An Empirical Analysis of Malicious Internet Banking Software Behavior

André Ricardo A. Grégio
CTI Renato Archer
Campinas, SP, Brazil
argregio@cti.gov.br

Vitor Monte Afonso
University of Campinas
Campinas, SP, Brazil
vitor@las.ic.unicamp.br

Victor Furuse Martins
University of Campinas
Campinas, SP, Brazil
furuse@gmail.com

Dario Simões Fernandes
University of Campinas
Campinas, SP, Brazil
dario@las.ic.unicamp.br

Paulo Lício de Geus
University of Campinas
Campinas, SP, Brazil
paulo@las.ic.unicamp.br

Mario Jino
University of Campinas
Campinas, SP, Brazil
jino@dca.fee.unicamp.br

ABSTRACT

"Bankers" are special types of malware whose targets are Internet banking users, mainly to obtain their credentials. Banker infections cause losses of billions of dollars worldwide. Thus, better understanding and detection of bankers is required. Due to their interactive nature, obtaining bankers' behaviors can be a difficult task for current dynamic analyzers. Also, existing tools specially crafted to detect bankers are usually limited to a specific type. In this article, we propose BanDIT, a dynamic analysis system that identifies behavior related to bankers combining visual analysis, network traffic pattern matching and filesystem monitoring. We analyzed over 1,500 malware samples to identify those whose target were online banks and reported the compromised IP and e-mail addresses found. We present an evaluation of their behavior and show that BanDIT was able to identify 98.8% of bankers in a manually labeled banker samples set.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Abuse and crime involving computers*

Keywords

Malicious software, Internet Banking security

1. INTRODUCTION

Crimeware, phishing Trojan or banking Trojan are terms used to refer to a special type of malware whose main objective is to obtain online banking credentials in order to steal money [7]. These malware samples, commonly known as bankers, make use of social engineering techniques and phishing e-mail messages to trick users into providing their Internet banking credentials, credit or debit card numbers, and security token values to take over the victim's online bank account or to resell these pieces of information in undergound markets. The high success rate of such attacks results in billions of dollars in losses worldwide [13, 9].

Recent examples of widespread bankers are ZeuS [4] and Spyeye [6], which infect users by modifying files and system libraries, and by injecting their code into system processes. Conversely, Brazilian bankers usually infect users' machines by falsely warning them that they need to update their Internet banking defense mechanisms or to confirm their authentication data. This type of banker has been seen in the Brazilian Internet space for at least ten years: in late 2006, they amounted to 30.7% of the observed bankers in [7]. More recently, a report from Kaspersky [10] pointed Brazil as the country most frequently targeted by bankers.

Brazilian bankers usually rely on phishing messages to propagate, presenting attachments or links that lead to the download of an executable file, which in turn presents a window to the user that resembles a legitimate browser window. Also, Brazil has about 28 million online bank users and there are no specific laws regarding computer-based crimes, turning the scenario into a favorable one for cyber criminals. This situation created an underground commerce for selling credit card numbers and do-it-yourself banker kits (offered at social networks, such as Orkut) that allow for quick compiling of customized variants. Thus, Brazilian bankers are an interesting topic due to their prevalence, to the threat they pose to a large base of users and to the infection features that differentiate them from other bankers. Therefore, studying the behavior of Brazilian bankers is an important requirement in order to allow for the development of effective counter measures against them.

Despite previous works [1, 5, 7] regarding the detection of bankers, there is a lack of initiatives to automatically identify this type of malware during the dynamic analysis in publicly available systems (e.g., Anubis [3], CWSandbox [14]). However, these systems are largely used by incident response teams around the world as a first step to provide information about a malware incident during the response procedures. Hence, the identification of a banker right on the dynamic analysis step should yield a more efficient development of protective measures, such as sending alerts to ISPs, deeper investigation of the banker's file and execution behavior to find out information about the possible damage extent, pro-

SAC'13 March 18-22, 2013, Coimbra, Portugal.
Copyright 2013 ACM 978-1-4503-1656-9/13/03 ...$15.00.

duction and deployment of blocklists, and early warning to law enforcement authorities. To this end, we propose BanDIT, a dynamic analysis system that provides identification of bankers and tracking of servers used in banker infections.

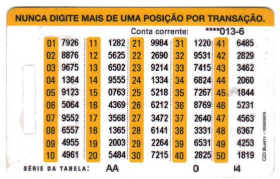
In spite of some of the techniques presented on this paper not being exactly novel, we are not aware of any other work that (i) yields the combined use of these techniques, (ii) applies them to identify bankers as a complement to dynamic analysis system reports, and (iii) focuses on the identification of Brazilian bankers based on their traits. Thus, the main purpose of BanDIT is to aggregate value to the dynamic analysis of malware by identifying bankers that behave like Brazilian ones, making it possible to screen them out for further analysis. The main contributions of this paper are as follows:

- We proposed a scheme to analyze and identify malware samples that act as bankers through a combination of visual similarity identification, network signature matching and file system change monitoring.
- We implemented a prototype of our proposed system (BanDIT) and analyzed over 15 hundred unique malware samples, mostly from phishing e-mail messages. Furthermore, we leveraged an empyrical analysis of the execution behavior presented by the bankers that were identified through BanDIT.
- We reported all the e-mail and IP addresses/domains that were involved in bankers infections (used for sending/receiving stolen credentials, or as a download base for other malware pieces). We also developed a Web site for tracking infected IP addresses related to malware samples that attack Internet banking users.

2. BACKGROUND

Internet banking access brought convenience to users, allowing them to perform money transfers, payments and account status verification without going physically to a bank agency. However, Internet access is also convenient to attackers, as they can subtract money funds anonymously while avoiding getting shot or arrested. To minimize risks, banks deploy security mechanisms intended to protect online transactions, which range from mailed password tables to hardware tokens (Figure 1). Together with the bank card passwords, these mechanisms provide additional authentication factors, making it harder to forge a valid transaction.

(a) Hardware token


NUNCA DIGITE MAIS DE UMA POSIÇÃO POR TRANSAÇÃO.

Conta corrente: ****013-6

01	7926	11	1282	21	9984	31	1220	41	6485
02	8876	12	5625	22	2690	32	9531	42	2829
03	9675	13	6502	23	9214	33	7415	43	3462
04	1364	14	9555	24	1334	34	6824	44	2060
05	9123	15	0763	25	5218	35	7267	45	1844
06	5064	16	4369	26	6212	36	8769	46	5231
07	9552	17	3568	27	3472	37	2640	47	4563
08	6557	18	1365	28	6141	38	3331	48	6367
09	4955	19	2003	29	2264	39	6531	49	4253
10	4961	20	5484	30	7215	40	2825	50	1819

SÉRIE DA TABELA: AA 0 4

(b) Password table

Figure 1: Internet banking auth. factors

Hence, to bypass those additional security mechanisms and to access the victim's balance, attackers need to know several pieces of information that an online bank uses to authenticate its clients (e.g., a PIN, a token value etc). Brazilian banks usually provide password tables or hardware tokens to some of their users as another authentication factor for Internet banking use. Both of them are described below.

Hardware Tokens. The purpose of a "security" token is to prevent attackers that obtained credentials of Internet banking clients to perform financial transactions. This is accomplished by forcing the client to provide a value that is available in the device's display to complete an online transaction. Although this type of device serves as an OTP generator, the complexity and costs of deployment make it prohibitive to be widely adopted by Brazilian banks.

Password Tables. A cheaper approach is the deployment of password tables, whose purpose resembles that of hardware tokens. However, the "positions" of those tables are reused along time for different transactions, and even in the same Internet banking session. As a result, table substitution may not occur frequently enough.

2.1 Anatomy of Banker Infections

Banker attacks affect both server and client sides. On the server side, possible actions that attackers can take are: i) to register "mistyped" bank domains and deploy sites that clone the legitimate Internet banking site, in order to deceive careless users and ii) to compromise vulnerable servers/sites, so they can host a clone of the targeted Internet banking site or the banker executable. Furthermore, attackers may spread phishing messages containing attached banker executables, or links to fake/compromised sites (to directly grab information using cloned interfaces or to download the banker).

On the client side, bankers leverage a myriad of techniques to mislead users, such as the overlapping of Internet banking form fields, the interception of network communications, the modification of name resolution and proxy configuration files, the sending of e-mail messages warning about updates to the Internet banking security solution installed on the victim's systems, the use of keyloggers that filter Internet banking content to log only important data and so on. In the last few years, we have been observing that most of the Brazilian bankers infect users through sending of phishing content that leads to the download of an executable. Once downloaded and installed, this executable presents a GUI (which may simulate a browser) that "guides" the victim into providing sensitive information (e.g., bank agency, account number, credit card number and verification code, Internet banking password, token values, password table values etc.)

This trend may be due to the fact that users can easily realize the fraud in infections where the attacker supplies a (weird) link to a cloned site, such as `http://something-in-another-country/http/www.mybank/index.php`. Apparently, it makes more sense to the user to download and execute files named `<bankname>-iToken.exe` or `<bankname>-security-update.exe`. Although Brazilian banks claim that they do not send messages regarding either sensitive data and security mechanisms, or asking for their clients' information, banker attacks have been and are still being performed successfully on a steady pace.

Other ways to inadvertently redirect the user to a cloned banking site may involve either the modification of the system "hosts" file, or the loading of a PAC (proxy auto-config) file. The "hosts" file maps hostnames to IP addresses, being the starting point to resolve a hostname query on the victim's system. Malware can modify this file to map known Internet banking URLs to IP addresses that are hosting cloned sites containing forms to steal users' credentials. The other method forces the browser to load a PAC file, which defines the proxy that a Web browser should use to access a given URL through the JavaScript function `FindProxyForURL(url, host)`. A PAC file created on the victim's system provides IP addresses that might lead to cloned sites, thus allowing us to identify owned Internet servers.

Based on the observation of the bankers' common behavior, we defined a (non-exhaustive) set of potentially dangerous activities that may be used to identify them. These activities are briefly described below:

- Stealing of user credentials or financial information, and of system or user data, such as username, machine name, hard disk serial number, O.S. version etc.
- E-mail sending from the infected system, to send out sensitive data or to deliver spam to extend the attack by futher propagation.
- Subversion of Internet browsing through the modification of the name resolution file (hosts) or the browser proxy using PAC files.
- Presence of images related to Internet banking sites, as banks do not send executable files to their clients (at least in Brazil), as well as an autonomous program not being supposed to have embedded bank images.

3. BANDIT OVERVIEW

For the purpose of identifying bankers during the dynamic analysis of malware, BanDIT (Banker Detection and Infection Tracking) combines visual similarity, network traffic pattern matching and file system monitoring techniques. Although BanDIT was conceived to identify bankers, it behaves like a standard dynamic analysis system. Thus, to develop BanDIT, we developed a Windows kernel driver that traces the execution of an executable and the processes it interacts with (through writing into their memory area or spawning children processes). This driver hooks the Windows SSDT (System Service Dispatch Table) and logs the system calls related to file writing and deletion, process creation and termination, registry creation, writing and removal, memory writing, and network connecting, sending, receiving and disconnecting.

Therefore, our driver may be installed inside virtual, emulated or bare-metal environments, assuming the operating system to be Windows XP. Inside the monitoring environment, we also developed a component based on *AutoIt*[1] to interact with screens and error popups, and to take screenshots. Moreover, we capture the network traffic off the analysis environment, so that payloads are reliably obtained. We also use *Foremost*[2] to extract the images or screens that are embedded in a banker binary file. An overview of BanDIT is shown in Figure 2. BanDIT's identification process relies on three steps, which are explained as follows:

Visual Similarity. In this step, we perform the recognition of known Internet banking logos, capitalizing on the fact that attackers are restricted on the number of variations in the bank site patterns, or else phishing is bound to fail. Thus, it is important that an attacker keeps the logo positions and colors to successfully lure the user. To accomplish the visual identification, we obtain figures from the banker binary file, from network connections that download images (even from the legitimate Internet banking site), and from windows presented by the banker during its execution. Gathering images from these sources increases our chances to obtain them if the banker avoids screenshots, ciphers the traffic or its binary is specially packed. To verify if a malware extracted image contains bank logos, we use JavaCV[3], an interface to the OpenCV[4] library that contains a class to search for an object inside an image using the Speed-Up Robust Features algorithm [2]. To do so, we developed a component that searches for the logos of the five major Brazilian banks within the extracted images. Moreover, we submit all images to Tesseract[5], an Optical Character Recognition engine. We then search the resulting text for keywords present in our specially crafted banker-related terms dictionary.

Network Traffic. We have been observing that, within our samples, bankers use to send stolen data from the victim's system through HTTP requests. The analysis of these requests revealed some patterns that allowed us to develop "network signatures" to verify for banker-related activities in the network traffic that is captured during the dynamic malware analysis. Listing 1 exemplifies the contents of a POST request related to a banker infection. We verified that some bankers send this type of request as soon as they are executed, followed by a similar one containing Internet banking information provided by a victim.

Listing 1: A banker POST request.

```
praquem=XXXX@gmail.com&titulo=
    NOME_DA_MAQUINA INFECT&texto=Memo1
DE$CO CLIENTE
+ 1 abriu modulo
```

The network traffic analysis step also allows us to obtain PAC files, images, URLs, e-mail and IP addresses of abused servers, which are used in our identification process and reported to competent authorities.

File System. To detect changes that bankers commonly perform in infected systems, such as the inclusion of PAC files, modifications to the "hosts" file, and attempts to disable security mechanisms (AVs, firewall, automatic updates), we monitor malware execution using our SSDT hooking driver. We are able to obtain PAC files if bankers drop them on the victim's system or by downloading them from URLs that are set as values in the registry `Internet Settings\ AutoConfigURL` key. To evaluate the obtained PAC files and to check for redirection of bank site URLs, we execute the PAC code using a JavaScript processor based on Rhino[6] and call the produced *FindProxyForURL* function using Internet banking domains as parameters. Thus, even if the JavaScript code found inside the PAC file is obfuscated, it will eventually deobfuscate itself and *FindProxyForURL* will be called. Furthermore, we obtain the "hosts" file from the infected system after the dynamic analysis is done, which is then searched for Internet banking domains.

3.1 Data Extraction and Dynamic Execution

The previous process of obtaining logos, creating a banker-related terms dicitionary, and developing network/file sys-

[1]http://www.autoitscript.com
[2]http://foremost.sourceforge.net/

[3]http://code.google.com/p/javacv/
[4]http://opencv.org/
[5]http://code.google.com/p/tesseract-ocr/
[6]http://www.mozilla.org/rhino

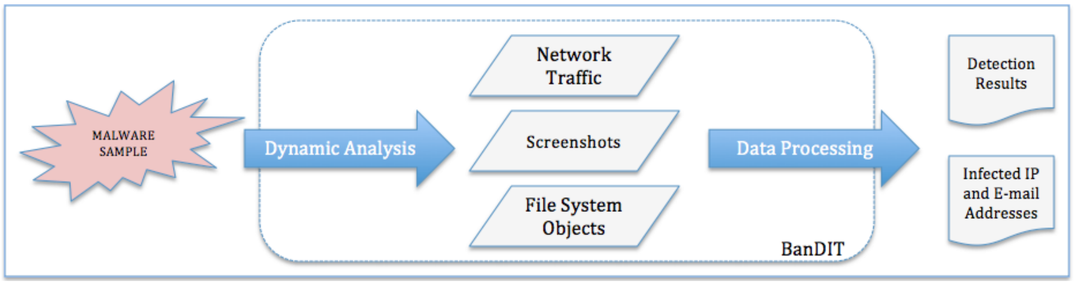


Figure 2: Overview of BanDIT.

tem signatures involved the manual analysis of 1,194 malware samples obtained from phishing (and analyzed) between August, 2010 and July, 2011.

For our current tests, we selected 1,653 malware samples mainly from phishing messages and partners/contributors, all of them obtained in 2012. We submitted these samples to BanDIT, which runs on a virtualized Windows XP with Service Pack 3 for four minutes. This process includes the external capture of network traffic and the extraction of images from the binary file using Foremost. Moreover, we scanned all samples with the Avira antivirus engine to obtain their assigned labels. We normalized these labels to focus on the type/family of the malware sample. Table 1 shows the distribution of samples among malware families.

4. EMPIRICAL EVALUATION

From Table 1, ≈ 23% of the 1,653 samples are not detected as malware by the antivirus engine (ID = 15), and ≈ 10% of the samples (ID = 16) are distributed among 63 distinct families. The remaining of the samples is divided into the 14 families with highest incidence. Below, we discuss the behaviors found in the samples, map their families and leverage BanDIT identification results.

Table 1: Distribution of malware samples among Avira-assigned families.

AV ID	AV Label	Total
01	Agent	2.2%
02	Atraps	6.4%
03	Banker	17.7%
04	Crypt	9.1%
05	Delf	2.7%
06	Delphi	5.8%
07	Downloader	1.1%
08	Gen	3.3%
09	Graftor	2.5%
10	Offend	3.0%
11	Spy	5.4%
12	Vb	4.4%
13	Virtool	1.0%
14	Zusy	2.0%
15	NOT DETECTED	23.2%
16	Other families	10.2%

4.1 Observed Behavior

Before testing BanDIT, we manually analyzed our samples and searched for the set of behaviors defined in Section 2.1. We found 1,520 samples that behaved like bankers. Figure 3 shows the amount of samples that presented each behavior. Information stealing is the most frequent behavior found in our dataset, followed by PAC loading, e-mail sending and the presence of bank-related images.

Table 2 associates the malware family IDs to each observed behavior. One can see that samples labeled as "bankers" presented all of the selected behaviors: information stealing, sending e-mail messages, loading PAC files, modifying the "hosts" file and carrying bank logos, as expected. However it is worth mentioning that, except for "Hosts Changing" and "Bank Images", all other banker-related behaviors were presented by all AV-assigned families, making the AV classification rather dull. Interestingly, bank-related images found in samples do not correlate to the specific "banker"AV label.

Table 2: Behaviors presented (columns) by malware families (rows) under manual inspection: IS = Information Stealing; ES = Email Sending; HC = Hosts Changing; PL = PAC Loading; BI = Bank Images

	IS	ES	HC	PL	BI
01	✓	✓		✓	✓
02	✓	✓		✓	✓
03	✓	✓	✓	✓	✓
04	✓	✓	✓	✓	✓
05	✓	✓		✓	✓
06	✓	✓		✓	✓
07	✓	✓		✓	
08	✓	✓		✓	✓
09	✓	✓		✓	✓
10	✓	✓		✓	✓
11	✓	✓		✓	✓
12	✓	✓		✓	✓
13	✓	✓		✓	
14	✓	✓		✓	✓
15	✓	✓		✓	✓
16	✓	✓	✓	✓	✓

4.2 BanDIT Results

To conclude our study about banker behavior, we evaluate BanDIT's results and compare our findings to the manual analysis previously provided. The main advantage of BanDIT is to aggregate different techniques into one component so as to identify banker-related behavior. Figure 4 shows how many of our 1,520 manually labeled samples (as bankers) were identified by each of the BanDIT's techniques. BanDIT was able to identify banker-related behavior in 1,502 (98.8%) samples. Network pattern matching was the most effective, but filesystem modification monitoring and image analysis were also important, since that both (combined) identified 539 (35.5%) samples.

Infection Tracking. During BanDIT's evaluation, we were able to identify 88 e-mail addresses that were compromised or specifically created to receive stolen data from banker's infected victims. Furthermore, we pinpointed 183 IP addresses of attacker-owned/compromised servers that have been used to host malware pieces, such as PAC files, or to act as fake bank sites. We found that 77.1% of the samples that evaded information through the network ended up contacting IP addresses from Brazil, United States and Germany. Additionally, 80.7% of the IP addresses used in PAC files and "hosts" files are from United States, Brazil and France. The e-mail accounts used by the bankers were notified to the corresponding e-mail providers and the IP addresses were notified to the competent authorities.

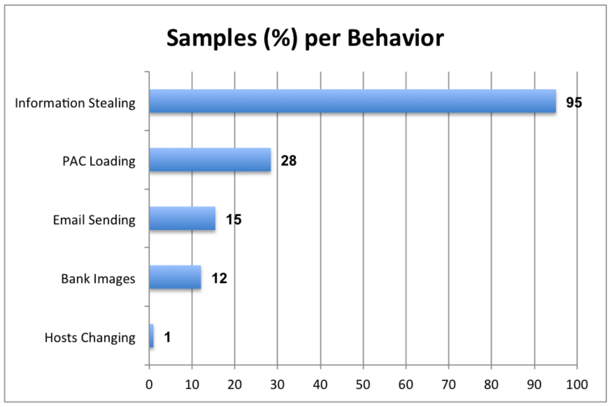


Figure 3: Percentage of banker samples per presented behavior.

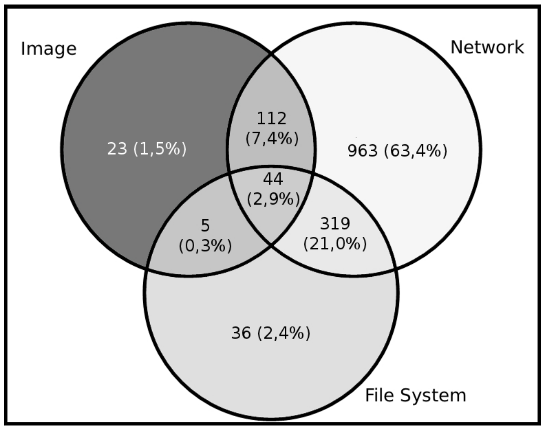


Figure 4: BanDIT's identification results

4.3 Discussion

In our analysis, we considered that banks do not send executable files through e-mail to their clients, as major Brazilian banks claim on their Web sites. The comparison of our manually obtained results to those produced by BanDIT shows that our system identified information stealing patterns on 94.6% of the samples, which stands pretty well to the 95.1% obtained by the manual process. This difference may be due to user interactions that might be required by the sample but were not done during the BanDIT analysis. Regarding the visual similarity searching, BanDIT found Internet banking related images on 12.1% of samples, the same amount that we found manually. This shows that the combination of logo searching and keyword matching in the text obtained by an OCR was very effective. BanDIT's filesystem monitoring led to the identification of changes in the "hosts" file or loading of PAC files in 26.6% of our samples; in contrast, manual inspection identified this type of behavior in 29.1% of the samples (PL+HC). In this case, the difference may be caused by problems during the sample execution, such analysis timeout or the need for human interaction. Finally, our manual search for e-mail sending allowed us to find compromised servers and e-mail addresses.

Although we chose not to address ZeuS and SpyEye samples, since there already are related work addressing them (e.g., [5], [1]), we also tested the `deBank` tool[7]. It is intended to detect banker variants from some known families, such as SpyEye and ZeuS through matching of certain patterns found in the infected system's memory. We executed, one at a time, around 70 SpyEye and ZeuS samples obtained from trackers[8], and then launched the tool. Unfortunately, none of them was detected by `deBank`, maybe because they are newer than the last available version of the tool. This test was interesting as it corroborated the importance of screening out banker samples for further analysis, since traditional detection tools can be easily bypassed. We conclude that behavior-based approaches, such as BanDIT, may be useful to help develop counter-measures, although they are not intended for real-time detection. In addition, BanDIT identification components can be easily linked to available dynamic analysis systems.

5. LIMITATIONS AND FUTURE WORK

Every signature-based scheme is prone to be bypassed once attackers find out the signature generation process. This causes an arms race between security people and cybercriminals, requiring this type of system to be constantly updated. Another limitation that our system suffers is related to evasion techniques that affect all dynamic malware analysis systems, (e.g., waiting for a certain number of reboots before presenting the malicious behavior). However, since we are able to extract images from the sample file, our visual similarity technique may identify a banker right away. Due to our observations that most Brazilian bankers send the stolen information unencrypted, we currently do not handle encrypted network traffic. Thus, the current implementation would not be able to identify stolen data sent by an encrypted channel. Also, rootkit-based malware could potentially subvert our filesystem monitoring component, however only two samples of our dataset loaded drivers and one of them was identified by BanDIT as a banker.

Therefore, the use of more than one detection method is important because it allows the system to identify a malicious beheavior even when one or two of the employed methods fail. Future works include making the infection tracking system available to the public, addressing encrypted network traffic and extending our network protocol coverage.

6. RELATED WORK

[7]`http://www.fitsec.com/tools/DeBank.exe`

[8]http://zeustracker.abuse.ch, http://spyeyetracker.abuse.ch

The financial motivation behind malware infections justifies (from the standpoint of a cybercriminal) the increasing spread of bankers. Next, we present some research works regarding banker detection.

In [7], the author discusses the behavior of bankers targeting German, Swedish and Brazilian Internet banks, and the mechanisms employed by banks to protect their customer accounts, such as one-time-passwords (OTP) and transaction numbers (TAN). He proposes a tool called "Mstrings", which searches the memory of an analysis system (during a malware sample's execution) for known Internet banking-related strings. Although this scheme is effective against bankers that act as keyloggers when the user is accessing the Internet bank site, it may be ineffective against bankers that rely on social engineering and that do not depend on the user access to the Internet bank site.

Buescher et al. [5] present a banker detection scheme that checks for hooks on Internet Explorer to steal user's information. To do so, the authors developed BankSafe, a component that verifies for API changes during Internet Explorer execution. After a malware sample runs in a controlled environment, BankSafe checks for hooking fingerprints to detect if some type of hook was installed on Internet Explorer during the analysis time. The authors claim that BankSafe produces very good detection results, however it is limited to detecting bankers that install hooks.

In [11], the authors present an approach to detect phishing Web pages, which is based on a visual detection scheme that lets the user know beforehand that his/her data is being intercepted. The phishing detection is based on three features extracted from the analyzed Web pages: the text portion, the images and the visual appearance of the Web page. The problem is being able to detect only phishing attempts that are based on fake Web pages accessed during a user's browsing session, missing bankers that forge an Internet bank site through windows from their own executable file (such as Brazilian ones).

Botzilla [12] uses a network signature detection scheme that is closely related to part of our work, since we also perform our network traffic detection step based on invariant content patterns. Botzilla's goal is to detect malware "phoning home", i.e. contacting an attacker controlled site to send information. Botzilla's signatures address some malware classes whose samples were collected from a large university network, yielding a detection rate of about 94.5%.

In [8], Holz et al. analyze the underground economy of stolen credentials and the phenomena of keyloggers that communicate with criminals through dropzones, i.e. publicly writable directories on attacker-owned servers, which serve as exchange points for the keylogger's collected data. Their analysis is focused on Limbo (a Browser Helper Object for Internet Explorer) and ZeuS (a keylogger attached to spam) samples. To automate the keylogger analysis, they developed *SimUser*, a tool based on *AutoIt* that simulates arbitrary user behavior—keystrokes, mouse movement, manipulation of windows—according to predefined templates.

7. CONCLUSION

As more sophisticated bankers appear, the need for countermeasures and fast response increases. Thus, it is important to identify and extract relevant information about bankers so that they can be screened out during dynamic analysis for further and deeper investigation. In this paper, we introduced BanDIT, a system that dynamically analyzes malware in order to identify bankers. BanDIT's identification process combines visual similarity searching, network traffic pattern matching and filesystem modification monitoring. The evaluation of BanDIT shows that it is effective to screen bankers out (98.8% of correct identifications) of a dataset of general malware. Moreover, BanDIT helps incident responders to warn victims in a timely manner, to notify infected ISPs and servers and to create blacklists that can be quickly deployed to avoid more banker infections to take place.

8. REFERENCES

[1] Tool release: A banking trojan detection tool. `www.fitsec.com/blog/index.php/2011/08/15/tool-release-a-banking-trojan-detection-tool`, 2011.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, pages 404–417, 2006.

[3] U. Bayer, C. Kruegel, and E. Kirda. TTAnalyze: A Tool for Analyzing Malware. In *15th Annual Conference of the European Institute for Computer Antivirus Research (EICAR)*, 2006.

[4] H. Binsalleeh, T. Ormerod, A. Boukhtouta, P. Sinha, A. Youssef, M. Debbabi, and L. Wang. On the Analysis of the Zeus Botnet Crimeware Toolkit. In *Eighth Annual International Conference on Privacy Security and Trust (PST)*, pages 31–38, August 2010.

[5] A. Buescher, F. Leder, and T. Siebert. Banksafe information stealer detection inside the web browser. In *Proceedings of the 14th international conference on Recent Advances in Intrusion Detection*, RAID'11, pages 262–280. Springer-Verlag, 2011.

[6] P. Coogan. Spyeye bot versus zeus bot. `http://www.symantec.com/connect/blogs/spyeye-bot-versus-zeus-bot`, 2010.

[7] F.-S. Corporation. The trojan money spinner, 2007. Available at `http://www.f-secure.com/weblog/archives/VB2007_TheTrojanMoneySpinner.pdf`.

[8] T. Holz, M. Engelberth, and F. Freiling. Learning more about the underground economy: a case-study of keyloggers and dropzones. In *Proceedings of the 14th European conference on Research in computer security*, ESORICS'09, pages 1–18. Springer-Verlag, 2009.

[9] C. Ilioiu. What is trojan banking, computer virus designed to attack online transactions. `http://www.financial-magazine.org/what-is-trojan-banking-computer-virus-designed-to-attack-online-transactions-66473.html`, 2012.

[10] Kaspersky. Number of the week: 780 new malicious programs designed to steal users' online banking data detected every day, 2012. Available at `http://www.kaspersky.com/about/news/virus/2012/Number_of_the_week_780_new_malicious_programs`.

[11] E. Medvet, E. Kirda, and C. Kruegel. Visual-similarity-based phishing detection. In *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, SecureComm '08, pages 22:1–22:6. ACM, 2008.

[12] K. Rieck, G. Schwenk, T. Limmer, T. Holz, and P. Laskov. Botzilla: detecting the "phoning home" of malicious software. In *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC)*, pages 1978–1984, 2010.

[13] UPI. New malware attacks target online banking. `http://www.upi.com/Science_News/2012/02/02/New-malware-attacks-target-online-banking/UPI-25351328224292/`, 2012. Accessed on May, 2012.

[14] C. Willems, T. Holz, and F. Freiling. Toward automated dynamic malware analysis using cwsandbox. *IEEE Security Privacy*, 2007.