The Untold Story of the Target Attack Step by Step

Aorato Labs | August 2014
Contents

Executive Summary........................................................................................................3
Mapping the Knowns and the Unknowns .................................................................4
Methodology ................................................................................................................5
Infographics: Target Attack Breakdown.................................................................6
Hackers Voyage from Network’s Boundaries to its Heart........................................7
  Step 1: Install Malware that Steals Credentials.......................................................8
  Step 2: Connect Using Stolen Credentials .............................................................8
  Step 3: Exploit a Web Application Vulnerability ....................................................8
  Step 4: Search Relevant Targets for Propagation ..................................................9
  Step 5: Steal Access Token from Domain Admins ...............................................10
  Step 6: Create a New Domain Admin Account Using the Stolen Token ................11
  Step 7: Propagate to Relevant Computers Using the New Admin Credentials ....11
    Bypassing Firewall and Other Network-based Security Solutions .....................12
      Running Remote Processes on Various Machines ..........................................12
  Step 8: Steal 70M PII. Do Not Find Credit Cards...............................................13
  Step 9: Install Malware. Steal 40M Credit Cards .................................................14
  Step 10: Send Stolen Data via Network Share ....................................................14
  Step 11: Send Stolen Data via FTP ..................................................................14
Target’s Attackers Tactics, Techniques and Procedures ..........................................15
Security Recommendations.......................................................................................17
  Recommendations to Industries Prone to Advanced Targeted Attacks ...............17
  Recommendations to Retailers Storing Credit Card Information ......................17
Appendix A: VISA’s Attack-Tools List .................................................................18
Appendix B: Dell Secureworks Attack-Tools List .................................................19

© 2014 by Aorato, inc. All rights reserved.
Executive Summary

In December 2013, in the midst of the busiest shopping season of the year, Target announced that it had been breached by attackers who had gotten away with 70M customers’ Personal Identifiable Information (PII). A few days later, Target admitted that 40M credit cards were stolen. The financial damages to Target currently stand at $148M, and according to analyst forecasts are estimated to reach $1B.

Although many details regarding the attack had surfaced and made it to the general audience, some aspects of the attack still remained in the dark. Specifically, how did the attackers reach into the heart of Target’s network, the POS (Point-of-Sale) system from their initial penetration point? Second, how were 70M users’ “Personally Identifiable Information” (PII) exposed?

We set to find the answers to these questions. By carefully analyzing the publicly available reports on the Target breach, we were able to build out the entire Target attack story.

While this story largely follows the general “APT kill chain” attack model which is relevant to nearly any industry targeted by advanced attackers, the Target attack introduces some additional nuances, specifically relevant to retail and other credit card processing targets. It suggests that a vertical focused cyber intelligence sharing system, such as R-CISC (Retail Cyber Intelligence Sharing Center) and R-ISAC (Retail Information Sharing and Analysis Center) can be highly beneficial.

In this report, we breakdown the Target attack to 11 detailed steps, beginning with the initial credential theft of Target’s HVAC contractor to the theft of PII and credit cards. Particular attention is given to those steps, unknown until now, such as how the attackers were able to propagate within the network. Throughout this report we highlight pertinent insights into the Tactics, Techniques and Procedures (TTPs) of the attackers. Finally, we provide recommendations on the needed security measures for mitigating similar advanced targeted attacks.

Key Findings:

**TTPs**: Attackers Tactics, Techniques and Procedures (TTPs) included general IT tools, protocols and procedures. Seldom did they use hacker-specific tools and malware.

**PtH**: Attackers used “Pass-the-Hash” (PtH) techniques to propagate through Target’s network.

**PII**: Attackers had gained access to 70M Personal Identifiable Information (PII) by exploiting a SQL server database.

**PCI**: PCI compliance actually improved the security posture of Target. Target’s compliance with PCI not only minimized the scope of the breach, but also forced the attackers to slow down as they re-assessed and changed their course of attack.

**AD**: Active Directory (AD) related activity was paramount to the attackers’ success.

---

3. [http://www.rila.org/rcisc/home/Pages/default.aspx](http://www.rila.org/rcisc/home/Pages/default.aspx)
Mapping the Knowns and the Unknowns

Before delving into the missing pieces of the attack puzzle, let’s map out the already known facts about the Target breach as explicitly revealed by publicly available reports:

1. The initial penetration point of the attackers was through stolen HVAC vendor’s credentials\(^5\).
2. The attackers used the vendor’s stolen credentials to gain access to a Target hosted web services for vendors.
3. Attackers deployed the “Kaptoxa” (pronounced “Kar-toe-sha”) malware on many Target’s POS machines which was used to steal credit card information.
4. Stolen credit cards were periodically sent to a central repository within Target’s network using standard Windows protocols (specifically, the SMB\(^6\) protocol).
5. The stolen data was exfiltrated from the central repository to the attackers’ controlled server via FTP.

These five steps have been extensively documented and technically analyzed. However, a knowledge gap exists when it comes to the following:

- How were the attackers able to move from their initial point of penetration, located on the boundary of Target’s network, to deploying malware in the heart of the network?
- Where is the explanation for the stealing of 70M of Target’s customers PII? This chain of events only explains the exfiltration of 40M credit cards.

In order to fill the information gaps, we meticulously read the publically available reports and advisories to discover on top of the five aforementioned steps, six additional previously undocumented steps. These steps provide the necessary explanation from the initial steps of penetration to the installation of malware on the POS machines and the theft of 70M PII.

---


Methodology

As stated above, in order to fill the information gaps, we meticulously read the threat advisories issued by the credit cards companies and security firms. Our primary sources were:

- VISA’s “Retail Merchants Targeted by Memory-Parsing Malware” report, issued on February 2014
- Dell Secureworks’ “Inside a Targeted Point-of-Sale Data Breach” report, issued on January 2014
- iSight Partners “KAPTOXA Point-of-Sale Compromise” report, issued on January 2014
- KerbsenSecurity blog post series on Target Data Breach
- “A ‘Kill Chain’ Analysis of the 2013 Target Data Breach” report, for the Senate Committee on Commerce, Science, and Transportation, issued on March 2014

In particular, we paid special attention to the list of the tools used by the attackers disclosed in the aforementioned advisories. The attack-tools list appears in the appendices.

While this work can be dismissed as merely educated guesswork and therefore may include some inevitable errors, it seems that our deductions and inference are solid and backed up with several different findings. In all cases we had included the evidence that led us to the conclusions so readers can follow our line of thought and judge for themselves.

We hope that this report sparks the discussion about the missing links in the Target’s breach story. More so, we hope this discussion expands outside the realms of the Target attack and to other advanced attacks - leading to the disclosure of more attack data and facts. We strongly believe that disclosing data about the attackers TTPs will benefit the security community in building stronger and more efficient defenses against such threats.

11 http://docs.ismgcorp.com/files/external/Target_Kill_Chain_Analysis_FINAL.pdf
Target Network

HVAC vendor computer

1. Install malware that steals credentials

2. Connect using stolen credentials

Target’s Web app for vendors

3. Exploit Web app vulnerability

4. Search relevant targets for propagation

Web app server

5. Steal access token from domain admin

6. Create new admin account using stolen token

Active Directory

7. Propagate to relevant computers using new admin credentials

8. Steal 70M PII. Do not find CCs

9. Install malware. Steal 40M CCs

DB

10. Send stolen data via network share

POS

FTP-enabled PC

11. Send stolen data via FTP

Attacker-controlled FTP server

Active Directory Authentication, Authorization, and other related Directory Services protocols.
Hackers Voyage from Network’s Boundaries to its Heart

In this section we discuss in depth how the attackers were able to propagate from a web interface - where its server resided on the boundaries of Target’s network - to the POS machines, the very heart of Target’s network.

The figure above illustrates the attack campaign, where the attackers performed the following steps:

1. **Install malware that steals credentials** from the computer of Target’s HVAC vendor.
2. **Connect using stolen credentials**. The stolen credentials of the HVAC vendor enable access to Target’s application dedicated to vendors.
3. **Exploit a web application vulnerability** on Target’s Web interface of the application dedicated to vendors. The exploit enables the attackers to execute code on the Web application’s server.
4. **Search relevant targets for propagation** by querying Active Directory from the Web application’s server. Queries are performed over the LDAP protocol.
5. **Steal access token from Domain Admin**. The attackers steal the token of the previously connected Domain Admin from the memory of the Web application’s server.
6. **Create a new Domain Admin account using the stolen token**. This new account is created in Active Directory.
7. **Propagate to relevant computers using the new Domain Admin credentials**. The relevant computers were identified in step (3), and the new Admin account was created in step (6).
8. **Steal 70M PII. Do not find credit cards**. The data is extracted from a PCI-compliant database, using the SQL protocol from a previously propagated computer. Since the database is PCI-compliant, no credit cards are stored on it.
9. **Install malware. Steal 40M Credit Cards**. The data is extracted by the Kaptoxa malware from the memory of the POS system.
10. **Send stolen data via network share**. Malware sends the extracted credit card and PII data, obtained in steps (8) and (9), to an FTP-enabled machine within Target’s internal network.
11. **Send stolen data via FTP** to attackers-controlled FTP server.

In the following subsections we deep dive into the details of each of these steps. We give particular attention to the six steps (steps 3-8) as these are additional, previously undocumented steps which provide the complete story of the Target attack path.
Step 1: Install Malware that Steals Credentials

According to publicly available sources\(^\text{12}\), the attackers infected the Target’s HVAC contractor with the Citadel malware through the use of a phishing email. The Citadel malware, is a “run-of-the-mill”, general purpose malware and had been documented to infect millions of computers in the past\(^\text{13}\). Citadel is known to be able to harvest web applications credentials stored within the infected machine browser.

Step 2: Connect Using Stolen Credentials

The attackers used the vendor’s stolen credentials to gain access to Target-hosted web services dedicated to vendors\(^\text{12}\). According to the contractor’s (Fazio Mechanical) official announcement\(^\text{14}\), the only access the HVAC vendor had was to some of Target’s vendor administrative systems: “Fazio Mechanical does not perform remote monitoring or control of heating, cooling or refrigeration systems for Target. Our data connection with Target was exclusively for electronic billing, contract submission and project management”.

As such, the credentials were used to access any one of these following services\(^\text{12}\):

1. The “Ariba” Web application: a billing system.
2. The “Partners Online” Web application: Target project management and contract submissions portal.
3. The Target’s Property Development Zone Web application.

Step 3: Exploit a Web Application Vulnerability

We know that the attackers stole the HVAC’s vendor’s credentials to Target’s internal web application, hosted on Target’s internal network. However, we also know that this system has a very specific functionality that does not allow arbitrary command execution which the attackers need in order to compromise the machine. How then were the attackers able to bypass this restriction? We sought to find out.

As a first step, the attackers had to find a vulnerability within the web application. Despite no public information regarding this vulnerability, we were able to find a clue hinting to its source. Looking at the supplied attack-tools list, we found a file named “xmlrpc.php”. This file immediately stands out, as all other files in the list are Windows executables while PHP files are used for running scripts within web applications.

This file suggests that the attackers were able to upload a PHP file by leveraging a vulnerability within the web application. The reason is that it is likely the web application had an upload functionality meant to upload legitimate documents (say, invoices). But as often happens in web applications, no security checks were performed in order to ensure that executable files are not uploaded.

In order for the attackers’ to disguise their malicious script as a popular PHP component, the attackers named the uploaded file as “xmlrpc.php”\(^\text{15}\). The code within this bogus xmlrpc.php script was probably a “web shell”\(^\text{16}\), a web based backdoor that allowed the attackers to upload files and execute arbitrary OS (Operating System) commands.

\(^{12}\) http://krebsonsecurity.com/2014/02/email-attack-on-vendor-set-up-breath-at-target/

\(^{13}\) http://www.eweek.com/security/microsoft-liberates-more-than-1.2-million-pcs-from-citadel-botnet/


TTP 1: The methodology of “hiding in plain sight” and disguising malicious components as legitimate, repeats itself throughout the attackers’ campaign.

TTP 2: Attackers maintain persistence for their programs within breached machines by running their programs as server scripts or services.

The use of an uploaded file to subvert a web application has been documented to be a popular penetration method among the attackers\(^{17}\). In fact, the method of using an Internet-facing web application as a springboard for penetrating an organization is not confined to Target’s breach. In the past, this method had played a pivotal role in other attacks\(^{18}\) such as that against the security vendor, Bit9\(^{19}\).

TTP 3: Attackers use uploaded files to infiltrate internal systems

Step 4: Search Relevant Targets for Propagation

Due to the Web vulnerability exploit, as detailed in the previous step, the attackers were able to run arbitrary OS (Operating System) commands. Consequently, the attackers were able to start their internal reconnaissance campaign in order to gather intelligence on Target’s internal network and find the relevant targets - servers that hold credit cards and credit card’s holders’ information.

To glean this type of information, the attackers’ targeted Active Directory. Active Directory, as its name suggests, contains the data on all members of the Domain\(^{20}\): users, computers and services.

TTP 4: Active Directory is pivotal to attackers to gather intelligence on the target’s infrastructure, users, computers and services

To query Active Directory there is no need for a specialized tool or special privileges. On the contrary - this functionality is supplied with internal Windows tools using the standard LDAP protocol. Furthermore, any Domain user can query the Active Directory. It makes sense then to imagine that the attackers, interested in identifying databases holding credit cards, simply retrieved all services that their Service Principal Name (SPN) contained the string “MSSQLSvc”\(^{21}\). Further inspection of the retrieved service names allowed the attackers to infer the purpose of each service by looking at the name of the server (e.g. a hypothetical SPN string of “MSSQLSvc/billingServer”). This scenario makes even more sense considering that the 70M PIIs from Target’s network were probably stored within a database. The reason is that the attack tools list contained a few SQL related tools, such as the osql.exe, isql.exe and the bcp.exe tools. These tools are analyzed more thoroughly later in the document.

Similarly, the attackers had probably applied such a process to find POS-related machines.

Once the attackers had found the names of their targets, whether SQL servers or POS machines, they were able to obtain the respective IP addresses by querying the DNS server, which is often co-located on the Active Directory server.

---

17 https://www.brighttalk.com/webcast/7451/110653 (slide 12)
19 http://www.computerworld.com/s/article/9237142/Hacking_victim_Bit9_blames_SQL_injection_flaw
20 http://en.wikipedia.org/wiki/Windows_domain
**Step 5: Steal Access Token from Domain Admins**

Given that the attackers had identified their relevant targets, they needed the proper privileges to access them. The holy grail of access within a Windows network is having Domain Admin privileges as these are the highest available permissions which empower a user to access every computer. Naturally, the attackers sought to get hold of Domain Admin privileges.

**TTP 5: Attackers require Domain Admin privileges to smoothly propagate within the network.**

Luckily for the attackers, they did not need to look far. Using a well-known attack technique, called Pass-the-Hash, the attackers were able to impersonate a valid user. How did we infer this?

A former member of Target’s security team admitted to cyber-investigative journalist, Brian Krebs, that: “Most, if not almost all, internal applications at Target used Active Directory (AD) credentials... the internal administrators would use their AD login to access the system from inside.”

To understand this statement, it is important to recognize that when a user interactively logs into a computer, Windows generates a token, called an NT hash which resides in the computer’s memory. This token replaces the password and allows the Single-Sign-On (SSO) experience, in which the users are only asked for their password once. These tokens may remain in memory until the server is booted.

Since servers are rarely booted, most chances are that the NT hashes of Target’s Domain Admin logging into the system servers were still in the memory of the system when the attackers broke in, allowing the attackers to obtain them from the machine memory and gain Domain Admin privileges (i.e. what’s called Pass-the-Hash).

Visa’s report further strengthens this assumption as it recommends to:

- Limit administrative privileges on users and applications.“
- Do not use NTLM or LM hash for password hashing as the algorithm is known to be compromised and susceptible to a Pass-the-Hash type of attack.”

The attackers obtained these NT hashes from the Web server’s memory using any of the following tools that are listed in the attack-tools list:

- **WCE**: Windows Credential Editor (WCE) is a known penetration test tool that can “Steal’ NTLM credentials from memory”22. Its use is indicated by the existence of the GETLSASRVADDR.exe23 file: “GETLSASRVADDR.exe is a tool (included with WCE) that can be used to obtain automatically the needed addresses for WCE to be able to read logon sessions and NTLM credentials from memory”.
  Another possible indication to the attackers’ use of WCE is the 2WCE.exe file, due to its similar name.
- **QuarksPwDump**: A tool that, according to its documentation, “Extracts domain accounts NT/ LM hashes + history”24.
- **Elcomsoft Proactive Password Auditor**: The tool’s documentation says: “If you have administrator rights on the machine you run PPA on, you can dump password hashes from its memory.”25

Using WCE, the attackers were able to perform the aforementioned Pass-the-Hash attack and reuse

---

22  www.ampliasecurity.com/research/wcefaq.html#whatiswce
23  http://www.ampliasecurity.com/research/wcefaq.html#whatislsasrvaddr
24  https://code.google.com/p/quarkspwdump
the token to impersonate the original owner of the NT hash, the Active Directory administrator.

Interested in learning more about Pass-the-Hash? Read [here](#).

**Step 6: Create a New Domain Admin Account Using the Stolen Token**

Once the attackers had gained the privileges of a Domain Admin through the use of NT hash and Pass-the-Hash attack, they proceeded to create a new Domain Admin account. The new account offered the attackers the following advantages over the stolen hash:

- **Persistency**: The NT hash becomes invalid when the victims change their password.
- **Password**: some services, such as the Remote Desktop (RDP), do not support the SSO paradigm and require the explicit use of the password. With the added user, the attackers control the password and can use these services as we will show below.

To add a Domain Admin, the attackers were not required to install any specialized tool, as this functionality is already supplied with following embedded Windows commands:

- `net user`: to add user to the Domain
- `net group`: to add the user to the Domain admins group

**TTP 6**: Attackers leverage existing Windows functionality to perform sensitive administrative tasks on the Domain.

Note, that the new Domain Admin account is created on Active Directory, making Active Directory very much aware to this activity.

As in previous steps, also here the attackers chose to “hide in plain sight”. The attackers did this by creating the username “best1_user” which mimics the username of BMC’s Bladelogic Server Automation product, a legitimate IT application. However, according to BMC statement, this username’s password is not the one set by Bladelogic.

At a later attack stage, the attackers’ malware used this “best1_user” account to send the credit cards data from the POS machine to a central repository within Target’s network.

**TTP 7**: Attackers create bogus accounts to maintain their persistency within the network.

Based on this attack step, Visa’s report recommends to “Periodically review systems (local and domain controllers) for unknown and dormant users.”

**Step 7: Propagate to Relevant Computers Using the New Admin Credentials**

With both the intelligence on the relevant computers (obtained in step 4) and the means to access them with Domain Admin credentials (obtained in step 6), the attackers were able to propagate towards the relevant targets.

At this stage, two obstacles stood in front of the attackers:
1. Bypassing firewall and other network-based security solutions that limit the attackers’ direct access to their relevant targets

2. Running remote processes on various machines in the chain towards their relevant targets

Let’s see how the attackers overcame each:

**Bypassing Firewall and Other Network-based Security Solutions**

The attackers used the “Angry IP Scanner” in order to identify network protection obstacles. The “Angry IP Scanner”, appearing too within the attack-tools list, detects which computers are network accessible from the current computer.

![Figure 2 Angry IP scanner](image)

To overcome such obstacles, the attackers had propagated through a series of servers (aka as a “tunnel”) to bypass the security measures. Evidence of this bypassing can be found in the existence of the port forwarding IT tool in the attack-tools list, aimed to defeat such firewall rules.

*Figure 3 The Port Forwarding utility used by the attackers to bypass firewall rules*

**TTP 8: Attacker bypass network-based security solutions, such as firewalls, by using a network tunnel**

**Running Remote Processes on Various Machines**

To execute processes on the remotely targeted servers, the attackers used the credentials in conjunction with both the Microsoft’s PsExec utility and a Remote Desktop Protocol (RDP) tool:

- The PsExec utility, present in the attack-tools list, is[^31]: “a light-weight telnet-replacement that lets you execute processes on other systems, complete with full interactivity for console applications, without having to manually install client software. PsExec's most powerful uses include launching interactive command-prompts on remote systems and remote-enabling tools”.

- Windows internal Remote Desktop (RDP) client provides full control over the remote machine with a Graphical User Interface (GUI). By using it, the attackers can also leverage on the GUI to assess whether the target is indeed valuable. As a result Visa’s recommendation is to “Deny Remote Desktop Protocol (RDP) logons whenever possible.”


Note, that for both PsExec and RDP, the authentication and the authorization of the user on the remote machine is done against the Active Directory, making Active Directory very much aware to this activity.

**TTP 9**: Attackers use PsExec and RDP to remotely execute processes on internal network’s machines.

The attackers did not stop at gaining a “one time” access to those systems which are dependent on credentials. What the attackers strived for was to maintain a more persistent access. The way they gained persistency was by using the “OrchestratorRunProgramService.exe” which is part of Microsoft’s Orchestrator management solution and appears in the attack-tools list. As its name suggests the purpose of this service is to enable the (remote) execution of other programs. By installing the service, the attackers once again achieved both a seemingly-legitimate foothold (i.e. “hiding in plain sight”), as well as persistency (as the service is automatically restarted on boot) which allowed them to remotely execute arbitrary code on the compromised server, such as running the “Kaptoxa” malware.

**Step 8: Steal 70M PII. Do Not Find Credit Cards.**

One of the mysteries we sought to find out was the methods used by the attackers to steal the 70M PII’s from Target’s network. It’s time to figure it out.

First, let’s re-cap what the attackers had up until now. They had the intelligence on the relevant databases obtained in the reconnaissance stage (step 4). They obtained new Domain Admin credentials (step 6). And, using the Domain Admin creds, they were able to propagate to the specific computer that enabled them to query a sensitive database (step 7).

The attack-tools list provides some clues to the methods used to obtain the data from the database server, as the list contains the following SQL related tools:

1. `osql.exe`: Microsoft’s SQL query tool
2. `isql.exe`: Microsoft’s SQL query tool
3. `bcp.exe`: Microsoft’s SQL bulk SQL copy tool

It makes sense that the attackers used the query tools (`osql` and `isql`) to retrieve a few entries in order to assess the value of the database. Once they had determined that the data was relevant to them, they proceeded to use the `bcp.exe` utility to retrieve all of the database contents.

The interesting part? We can assume that the attackers needed to change their mode of operation due to Target’s PCI compliancy requirements. The reason is that normally all information resides in the database so it makes sense that the attackers’ first goal was to attack it. However, once the attackers discovered that the Target database was under PCI compliancy, the attackers had to switch to their contingency plan and directly attack the POS systems. To recall, section PCI-DSS section 3.2 states that: “Do not store sensitive authentication data after authorization (even if encrypted). If sensitive authentication data is received, render all data unrecoverable upon completion of the authorization process”

Target Attack, Step by Step

Therefore, it seems that although being PCI complaint did not stop Target from being breached, it did:

1. Protect the credit cards details of 30M of Target customers, as the database contained 70M records compared to only 40M records extracted from POS systems. This is more than 40% reduction of the incident’s repercussions.

2. Buy some precious time for the defensive side to regroup and defeat the attackers before a single credit card had been stolen. Unfortunately, this time was not properly used as Target remained unaware to the attack.

Step 9: Install Malware. Steal 40M Credit Cards

Once the attackers found out that they were not able to extract the credit card details from the database they switched to their contingency plan and installed the Kaptoxa malware on all of the POS machines, using the same propagation methods that were discussed in Step 7.

The malware scanned the memory of the POS machine and when identifying a credit card, it saved it to a local file.

Also in this step the attackers used the Windows Services functionality as a method to gain persistency. To “hide in plain sight” they named the new Windows Service as “POSWDS”, which seems to be legitimate at a first glance.

Step 10: Send Stolen Data via Network Share

To exfiltrate the credit card data obtained by the malware, the malware had created a remote file share on a remote, FTP-enabled machine by using Windows internal “net use” command and the Domain Admin credentials.

The malware then periodically copied its local file, containing credit card details to the remote share.

TTP 10: Attackers use standard IT procedures and protocols (such as utilizing a network share), where possible, to perform their attacks.

Note that both the authorizing of a remote share creation and copying files to it is done against Active Directory, making Active Directory very much aware of this activity.

Step 11: Send Stolen Data via FTP

Once the data, either credit cards or the content of the database, arrived to the FTP-enabled machine, a script on the machine (installed in the very same way as the malware, through the use of the techniques described in step 7) sent the file to attackers’ controlled FTP account, using Windows internal FTP client.
Target’s Attackers Tactics, Techniques and Procedures

Generally speaking, the Target attackers largely followed the general APT “kill chain” attack model. However, the Target attack presents unique nuances to the model. These nuances stem from the fact that operations aiming to steal credit cards are inherently different from classic APT operations aimed at intelligence gathering and infrastructure sabotage.

The main difference is that credit card-oriented attacks are bound to be revealed in a relatively short time as the monetization path of the attackers must include massive usage of the stolen credit cards that will get detected by the credit cards vendor’s fraud departments.

Consequently, attackers in credit card-oriented attacks do not invest in the infrastructure and automation aspects of the attack. Specifically:

- All the exploratory steps of the campaign are highly manual, using command line tools and UI applications. The only automation seen in the attack process was the ongoing routine part of the Kaptoxa malware, used to continuously steal credit cards data from POS and send it over to the attackers’ servers. In fact, had the attackers been able to extract the credit cards details from the first database, the breach would not have had any automated aspect at all!

- Judging from the available information, the attackers did not create the Command and Control (often named C&C/C2/CnC) channel infrastructure often associated with APT attacks. Rather, the attackers manually controlled the operation from within the network.

- The attackers mainly used “normal” IT tools. Malware was used only when no relevant IT tool existed: e.g. for scraping the memory of a POS process

- Rather than trying to keep invisible, e.g. by using root kits, attackers “hide in plain sight”. Hiding in plain sight had provided “good enough” camouflage for this limited-in-time operation. Notable examples of this technique:
  - Using some general purpose, legitimate tools, such as Microsoft Orchestrator for nefarious purposes.
  - Disguising the attackers by adding bogus accounts considered to be legitimate accounts of IT systems
  - Masking malware files by naming them as legitimate files
As mentioned, the Target attackers also adopted general attack practices typically used in various APT campaigns. Although these following insights were gleaned from the Target attack, these TTPs can not only help the retail industry, but are also relevant to any industry facing the threat of advanced targeted attacks such as finance, hospitality, hi-tech, manufacturing, pharmaceutical, mining, etc.:

- Attackers’ main method of penetration and propagation into the victim’s network is by using stolen credentials - and not by exploiting vulnerabilities.
  - Initial compromise with stolen credentials to a remotely accessible system within the internal network
  - “Pass-the-Hash” attack to obtain Domain Admin privileges and create a new Domain Admin account
  - Using the Domain Admin credentials
- Attackers use a staged approach to propagation. At first, they obtain a foothold within the new target through a manual connection (e.g RDP) to assess its value. If found valuable, they upgrade their grip to a persistent one, by:
  - Adding a service to the target system
  - Adding a backdoor (web server’s xmlrpc.php)
Security Recommendations

Recommendations to Industries Prone to Advanced Targeted Attacks

We recommend that potential targets of advanced targeted attacks should follow the best practices listed here to protect themselves against such a cyber-attack:

- Harden access controls
  - Monitor and profile access patterns to systems to identify abnormal and rogue access patterns.
  - Where possible, use Multi Factor Authentication (MFA) to sensitive systems to reduce risks associated with credentials stealing. Note, that MFA does not necessarily eliminate all risks associated with credentials stealing.
  - Segregate networks, limit allowed protocols usage and limit users’ excessive privileges. Note that network segregation will slow down attackers but will not eliminate the threat altogether.
- Monitor users’ lists for the addition of new users, especially privileged ones.
- Monitor for signs of reconnaissance and information gathering. Pay special attention to excessive and abnormal LDAP queries.
- For sensitive, single purpose servers, consider the whitelisting of allowed programs.
- Don’t rely on Anti-Malware solutions as a primary mitigation measure since attackers mostly leverage legitimate IT tools.
- Place security and monitoring controls around Active Directory as it is involved in nearly all stages of the attack.
- Participate in Information Sharing and Analysis Center (ISAC) and Cyber Intelligence Sharing Center (CISC) groups to gain valuable intelligence on attackers’ Tactics, Techniques and Procedures (TTPs)

Recommendations to Retailers Storing Credit Card Information

Additional and specific recommendation for retailers, restaurants, hotels and other enterprises which need to protect themselves from the mass theft of credit cards should follow these set of best practices:

- Invest in PCI-compliance.
- Consider the whitelisting of allowed programs for POS systems.
- Don’t rely on outbound communication monitoring for C&C channels as a primary mitigation measure since attackers do not use traditional command and control (C&C) channels.
- Participate in Retail Information Sharing and Analysis Center (R-ISAC) and Retail Cyber Intelligence Sharing Center (R-CISC) groups to gain valuable intelligence on retail attackers’ Tactics, Techniques and Procedures (TTPs)

Appendix A: VISA’s Attack-Tools List


<table>
<thead>
<tr>
<th>Filename</th>
<th>MD5 Hash Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>svchosts.exe</td>
<td>ce0296e2d77ec3bb112e270fc260f274</td>
</tr>
<tr>
<td>svchosts.exe</td>
<td>ce0296e2d77ec3bb112e270fc260f274</td>
</tr>
<tr>
<td>svchosts.exe</td>
<td>f7c20a277929c4cb70999affb03388e</td>
</tr>
<tr>
<td>2wce.exe</td>
<td>93405c57e915680f0182650fb75c47ee</td>
</tr>
<tr>
<td>DUMPSEC.exe</td>
<td>65dd8d2d9604d43a0e0b105024f9264</td>
</tr>
<tr>
<td>ftpremt.exe</td>
<td>abb234773b0ad268f9a554c7ee597489</td>
</tr>
<tr>
<td>ftpext.exe</td>
<td>4352e635046aa624d5f9084d5619e82</td>
</tr>
<tr>
<td>getlsasrvadd.exe</td>
<td>0b3384d61ea345f16c4a34b33e9276bc</td>
</tr>
<tr>
<td>ips.exe</td>
<td>6c1bc0b12997689c84c12cc70996a75</td>
</tr>
<tr>
<td>isatax64.exe</td>
<td>453810a770573d0f0e7014978cc404</td>
</tr>
<tr>
<td>local.exe</td>
<td>08644155f5c8f40cc23942c5c5068f</td>
</tr>
<tr>
<td>lstsrvall.exe</td>
<td>623e4626d269324da62c055228eae61f</td>
</tr>
<tr>
<td>mmon.exe</td>
<td>290c26433a0d9d14f1252e46b120643</td>
</tr>
<tr>
<td>mtmp.exe</td>
<td>db0450080be21ed08efc897eb3bd9e</td>
</tr>
<tr>
<td>netc.exe</td>
<td>322e136cb50db030e0d6eb2071da1ba7</td>
</tr>
<tr>
<td>netc.exe</td>
<td>322e136cb50db030e0d6eb2071da1ba7</td>
</tr>
<tr>
<td>notcp.exe</td>
<td>a35e944762f82aa5656da453dcba20d1</td>
</tr>
<tr>
<td>osql.exe</td>
<td>4b9b36800db395d8a95f331c46089e947</td>
</tr>
<tr>
<td>osql.rll</td>
<td>df5dbbcac6e6d12329f1b8a5c4ce0e9</td>
</tr>
<tr>
<td>pmap.exe</td>
<td>814b88ca4ef695f63f3af11912a1c807</td>
</tr>
<tr>
<td>portfwd.exe</td>
<td>d975fc67da11c9eb56254d5e6ebe0a</td>
</tr>
<tr>
<td>psexec.exe</td>
<td>a35e944762f82aa5656da453dcba20d1</td>
</tr>
<tr>
<td>qrark.zip</td>
<td>2cd8d8d3f1a821eef4564905367281</td>
</tr>
<tr>
<td>svchosts.exe</td>
<td>2cd8d8d3f1a821eef4564905367281</td>
</tr>
<tr>
<td>xmlrpc.php</td>
<td>c583bdcecc1c6651c82a2a95736799d</td>
</tr>
<tr>
<td>query.exe</td>
<td>a109c617e0cc92c79e9db97c8964cb4</td>
</tr>
<tr>
<td>release.exe</td>
<td>f45f8d74f769100e8502581f84d1a6e</td>
</tr>
<tr>
<td>svchosts.exe</td>
<td>1d2f0491678fbc6858ff2a5d61d3003</td>
</tr>
<tr>
<td>wmislog.exe</td>
<td>e2db09553f23a8abc5633f6bf1a0b49</td>
</tr>
<tr>
<td>svchosts.exe</td>
<td>c0c9c0f5a9c7a3a5043ad9a0afa5fd</td>
</tr>
<tr>
<td>bcp.exe</td>
<td>3f007d6612dc9910a55423e8686dac</td>
</tr>
<tr>
<td>osql.exe</td>
<td>021379a37f6fcb66db59ab73f7b13e</td>
</tr>
<tr>
<td>psexec.exe</td>
<td>a35e944762f82aa5656da453dcba20d1</td>
</tr>
<tr>
<td>bladelogic.exe</td>
<td>433a2750429d805907a484ff666163</td>
</tr>
<tr>
<td>System32.exe</td>
<td>b9cf8e70681755c1711c38944695eeaa</td>
</tr>
<tr>
<td>Svccs.exe</td>
<td>25f7b169b43c4d5db472af0e09b035</td>
</tr>
<tr>
<td>oposvc.exe</td>
<td>dd90c44af5da730b8c979667ae8fd3</td>
</tr>
<tr>
<td>svchosts.exe</td>
<td>0561344c4e446007f2d7a94679508ed</td>
</tr>
</tbody>
</table>
Appendix B: Dell Secureworks Attack-Tools List


<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryExpress.exe</td>
<td>Portable SQL client for Microsoft SQL (MSSQL) Server and Oracle databases</td>
</tr>
<tr>
<td>psexec.exe</td>
<td>Microsoft Sysinternals PsExec tool for running processes on remote systems</td>
</tr>
<tr>
<td>ppa_setup_en.msi</td>
<td>Elcomsoft Proactive Password Auditor password cracking tool</td>
</tr>
<tr>
<td>portforward.exe</td>
<td>Network port forwarding tool</td>
</tr>
<tr>
<td>osql.dll</td>
<td>MSSQL query tool resource DLL</td>
</tr>
<tr>
<td>osql.exe</td>
<td>MSSQL query tool</td>
</tr>
<tr>
<td>Isql.exe</td>
<td>MSSQL query tool</td>
</tr>
<tr>
<td>OrchestratorRunProgramService.exe</td>
<td>Microsoft System Center 2012 SP1 Orchestrator</td>
</tr>
<tr>
<td>netcat.exe</td>
<td>Netcat network utility for reading and writing data across the network</td>
</tr>
<tr>
<td>ipscan.exe</td>
<td>Angry IP network scanner</td>
</tr>
<tr>
<td>dumpsec.exe</td>
<td>Somarsoft DumpSec. Dumps Access Control List (ACL) information for files, registry, and network shares</td>
</tr>
<tr>
<td>bcp.exe</td>
<td>MSSQL bulk SQL copy tool</td>
</tr>
</tbody>
</table>
Aorato protects organizations from advanced attacks. Recognizing Active Directory’s pivotal role in the network, Aorato’s flagship product, DAF™, automatically learns the behaviors of all entities engaging directly, or indirectly, with Active Directory. By profiling the entities, DAF™ builds an interaction graph between all entities in order to detect in real-time suspicious entity behavior. Aorato is backed by strategic investors, including Eric Schmidt (Innovation Endeavors), Accel Partners, and the founders of Imperva and Trusteer.

To learn more about Aorato, please visit: http://www.aorato.com

© 2014 by Aorato, inc. All rights reserved.