The Art of Cyber Bank Robbery

Stealing Your Money Through Insidious Attacks

Aditya K. Sood, Michigan State University
Richard J. Enbody, Michigan State University

Abstract. Cyber criminals are using advanced attacks to exploit online banking systems and services to covertly steal money. This paper describes the tactics currently used by cyber criminals to conduct cyber bank robbery.

Introduction

Cyber criminals use botnets (malware) for a wide range of cyber crimes, and these attacks are increasing. The economics of e-crime and the related underground market have been studied which reveal a significant increase in online fraud [1]. Internet banking (e-banking) has transformed the economic and financial culture of the world. Over time, banks have strengthened the security of their servers to the point that attackers now target end-user systems. Server-side defenses are easier for banks because the banks have control over their servers. As client computers are outside of the banks' control, this makes it harder for the banks to subvert insidious attacks conducted on end-user systems. Due to this reason, Internet-based threats are posing security challenges to online banking. Given the increasing sophistication of attacks on the client side, it is imperative to build robust protection mechanisms on the client side that can be managed from the server side.

Necessity is the mother of invention. This aphorism applies to the current creativity of cyber criminals. Ever more sophisticated defenses have spurred attackers to develop more advanced attacks. The resulting innovative system-exploitation tactics exfiltrate data from infected clients around the world. The web browser is the primary user interface to the Internet and thus is a centralized target for attacks. The attackers design sophisticated client side malicious code that subverts a browser's functionality to harvest credentials and to perform money transfers on-the-fly in a hidden manner. The fact that these attacks are designed and structured around browsers shows how critical it has become to secure browser software.

Today, the most common platform for broad attacks on banking is via botnets. Those attacks are causing significant losses both in fraud and in defensive costs. Selling and renting botnet frameworks are an integral part of the underground economy's revenue model. Hundreds of millions of dollars are earned by cyber criminals, and billions of dollars are expended keeping those losses in check.

In 2009, Cormac and Dinei [2] conducted a study on the economics of the underground economy and estimated that a botnet herder earns approximately $0.50 per machine per year. For a botnet of 50,000 machines, a botnet herder could earn approximately $25,000. Recent botnets such as Zeus, SpyEye, and Citadel have infected millions of machines. If the same formula is applied, potential earnings are in millions of dollars every year. Some income comes from renting out the infected machines, but there are also Pay Per Infection (PPI) services where bot herders charge customers to distribute malware for a fee across their botnet. PPI rates vary significantly depending on where targeted machines are located. For example, $130 to $150 is charged per 1,000 machines to load malware on computers located in the U.S., but the rate is as low as $3 to $5 for locations in Asian countries such as China. In either case, providers of PPI services can earn millions of dollars annually.

On the defensive side, Anderson et al. in their study of cyber crime [3] pointed out that botnet mitigations cost $ 3.2 billion for anti-virus software alone. Globally, the study estimated that companies spend roughly $10 billion annually to provide defenses against cyber crimes. In addition, they projected that total global law enforcement expenditures were approximately $400 million for cyber crime. The study also concluded that global online banking fraud losses were close to $300 million, and to prevent additional frauds, banks spent approximately $1 billion. Florencio and Herley of Microsoft Research [21] found that credentials are offered in the underground market at $0.05 on the dollar value of the account. It leads them to observe that converting credentials to cash is the hard part and only a few stolen credentials result in actual theft. They analyze that the biggest cost comes from defensive costs and Anderson's data supports that conclusion.

In this paper, we present the cyber bank robbery model that is used by cyber criminals to conduct online frauds using automated exploitation frameworks such as botnets. This model is used for attacking end-user systems and mobile platforms.

Overview and Threat Model

Skilled cyber criminals are responsible for the majority of online bank fraud. The attack process can be outlined as follows:

- **Infection Entry Point and Exploitation:** A cyber criminal begins by co-opting a high-volume website to host an automated exploitation framework. That framework exploits browsers having vulnerable components using what is known as a drive-by download. The users are coerced to visit the infected website using techniques such as phishing. In addition, malicious applications can also be installed on mobile devices to control communication.
- **Data Exfiltration:** A bot is installed on the infected system that connects back to a C&C computer. For example, if the cyber criminal wants to attack Bank of America (BofA) sessions, it commands the bot to download the appropriate plugin. The bot hijacks (hooks) the communication channel initiated by the browser with the BofA website to steal account information, credentials, registered email addresses, etc. The key point is that the attack exploits client-side software, the browser in particular. Apart from that, the bots can

simply send phishing emails that exploit brand reputation of online websites and trick users to provide sensitive information. In mobile devices, apart from HTTP, SMS is used as a carrier for exfiltrating data.

- **Fraud:** Once the data is exfiltrated from the user machine, cyber criminals either sell it in the underground community or use it themselves. In advanced attacks, malicious code can execute fraudulent transactions directly from the infected systems. All these features depend on the design of bots.

This paper presents a model of cyber bank robbery structured into four phases. Phase 1 describes malware design. Phase 2 presents strategies to get malware onto users' computers and mobile devices. Phase 3 chronicles the exfiltration of sensitive data and automated transactions. Phase 4 covers the transformation of data to money. To conclude, we discuss different security mechanisms deployed by banks to combat online fraud and their shortcomings.

We use the following terminology: malware refers to any malicious code that modifies the behavior of target components. A bot is an automated malware that communicates with a remote server and performs multiple tasks in an infected system in a stealthy manner.

1. Phase 1: Malware Design

Botnets play a critical role in widespread infections on the Internet. A botnet is a network of compromised machines that are infected with bots. Bots steal sensitive information such as banking credentials from target users and have the ability to perform other nefarious tasks. The bots are sophisticated and implement advanced techniques to bypass antivirus engines and other host-based protection software [4].

Present-day bots have the capability to co-opt the communication flow in browsers through Man-in-the-Browser (MitB) attacks. These attacks enable the bots to harvest credentials using techniques such as form grabbing and web injects (explained later in this paper). In addition, the MitB attack allows the bots to make automated fraudulent transactions by exploiting the active session with the banks. Because these attacks are executed from the infected system, they are mostly hidden from the banks. MitB functionality has revolutionized the design of third-generation botnets. Since a browser is a user's window to the Internet, it is the target of attackers: controlling the browser controls the interaction. As operating systems have become hardened, attackers find attacking applications such as browsers to be easier. A detailed browser-malware taxonomy [5] exists that discusses the various classes of browser-based malware. Understanding browser-based malware is necessary to comprehend the strategies opted by malware authors to conduct stealthy attacks on the end user systems.

On similar benchmark, Man-in-the-Mobile (MitMo) attacks are conducted in mobile devices to manipulate and hijack the functionalities of installed applications. In these attacks, malicious applications use a camouflaging trick to hide their identity and trick users to believe them as authentic ones.

The cyber criminals are designing malicious code for computer systems as well as mobile platforms. The most prominent malware designs that are used in online banking frauds are discussed next.

1.1 Man-in-the-Browser (MitB) Agents

The evolution of MitB [6] attacks has given birth to advanced client-side attacks. MitB attacks are similar to Man-in-the-Middle (MitM) attacks, but exist within the operating systems to exploit browsers. MitB agents can be thought of as userland rootkits that subvert the integrity of browsers by hooking [7] selective Dynamic Link Libraries (DLL) to control the execution flow of various browser functions. When the browser calls a communication function, the hook diverts control to malicious code. This approach allows cyber criminals to conduct stealth attacks by manipulating the communication channel between browsers and the remote servers.

Hooking is an integral to many operating systems and is used frequently in Windows. In the context of browser exploits, hooking allows running processes to alter the behavior of various components in the system by intercepting the interprocess communication channel. The latest bots use inline function hooking [8] which is hard to detect because it uses hot patching and late binding, that is, the hook is actually executed during runtime. MitB agents are capable of stealing data, manipulating content and automating the critical operations without the intervention of users. Web injects and form grabbing are the two most widely used MitB techniques that implement hooking to control browser operations. These are discussed in the next sections.

1.2 Browser Rootkits

Browser rootkits [9] are defined as advanced levels of malware that hide inside browsers and perform unauthorized operations without users' knowledge. The concept of a browser rootkit originated from system rootkits that are capable of hiding and covertly interacting with the system components. Browser rootkits are malicious extensions (add ons) that use JavaScript to manipulate the content of web pages. In addition, browser rootkits can easily alter the look and feel of the web pages to fool users and trick them into performing illegitimate operations. These are also capable of altering information [10] in active sessions, account profiles, online transactions, etc. after the user successfully authenticates to an online banking website. The browser rootkits are primarily designed to execute fraudulent transactions when a user activates a session with an end server.

1.3 Man-in-the-Mobile (MitMo) Agents

With the advent of mobile technologies, cyber criminals have started targeting smart phones. Mobile platforms such as Android have been the target of cyber criminals. In the last few years, a number of mobile-based botnets have been revealed that subverted the integrity of mobile platforms to conduct attacks and exfiltrate sensitive information. For example: the existence of mobile variants of Zeus and SpyEye i.e. Zitmo and Spitmo [25] respectively show that the design of botnets is evolving with new technologies. Mobile botnets

[26] are similar to standard botnets but they aim specifically to exploit mobile architectures. Mobile bots are termed as MitMo agents that are malicious applications installed to thwart the security model of the mobile device and exfiltrate data accordingly. These are designed to control the communication channel initiated by legitimate applications with legitimate servers in a stealthy manner.

Malicious mobile applications work in conjunction with traditional botnets to subvert the multi channel protection mechanisms such as two-factor authentication (TFA) [29]. Malicious applications are designed to conduct piggybacking attacks [27] to monitor the state of target application (such as banks) and stealing information during transmission. Fake applications can also be forced to be installed on mobile devices that trick the users to provide sensitive information. On Android [30], apart from exploiting vulnerabilities, malware authors use infection techniques such as stealthy assets, infected boot images, time specific code execution, etc. to hide malicious codes. Android being open source is the preferred choice of cyber criminals. Because Blackberry and Apple use closed source operating systems, the ratio of mobile malware attacking these platforms is less than on Android.

1.4 Automated Phishing Bots

Apart from browser-based exploitation, bots are also designed to trigger phishing attacks. End users are tricked to visit illegitimate domains hosting fake web pages that appear similar to legitimate bank sites. Bots can send thousands of phishing emails at a time to a large set of users on the Internet. Honeynet [22] talks about how bots can be used to send phishing emails directly from infected computers and also from C&C panels. The phishing attacks are not new and have been in existence for years. But, the amazing part is that these attacks still exist and play a significant role in data exfiltration today. No stealthy technique is deployed during these attacks because phishing is based on social engineering to exploit the trust and knowledge of users. Botnets such as Grum and Festi [24] are specifically designed for conducting phishing attacks including spamming. On the contrary, Spamhaus [23] is an effort that is used to track botnets that send spam.

2. Phase 2: Malware Distribution

The following section is an examination of tactics chosen by cyber criminals to widely infect systems. Broad-based attacks (mass infections) have evolved over time and currently a popular technique is to drive victims to websites where they will be served malware or redirected to sites that serve malware. A target website is often a legitimate website that has been corrupted (e.g., injected with a malicious iframe) to send visitors to a malicious site. Some of the most-widely used malware distribution strategies are discussed below:

- Phishing is used to drive users to sites hosting a drive-by download attack [11]. A drive-by download attack silently exploits vulnerabilities in browser components to download malware without user action. This malware is capable of executing MitB attacks to perform fraudulent transactions and data exfiltration from the infected system. To automate the exploitation, cyber criminals have designed Browser Exploit Packs (BEPs) such as BlackHole. A browser exploit pack fingerprints the user's browser to identify vulnerabilities and then load the appropriate exploit. BEPs are sold as a crimeware service that charges buyers using a PPI model as discussed earlier.
- The popularity of Online Social Networks (OSNs) makes them attractive targets for attackers to distribute malware by exploiting trust among users. The attackers use the social network platforms and trust among "friends" to direct "friends" to malicious websites. For example, Likejacking attacks cause users to inadvertently "like" a malicious site that tricks a user to download malware.
- Bots are also distributed in traditional ways such as in warez or freeware that are downloaded from the illegitimate websites on the Internet carrying malware. Also, fake antivirus and other phony tools are still used to trick users to download malicious code.
- Bots have a built-in functionality of spreading using which they infect peripheral devices such as USBs to transmit themselves to different machines. In addition, spreaders can also infect Instant Messaging (IM) software and OSNs.
- Mobile bots and malicious applications are distributed as repackaged applications that mean the malicious code is hidden inside a legitimate application. The repackaged applications are distributed on alternate markets. Existence of vulnerabilities present in legitimate market stores also allows the attackers to host malicious applications. Other carriers include Over-the-Air (OTA) installation, mobile malvertising, etc.

Together these methods are sufficiently effective in distributing bots. The resulting zombie machines (infected systems) are managed remotely through a centralized C&C server that is owned and operated by a botmaster (or bot herder). Once a cyber criminal has controlled a set of infected computers, the next step in financial fraud is to collect credentials or conduct automated transactions.

3. Phase 3: Data Exfiltration and Stealthy Operations

Data exfiltration refers to transferring sensitive data from an infected machine to a remote C&C server. Multiple techniques exist; the most widely deployed data exfiltration and automated injection techniques used by banking malware are discussed below.

3.1 Form grabbing and Keylogging

Form grabbing is an impressive technique for extracting data present in web forms. This technique is more advanced than keylogging—a tool that results in a lot of irrelevant data that must be sifted through to find desired information such as credentials. In contrast, form grabbing grabs only the HTTP Post data sent as a part of form submission request. In particular, form grabbing greatly simplifies and automates the extraction of banking credentials making this process available for the less sophisticated criminal. However, with recent

botnets such as Citadel, both keylogging and form grabbing techniques are deployed for assurance purposes.

Form grabbing works on forms that users fill out and submit to a bank—especially forms used for logging and online transactions. As the browser is already hooked (MitB), a bot agent can easily snoop the communication channel between the client and the server. As soon as the user submits the form, the bot agent extracts the data present in the forms, generates a socket in the system and transmits the data back to a C&C server. Data in all the HTTP POST requests can be exfiltrated from the system without a user's knowledge [12].

```
set_url https://www.wellsfargo.com/* G
data_before
<span class="mozcloak"><input type="password"*</span>
data_end
data_inject
<br><strong><label for="atmpin">ATM PIN</label>:</strong> <br />
<span class="mozcloak"><input type="password" accesskey="A" id="atmpin" name="USpass"
size="13" maxlength="14" style="width:147px" tabindex="2" /></span>
data_end
data_after
data_end
```

Listing 1 - WI rule written against Wells Fargo Bank

3.2 Web Injects

Web Injects (WI) is an advanced technique of content injection. When a user submits a form and waits for a response from a web server, a bot agent is activated and starts injecting illegitimate content into the incoming HTTP responses. This process tricks the user into believing the web server has sent all of the content. WI is effective in coercing users to provide information that is otherwise not easy to attain. For example, an attacker could request a PIN, a Social Security number, or a second-channel SMS number. This attack is a variant of a MitB attack because it hooks various read/write functions in browser libraries to inject data. This technique is implemented as follows:

- Cyber criminals have to design specific rules for a bot agent to perform WI. A bot agent reads various rules from a static file and then uses hooking to apply those rules to modify incoming HTTP responses. Rules are tied to specific web pages, e.g., the login page of a bank.
- It is crucial that the rules are structured properly because inappropriate WI rules can seriously disrupt the web page layout and the dynamic execution of JavaScripts. Wild modification of the web stream will be obvious and hence ineffective. For successful WI, the injected content has to work inline without any display of errors or notifications to the users.
- Cyber criminals are required to define several parameters to write different WI rules. The WI rules are written explicitly for every GET and POST request with a dedicated URL. There are two specific parts of the WI rule. First, it is required to define the target URL (bank website, etc.) whose content is to be hooked and modified. Second, in every rule it is required to define the layout of the web pages, e.g. specify a portion of the webpage in which the content is to be injected in order to render the content appropriately in the browser.

Listing 1 shows a WI rule extracted from an infected machine. The rule injects additional input asking for a user's ATM PIN. It is an unusual request from a bank, but since the page is otherwise legitimate, trust compels a user to enter the information. This injection is placed before the password input box (specified by the data_before tag)—injecting inline as the web page enters the browser. The details of the parameters used to write a WI rule are discussed in [13]. As WI is a problem at the client side, banks currently have no robust protection against this attack. In addition, cyber criminals can inject sophisticated JavaScripts to perform online transactions automatically. For example, a bot injects malicious JavaScript during an active session with the server. The JavaScript interacts with the server and initiates a transfer from the user's account to an offshore institution. When the server sends a notification about a change in balance in the account, the incoming data (balance amount) is manipulated to reflect a different number. The user is tricked to believe that the account balance is intact. A bot can also generate unauthorized messages on behalf of the server.

3.3 Custom Plugins

Modern botnets implement a plug-in framework for executing a variety of attacks. The plug-in framework extends the capability of botnets by allowing the cyber criminals to write custom code that can be easily incorporated into running botnets. During our analysis of the SpyEye botnet [15], we came across interesting plug-ins that are used for data exfiltration. These are as follows:

- A browser certificate-grabber plug-in captures information about various certificates that are present in the browser storage repository and are used to verify the integrity of communicating parties.
- A credit card-grabber plug-in that is designed specifically to extract credit card information during an active session with a bank's server.
- A screenshot stealer and video grabber plug-ins that capture screenshots and videos of the browser when a user performs online banking. In addition, cyber criminals configure plug-ins in such a manner that a screenshot is captured based on the movements of the mouse cursor.
- Cyber criminals can also design plug-ins specific to a bank's website. For example, the SpyEye botnet has built-in information stealing plug-in that is designed specifically for BofA.

3.4 Mobile Platforms: SMS and HTTP as Data Carriers

Most of the mobile platforms are smart phones these days that provide the same functionality as standard computers, so data exfiltration models remains the same. The mobile bots and malicious applications can perform keylogging and monitoring of data that is transmitted through the device. Generally, mobile bots can communicate over HTTP and control the communication flow. The primary addition in the data exfiltration process apart from standard protocols is the use of SMS as a carrier for transmitting data. It means the mobile bots can steal sensitive information and use the SMS capability

of the device to send data to a backend domain managed by the cyber criminal. Mobile bots can perform piggybacking on legitimate applications and steal data by controlling specific events such as when the applications send data to a banking server. As discussed earlier, mobile bots can also circumvent the TFA process that uses SMS (mobile) as a second channel. Zitmo and Spitmo are the examples of mobile malware that support this fact.

3.5 Phished Web Pages

As discussed in the malware design section, automated bots are used for sending phishing emails with luring links. The phishing emails are constructed in a sophisticated manner that it becomes easy to force the users to visit the phished website. Once the user clicks the embedded link, the browser opens the phished website, which contains web forms that ask specific information from the users. Since the web pages look legitimate, users provide sensitive information such as credentials, credit card numbers, etc. This is an old-school trick, but works neatly in exfiltrating data from infected end user machines.

4. Phase 4: Underground Business

At some point, stolen data must be converted to cash, and for that we turn to the underground economy. In the underground market, there are three basic players: sellers, buyers and money mules. Sellers sell the data, buyers purchase the data, and money mules convert data to cash.

4.1 Underground Forums and IRC Channels as Business Platforms

Internet Relay Chat (IRC) [19] channels are used as the primary business platform in the underground economy because it allows cyber criminals to remain anonymous. Cyber criminals use Virtual Private Network (VPN) to initiate connections to IRC servers for registering communication channels. With the existence of invisible IRC, the communication channels are unreadable, encrypted and untraceable.

Once data is successfully stolen from infected machines, cyber criminals need to sell it. During our study, we analyzed underground forums that advertise various IRC channels used by cyber criminals to sell sensitive information. Automated MIRC scripts regularly advertise updates and availability of the stolen data. Sellers advertise a unique ICQ code with an IRC channel that a buyer can use to connect directly so the buyer is unable to identify the seller.

Data is sold in the form of dumps as shown in Figure 1 that are sent to the buyer once the seller receives payment.

Sellers require money in the form of Liberty-Reserve, Western Union, Money Gram, etc., which are e-currencies that can be converted into Euros, dollars or pounds. E-currency involves an intermediate third-party who does not reveal the identity of the buyer or the seller to maintain anonymity. The underground business is based on an implicit trust between the buyer and the seller that the seller will release purchased data upon receiving payment—there is no third party to turn to for resolving disputes.

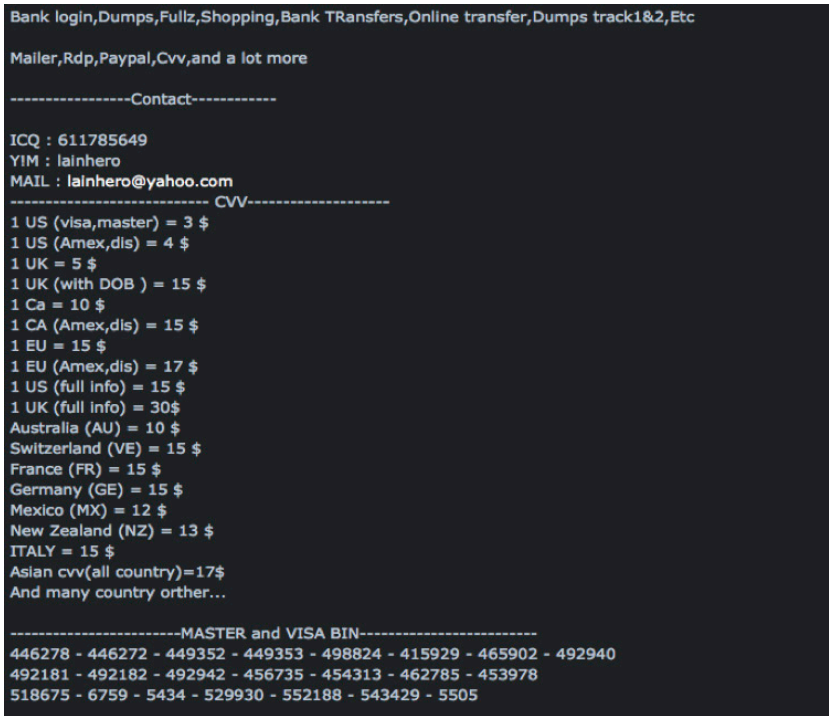

Figure 1 - Advertising Dumps of the Stolen Bank Data (Source: Underground Forum <http://madtrade.org/>)

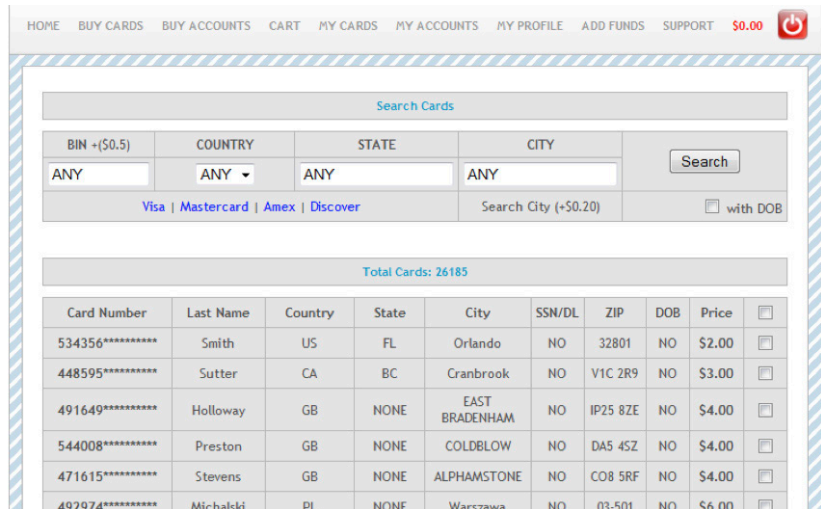

Figure 2 - Credit Card Shop in Action

4.2 Credit Card (Plastique) Shops

Credit card shops are e-shops that exist in the underground market to sell stolen credit card information. The credit card shops are similar to regular e-commerce websites. The buyer visits various underground websites to find information about the credit card sellers, and obtain the address of the credit card shops from various IRC channels and underground forums. The buyer then has to register with the shop. Once the registration is complete, the buyer can easily navigate the credit card shop and select credit cards for purchasing. Currently, the stolen credit card information is sold at very cheap rates ranging from $2 to $20. Figure 2 shows the layout of one current credit card shop we found during penetration testing of domains associated with malware.

Figure 3- Service Advertisements for Offshore Money Transfers (Source: Underground Forum http://madtrade.org/)

4.3 Money Mules

Money mules [18] are transfer agents hired to convert data into cash. For a fee, money mules use credentials (data) to extract money from a bank and then transfer the money to offshore accounts, often as e-currency. For bank transactions, money mules must usually have accounts in the banks that are targeted by cyber criminals for transferring funds—a requirement that puts mules at risk.

Most banks have strong security measures for transferring money outside a bank, but little security for transfers within a bank so it is common to transfer within a bank. We assume that credentials have been collected using techniques such as form grabbing as described above.

- Sometimes additional information is needed such as the user's account including registered email and password. It can be easily collected using techniques such as Form-grabbing or Web Injects as described above. If the bank uses TFA, the associated information such as an SMS number can be gathered in the same way. Hijacking sessions while in progress as outlined above can circumvent one-time passwords.
- With that information, the buyer needs to enlist a mule so the buyer needs the mule's name, account number, and routing number. Given restrictions on transfer amounts, multiple transactions or multiple mules may be needed.
- A buyer can use account credentials to transfer money to a mule or a mule's credentials can be provided back to the seller to build transactions into a victim's live session. The seller can use WI to inject the mule's credentials into web pages using JavaScript. The script causes a fraudulent transaction during a user's session to transfer money directly to the mule's account.
- Once money has been transferred to a mule's account, the buyer sends a confirmation to the money mule, e.g., a screenshot. Upon receiving the confirmation, the money mule moves the money outside the bank. The transfer may be to cash, to an overseas account, to merchandise, or to e-currency. Upon transferring the money, the mule will extract a fee for their services. The fee can vary significantly depending on the complexity of service provided, but we have observed fees ranging from 2% to 10%. Figure 3 shows an advertisement for this kind of service in the underground market. Money mules are prevalent in regions that currently lack strong cyber laws: Eastern Europe, Russia, Middle East, etc.
- An optional fourth actor may be present—a bank insider who can be thought of as a type of money mule. A bank employee can facilitate overseas transfers, especially large transfers. An overseas transfer needs another money mule at the other end to complete the transaction.

Underground markets facilitate the buying and selling of the stolen data without revealing the identity of the players.

5. Existing Countermeasures and Defensive Mechanisms

Banks are deploying several interesting techniques to combat online fraud. Several of them are discussed as follows:

- The majority of banks implement SSL that protects customers from network layer attacks by encrypting the channel between end points. While worthwhile, this practice is not suffice to combat browser-based data exfiltration attacks conducted by MitB agents. By working within the browser, the attack is done before SSL encrypts the data.
- Banks also deploy multi-factor authentication systems using multiple channels to authenticate clients. A popular one is TFA. Display tokens such as RSA Secure ID, Safenet's e-token and Vasco secure tokens use either time-based or sequence-based algorithms to generate unique tokens for authentication or digital transaction signing. The user possesses a small device that generates tokens at regular intervals. The token is used as a second factor in authentication. For example, HSBC bank uses RSA Secure ID, and BofA uses Safe Pass.
- In a variation on TFA, some banks use one-time passwords [20] for authentication. Banks store the information about users' computers including IP address, browser, geo IP location, etc. If the bank's server finds that the information has changed, it activates the OTP scheme. The bank will have on file either an email address or mobile number for receiving the OTP. Using this second channel, the OTP is sent to the user. JP Morgan Chase bank is an example of a bank that implements this procedure.
- Banks have also implemented site-key authentication to thwart phishing attacks. During account registration, the user selects an image with a key for additional verification. The legitimate account login page includes this site key which assures the user of the authenticity of the website. Typically, a complete site key consists of an image, selected text and challenge questions. Generally, the challenge questions are asked when the connected computer is not recognized. BofA and HDFC bank are examples of banks that incorporate this functionality. Note that this technique does not help prevent MitB attacks.
- Some banks recommend third party monitoring solutions such as Trusteer Rapport [17]. It is an active fraud prevention and account takeover detection solution, and users are advised to install it before using banking websites. Companies like Netqin [28] provide mobile anti-malware solutions to protect the integrity of mobile devices.
- Banks have also built a protection against keylogging attacks in the form of virtual keyboards using JavaScript. This technique prevents keylogging but fails to protect against form grabbing. A few banks are using client-side password encryption to defend against the reuse of stolen credentials. The State Bank of India (SBI) is following this practice.
- Apart from technical solutions, banks also perform forensic investigative analysis of money fraud problems reported by users. This includes analyzing the anomalies that persist in transactions. The anti-fraud teams collaborate with government agencies to unmask the players behind these frauds.

Banks are taking a variety of steps to fight against a variety of cyber crime, but none prevent current MitB attacks. TFA is an effective defense against the use of stolen credentials, but WI can allow criminals to collect information on the second channel. TFA raises the bar and WI provides a work-around, but it is a difficult work-around.

6. State of Cyber Laws

Nations with advanced economies such as the U.S. or the UK have begun to implement cyber laws. The biggest problem in eradicating cyber crime globally is the lack of centralized cyber laws. The proposed cyber laws are country specific and cannot be enforced across borders (except to a limited extent through existing treaties). Quite naturally, countries are most concerned with cyber crimes that impact their own institutions, so law enforcement agencies are more interested in investigating or prosecuting cyber criminals that exploit the integrity of their own country's critical infrastructure. Contributing to the problem is the international nature of cyber crime. Cyberspace has no borders so cyber criminals can work anywhere. Many countries have still not implemented strong cyber laws and that is a problem for managing cyber crime internationally. The laws that have been implemented vary considerably—the crimes are too new to have developed widespread standards. The U.S. is one of the leaders in making and implementing cyber laws [16] but those laws cannot be enforced globally. As an example, U.S. cyber law 18 USC 1030 deals with crimes that are conducted through compromised (unauthorized access) computers and further using them to execute identity fraud against financial institutions. A convicted person can get five to 10 years in prison. Clearly, more needs to be done and countries are working to build a robust approach against cyber crime. The efforts must be international, if we are to build a secure cyberspace.

Conclusion

In this paper, we presented attack methods for conducting online bank fraud. To carry out fraud, cyber criminals have created sophisticated methods of malware distribution, infection, and data exfiltration. One important trend is toward infecting users' systems rather than attacking banks' servers. The criminals coerce users to visit malicious domains where drive-by downloads use browser vulnerabilities to download malware. The malware hooks browser functions to allow form data (credentials) to be grabbed from banking sessions. On mobile devices, malicious applications are installed that perform piggybacking, hijacking communication channels of other legitimate applications and transmitting data using HTTP or SMS to remote servers. The sensitive information is sent to cyber criminals who convert data to cash using different channels. Some banks have implemented OTP and TFA—and these authentication systems work well against some attacks—but they fail to provide adequate protection against MitB and MitMo attacks. As a result, cyber bank fraud has become a critical problem on the Internet. To secure online banking, multilayer defenses including user education are needed.

ABOUT THE AUTHORS

Aditya K. Sood is a senior security researcher/consultant and PhD candidate at Michigan State University. His research interests include web security, malware analysis, mobile security, and penetration testing. Sood has an MS in cyber law and information security from the Indian Institute of Information Technology, India. He is regular speaker at industry wide security conference and have contributed to number of security magazines and journals.

30 Newport Parkway, Apt 2111
Jersey City, NJ -07310, USA
Phone: 517-755-9911
E-mail: soodadit@cse.msu.edu

Richard J. Enbody, Ph.D., is associate professor in the Department of Computer Science and Engineering at Michigan State University (USA) where he joined the faculty in 1987. Enbody has served as acting and associate chair of the department and as director of the computer engineering undergraduate program. His research interests include computer security; computer architecture; web-based distance education; and parallel processing.

Department of Computer Science and Engineering
Michigan State University
3115 Engineering Building
East Lansing, Michigan 48824
Phone: 517-353-3389
Fax: 517-432-1061
E-mail: enbody@cse.msu.edu

REFERENCES

1. V. Garg, C. Kanich and L Camp, Analysis of eCrime in Crowd-sourced Labor Markets: Mechanical Turk vs. Freelancer, 11th Workshop on the Economics of Information Security (WEIS), 2012
2. C. Herley and D. Florencio, Nobody Sells Gold for the Price of Silver: Dishonesty, Uncertainty and the Underground Economy, In Proceedings (online) of the Workshop on Economics of Information Security, June 2009.
3. R. Anderson, C. Barton, R. Bohme, R. Clayton, M. Eeten, M. Levi, T. Moore and S. Savage , Measuring the Cost of Cybercrime, 11th Workshop on the Economics of Information Security (WEIS), 2012
4. G. Ollman, Serial Variant Evasion Tactics. Damballa Whitepaper. <http://www.damballa.com/downloads/r_pubs/WP_SerialVariantEvasionTactics.pdf>
5. A. Sood and R. Enbody, A Browser Malware Taxonomy, Virus Bulletin Magazine, June 2011 <http://secniche.org/released/VB_BRW_MAL_TAX_AKS_RJE.pdf>
6. K. Curran and T. Dougan. Man in the Browser Attacks. International Journal of Ambient Computing and Intelligence. Vol (4) -1, 2012
7. N. Harbour, Win at Reversing - API Tracing and Sandboxing through Inline Hooking, In 17th Annual DEFCON Conference, 2009
8. J. Butler and P. Silberman, RAIDE - Rootkit Analysis Identification and Elimination, In BlackHat Security Conference, 2006.
9. C. Devaux and J. Lenoir, Browser Rootkits, Hack Luxembourg Conference, 2008 <http://archive.hack.lu/2008/rootkits-navigateurs.pdf>
10. C. Jackson, D. Boneh and J. Mitchell, Transaction Generators - Rootkits For Web, In Usenix HotSec Conference, 2007
11. M. Cova, C. Kruegel and G. Vigna. Detection and analysis of drive-by-download attacks and malicious JavaScript code. In Proceedings of the 19th international conference on World wide web, 2010
12. Malware-at-Stake Blog, (SpyEye & Zeus) Web Injects - Parameters, <http://secniche.blogspot.com/2011/07/spyeye-zeus-web-injects-parameters-and.html>
13. A. Sood, R. Enbody and R. Bansal, The Art of Stealing Banking Information - Form Grabbing on Fire, Virus Bulleting Magazine, November, 2001.
14. Chase - Malware and Virus, Do Not Fill Out Pop-Up Windows Like This <https://www.chase.com/index.jsp?pg_name=ccpmapp/privacy_security/fraud/page/virus_malware_examples>
15. A. Sood, R. Enbody and R. Bansal, Dissecting SpyEye – Understanding the design of third generation botnets, Elsevier Computer Networks Journal, Online Print, August 2012
16. A. Rees, Cybercrime Laws of the United States, October, 2006. <http://www.oas.org/juridico/spanish/us_cyb_laws.pdf>
17. Trusteer Rapport, User Guide, <http://www.trusteer.com/support/user-guide/3.5.1201/Rapport_UG_3_5_1201_4.pdf>
18. M. Aston, S. McCombie, B. Reardon, and P. Watters. A Preliminary Profiling of Internet Money Mules: An Australian Perspective. In Proceedings of the 2009 Symposia and Workshops on Ubiquitous, Auto nomic and Trusted Computing, 2009.
19. C. Mazzariello. IRC Traffic Analysis for Botnet Detection. In Proceedings of The Fourth International Conference on Information Assurance and Security, 2008
20. A. Rubin, Independent one-time passwords. In Proceedings of the 5th conference on USENIX UNIX Security Symposium, 1995
21. Dinei Florencio, Cormac Herley, "Is Everything We Know about Password Stealing Wrong?" IEEE Security & Privacy, vol. 10, no. 6, pp. 63-69, Nov.-Dec., 2012
22. Honeynet Project, Phishing Using Botnets, <http://www.honeynet.org/node/92>
23. Spamhaus, <http://www.spamhaus.org>
24. T. Morrison, Spam botnets: The fall of Grum and the rise of Festi, <http://www.spamhaus.org/news/article/685/spam-botnets-the-fall-of-grum-and-the-rise-of-festi>
25. C. Castillo, Spitmo vs. Zitmo: Banking Trojans Target Android, <http://blogs.mcafee.com/mcafee-labs/spitmo-vs-zitmo-banking-trojans-target-android>
26. Y. Zhou and X. Jiang, "Dissecting Android Malware: Characterization and Evolution, <http://www.csc.ncsu.edu/faculty/jiang/pubs/OAKLAND12.pdf>
27. W. Zhou, Y. Zhou, M. Grace, X. Jiang and S. Zou, "Fast, Scalable Detection of `Piggybacked' Mobile Applications, <http://www.csc.ncsu.edu/faculty/jiang/pubs/CODASPY13.pdf>
28. NetQin, NetQin Mobile Security, <http://www.netqin.com/en/antivirus>
29. SecNiche Security Blog, <http://secniche.blogspot.com/2012/08/digital-forensics-magazine-dismantling.html>
30. A. Sood, ToorCon 14 (2012): Malandroid - The Crux of Android Infections, <http://zeroknock.blogspot.com/2013/05/toorcon-14-2012-malandroid-crux-of.html>