>
</tr>
<tr>
<td>live.com</td>
<td>Search/Email</td>
</tr>
<tr>
<td>hi5.com</td>
<td>Social Networking</td>
</tr>
</tbody>
</table>
online bank account login credentials [31]. Furthermore, Franklin et al. classified and assessed the value of compromised credentials for financial and other personal information that is bought and sold in the underground Internet economy [7]. Unlike these studies, we analyzed live data that was sent directly to us by bots. This allows us to gain further insights, such as the timing relationships between events.

8. CONCLUSIONS AND FUTURE WORK

In this paper we presented our experience with the takeover of a large-scale botnet. Controlling hundreds of thousands of hosts that were volunteering Gigabytes of sensitive information, provided us with the unique opportunity to understand both the characteristics of the botnet victims and the potential for profit and malicious activity of the botnet creators.

There are a number of lessons learned from the analysis of the data we collected, as well as from the process of obtaining (and losing) the botnet. First of all, we found that previous evaluations of botnet sizes based on the count of distinct IPs might be grossly overestimated. In particular, we found that, in our case, the number of unique IP was one order of magnitude larger than the actual number of infected hosts.

Second, the victims of botnets are users with poorly maintained machines that choose easily guessable passwords to protect access to sensitive sites. This is evidence that the malware problem is fundamentally a cultural problem. Even though people are educated and understand well concepts such as the physical security and the necessary maintenance of a car, they do not understand the consequences of irresponsible behavior when using a computer. Therefore, in addition to novel tools and techniques to combat botnets and other forms of malware, it is necessary to better educate the Internet citizens so that the number of potential victims is reduced.

Another insight obtained from the experience of taking over the botnet was that interacting with registrars, hosting facilities, victim institutions, and law enforcement is a rather complicated process. In some cases, simply identifying the point of contact for one of the registrars involved required several days of frustrating attempts. We are sure that we have not been the first to experience this type of confusion and lack of coordination among the many pieces of the botnet puzzle. However, in this case we believe that simple rules of behavior imposed by the US government would go a long way toward preventing (or sanctioning) obviously-malicious behavior. Even though botnets are a global problem, the United States could effectively enforce rules of behavior that might make it harder for the botmaster to use the nation’s cyber infrastructure with impunity.

Acknowledgments

The research was supported by the National Science Foundation, under grant CNS-0831408.

We would like to acknowledge the following people and groups for their help during this project: David Dagon, MELANI/Gov-CERT.ch, and the Malware Domain List community.*

9. REFERENCES


*Email: cert@melani.admin.ch, URL: http://www.melani.admin.ch/index.html?lang=en


[38] The Spamhaus Project. ZEN. http://www.spamhaus.org/zen/.


APPENDIX

A. MAN-IN-THE-BROWSER PHISHING ATTACKS

Figure 10: A man-in-the-browser phishing attack.

Figure 10 is a screen-shot taken during a man-in-the-browser phishing attack. The user has just logged in into her account at the online banking web site of Wells Fargo. Torpig injects in the browser a form that asks for the user’s passcode, date of birth, social security number, mother’s maiden name, credit card number, expiration data, validation code (CVV2), and ATM PIN. Notice that the URL correctly points at https://online.wellsfargo.com/signon, the SSL certificate has been validated and the address bar displays a padlock, and that the page has the same style of legitimate pages on the original web site.

B. PASSWORD STRENGTH

Figure 11: Number of passwords cracked in 90 minutes by the John the Ripper password cracker tool. Vertical lines indicate when John switches cracking mode. The first vertical line represents the switching from simple transformation techniques (“single” mode) to wordlist cracking, the second from wordlist to brute-force (“incremental”).