Target "Hacker Tools" Provide Breach Insight

19 January 2014

Analyst: Gary Warner, CTO, Malcovery Security

On January 16, 2014, iSight Partners released a report about the KAPTOXA malware from the Target breach, which was posted publicly on the Wall Street Journal website at the URL:

http://online.wsj.com/public/resources/documents/target.pdf

The document was removed from that location for reasons unknown to this analyst.

Brian Krebs, of KrebsonSecurity.com, also provided a link to a malware report at ThreatExpert that has also since been removed.

http://krebsonsecurity.com/wp-content/uploads/2014/01/poswds-threatexpert-report.pdf

What can we learn about the Target breach by studying the reports that have been shared in these two locations? Especially when considered in light of what is known of previous data breaches? First, we will consider a possible scenario involving the Heartland Payments breach. Then we will look at the tools revealed by the iSight document from the Wall Street Journal URL.Lastly, we will consider what the ThreatExpert report means with regards to the limitations of anti-virus software as a possibly useful tool for detecting such breaches.

Techniques that may be similar to the Target breach were used by the Alberto Gonzalez gang, as illustrated in an indictment against Vladimir Drinkman, Aleksandr Kalinin, Roman Kotov, Mikhail Rytikov, Dmitriy Smilianet. In the indictment, which was only unsealed on December 17, 2013, Drinkman et al were called co-conspirators of Alberto Gonzalez (famous for the TJX breach), Damon Toey, and Vladislav Horohorin (BadB). Drinkman and his gang of Russian hackers were active from at least August 2005 through at least July 2012 and were charged with stealing data from NASDAQ, 7-Eleven, Carrefour, JCPenney, Hannaford Brothers, Heartland Payment Systems, Wet Seal, Commidea, Dexia Bank, JetBlue Airways, Dow Jones, an unspecified bank in Abu Dhabi, Euronet, Visa Jordan, Global Payment Systems, Diners Singapore (a regional branch of Diner's Club), Ingenicard. In each of these cases, an SQL Injection attack resulted in malware being placed on the network and credit card or personal information being exfiltrated from the network.

According to the indictment for the above, Gonzalez and Toey would travel to retail outlets and make observations about which Point of Sale terminal software was being used, afterwards, they would pass the information to the hacker crew who would penetrate the network, customize and load the malware, and exfiltrate the stolen data. The breaches continued even after Gonzalez and Toey were in custody. Here is a timeline of those cases:

NASDAQ – initial point of entry, an SQL injection in the “Password Reminder” website (May 19, 2007). By August 12, 2007 the hackers communicated that they had cracked the admin password and identified 30 SQL servers and “we can run whatever on them.”

7-Eleven - (August 2007) - SQL Injection lead to malware that extracted card data from databases)

Carrefour S.A - (October 2007) - - SQL injection lead to malware that extracted card data for 2 million credit cards

JCPenney - (October 2007) - SQL Injection lead to malware placed on the network that extracted card data from databases

Hannaford Brothers - (November 2007) - SQL Injection lead to malware placed on the network that extracted card data for 4.2 million credit cards

Heartland Payment Systems (December 2007) - SQL Injection lead to malware placed on the network that extracted card data for 130 million credit cards, leading to $200 Million in losses

Wet Seal - (January 2008) - SQL Injection lead to malware placed on the network that extracted card data from databases

Commidea Ltd. - (March-November 2008) - malware was used to extract card data and exfiltrate the data for 30 million credit cards

Dexia Bank Belgium - (February 2008 to February 2009) - SQL Injection resulted in malware placed on the network that exfiltrated card data leading to a $1.7 Million loss

JetBlue Airways - (Jan 2008 - February 2011) - malware placed on network exfiltrated Personal Data of employees

Dow Jones, Inc. - (2009) - at least 10,000 sets of Log-In Credentials stolen via malware placed on network

"Bank A" - (Dec 2010 to March 2011) - malware placed on an unnamed bank HQ'ed in Abu Dhabi, United Arab Emirates used to facilitated theft of Card Numbers.

Euronet - (July 2010 to October 2011) - SQL injection lead to malware that extracted login credentials resulting in more than 2 million cards being stolen

Visa Jordan Card Services - (Feb 2011 to March 2011) - SQL Injection lead to malware placed on network that exfiltrated card data for 800,000 cards

Global Payment Systems - (January 2011 to March 2012) - SQL Injection lead to malware placed on network that exfiltrated card data for 950,000 cards, leading to $92.7 Million in losses

Diners Club International, Singapore - (June 2011) - SQL Injection lead to malware placed on network that exfiltrated card data for 500,000 Diners Club cards resulting in $312,000 in losses

Ingenicard US, Inc. - (March 2012 to December 2012) - SQL Injection resulted in malware placed on the network that was used to facilitate ATM withdrawals of more than $9 million in 24 hours

I. Heartland Payments

One of the first truly sizable carding breaches happened in 2007, when data on 130 million credit cards was stolen. As an active security blogger at that time, I received an anonymous email that walked through some of the back story of the breach. While I cannot confirm the details, it certainly seems a plausible method of attack. So, please consider the story below as a PLAUSIBLE method for the Heartland Payments breach, rather than a confirmed factual account.

First, Heartland Payments had a public facing website that provided a front-end to a Microsoft SQL Server. According to our possible scenario, a hacker exploited the SQL server through SQL Injection and was able to cause a command to execute on the server, because of MS-SQL’s ability to execute a “system call” to launch a third party command with the authority of the MS-SQL userid. In this case, the command that was given was to use “TFTP” (Trivial FTP) to download a copy of NetCat from a webserver in Russia. NetCat is a tool that allows a computer INSIDE the network to establish an OUTBOUND connection to an OUTSIDE address which then is provided a network terminal allowing the person at the remote computer to issue commands in what appears to be a DOS CMD Window running on the inside computer.

After executing NetCat, the remote user was able to issue commands to download several tools. They downloaded a password hash dumper and a port scanner. After dumping the cache of userids and password hashes from the local machine, the hacker realized that a domain-level admin account, VERITAS, existed on the local machine. The password cracked easily, as it was actually the default password for the Veritas backup software – BACKUP. Distributed network backup systems have to have administrative privileges on the computers that they are scheduled to backup, because they need to be able to backup all of the files on the server, regardless of who owns the local file.

After using the Administrator equivalent account to create their own userids, the hackers went exploring, using their port scanner to identify other database servers on the network and interacting with them through an SQL Query tool to learn more about their structure and contents. Scripts were created to schedule certain queries to dump credit card related data from the databases to “flat files”

which were then zipped up and exported from the network via an outbound FTP File Transfer call, which posted the daily card transactions to an external network site from which the hacker could retrieve them. This script ran daily for many months before being discovered.

II. iSight Partners list of Target files found

When we retrieved the PDF file from the Wall Street Journal website, we created a list of all the hashes that were listed in the document. We then went exploring to find out whether any of these were well-known malware files. We started by submitting the entire list to VirusTotal to search to see whether the file was already known by VirusTotal. Only five of the files had been previously submitted and detected, but none of these were actually malware. Seven additional files had been submitted and declared to be “safe” by all anti-virus applications. In the table below you can see the hashes, the VirusTotal detection ratio, and some notes by Malcovery about what the tool or file actually is. In several cases the clue to file identification came from the “Additional Info” tab at VirusTotal where all of the names of all submitted files are provided.

MD5 Hash from iSight Report	VirusTotal Detection	Type	Notes
d975fc6cda111c9eb560254d5eedbe0a	VT 29 of 42	HackTool	Portforward.exe found in Portforward.rar and Portforward_bin.zip
814b88ca4ef695fea3faf11912a1c807	UNKNOWN		
df5dbcbcac6e6d12329f1bc8a5c4c0e9	0 of 45	benign	“osql.rll” Microsoft SQL Server file
4b9b36800db395d8a95f331c4608e947	0 of 49	benign	“osql.exe” aka “CiscoLog.exe” (Version 1998.11.13.0) normally installs in \mssql7\binn\. Frequently repackaged for distribution with MS-SQL based packages.
02137a937f6fbc66dbc59ab73f7b1d3e	0 of 47	Benign NIST	“osql.exe”
f4bdc5e507d887d5d2cd2c4c61cfcfe1	0 of 46	Benign	“OrchestratorRun ProgramService.exe” Part of the Microsoft System Center Orchestrator.
a35e944762f82aae556da453dcba20d1	UNKNOWN		
322e136cb50db03e0d63eb2071da1ba7	24 of 48	HackTool	NetCat – netc.exe
e2db09553f23a8abc85633f6bf1a0b49	UNKNOWN		
290c26433a0d9d14f1252e46b1204643	UNKNOWN		
623e4626d269324da62c0552289ae61f	UNKNOWN		
453810a77057d30f0ee7014978cdc404	UNKNOWN		
6c1bcf0b1297689c8c4c12cc70996a75	18 of 48	HackTool	Angry_ip_scanner_2_21.exe
0b33b4d61ea345f16c4a34b33e9276bc	UNKNOWN		
4352e635046aa624dff59084d5619e82	UNKNOWN		
65dd8d2d9604d43a0ebd105024f09264	0 of 46	HackTool	Somarsoft DumpSec – “dumpsec is a security auditing program for Windows. It dumps the permissions (DACLs) and audit settings (SACLs) for file system, registry, printers, and shares in a concise readable format, so that holes in system security are readily apparent.

			www.systemtools.com/somarsoft/index.html
433a2750429d805907aa4848ff666163	UNKNOWN		
3f00dd56b1dc9d9910a554023e868dac	0 of 48	Benign NIST	BCP.exe (Microsoft product) BCP stands for Bulk CoPy. It is a Microsoft utility for dumping data from SQL servers: technet.microsoft.com/en-us/library/ms162802.aspx
93405c57e915680f0182650fb75c47ee	UNKNOWN		
ab6fb405ef8f06ee98be0b9da5250607	UNKNOWN		
793860864d74ee6ed719d57b0a3f3294	9 of 46		PPA_setup_en.msi (ElcomSoft password cracking tool). Proactive Password Auditor. See: www.elcomsoft.com/ppa.html
aeee996fd3484f28e5cd85fe26b6bdcd	1 of 48		PsExec.exe (included in password cracking kits such as "mimikatz_trunk.zip" demonstrated in this YouTube video: www.youtube.com/watch?v=leKRxe4bZ6M but more commonly seen associated with SysInternals tools.)
2cd8dddaf1a821eeff45649053672281	UNKNOWN		QuarksPwDump_cr.exe
a109c617ecc92c27e9dab972c8964cb4	0 of 47		QueryExpress.exe - www.albahari.com/queryexpress.aspx 100kb exe that allows queries against SQL server as a stand-alone
f6877447d2bd0199ad2f073a391aacde	UNKNOWN		
e2db09553f23a8abc85633f6bf1a0b49	UNKNOWN		

Given that list of files and what we know about the possible scenario of the Heartland Payments breach, we can describe a likely scenario for the Target Breach. By reviewing this scenario, we hope readers will gain some insights into possible improvements in their own security for detecting such a breach attempt in the future on their own network.

a. Getting tools on the net - NetCat

While we do not know how the hackers got in at Target, once they were in, NetCat.exe (possibly named something else) was retrieved from the network. If this breach is similar to Heartland Payments, NetCat would have been a primary way in which the hackers were able to load up their tools onto the internal network in order to perform the reconnaissance necessary to customize the malware that we know they put into place. The MD5 of the NetCat executable is 322e136cb50db03e0d63eb2071da1ba7 , found in the list provided by iSight Partners.

b. Passwords – we need some!

PsExec.exe, the file iSight identified as aeee996fd3484f28e5cd85fe26b6bdcd, is a tool originally by SysInternals, now owned and marketed by Microsoft, which allows a system administrator to export a list of password hashes from a Windows computer. Several tutorials on hacking explain how this tool can be used in conjunction with other tools to identify the passwords of Windows accounts. Quarks PW Dump tool (see description in Figure 1) was also among the files reported by iSight. While PsExec is great for identifying userids for the local machine, Quarks adds the capability to dump cached Domain passwords.

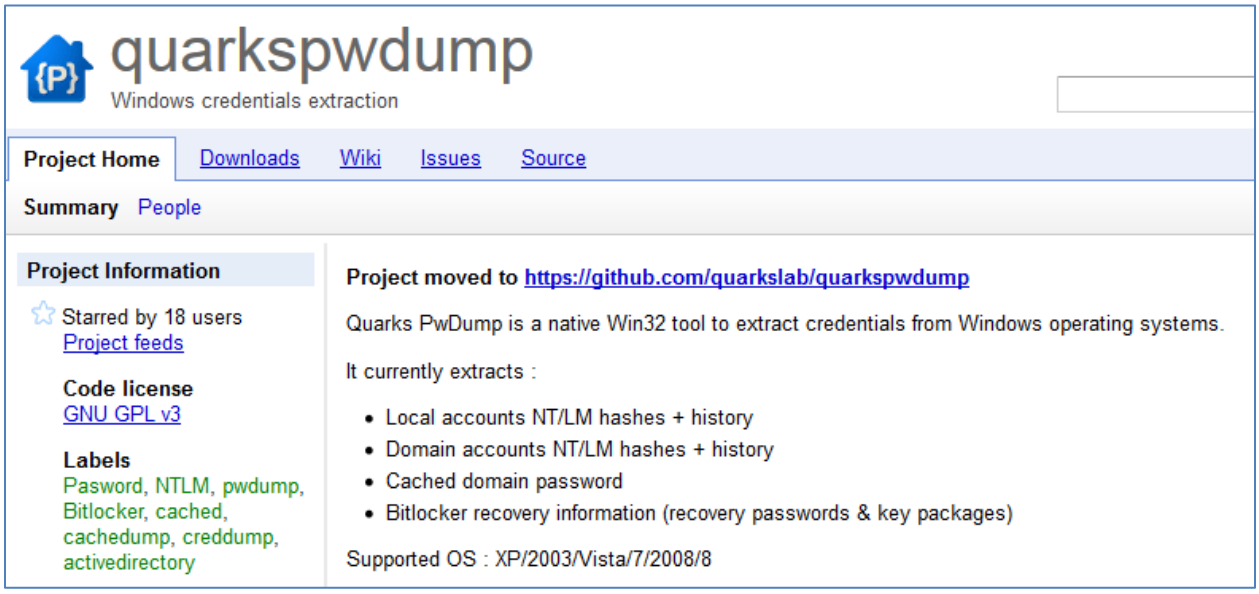

Figure 1. Quarks PwDump - *https://code.google.com/p/quarkspwdump/*

One excellent tool for testing the security of Windows passwords is from ElcomSoft, a Russian security company whose excellent tools for cracking both Windows, Archive (zip files), and Office files have helped me on many occasions when an employee departed from my organization or forgot a critical password. The file that iSight identified by the MD5 hash 793860864d74ee6ed719d57b0a3f3294 is the Microsoft Installer file of the production version of PPA, Elcom’s Proactive Password Auditor. (See Figure 2 for description).

Figure 2 Proactive Password Auditor - *www.elcomsoft.com/ppa.html*

Once we have identified an administrative account, the next objective would be to begin exploring which userids on which systems have special privileges and also to begin to explore file collections that may be of high value. One tool mentioned (by its hash) in the iSight report is SomarSoft’s DumpSec (See Figure 3). SomarSoft’s website says that this tool is “a security auditing program for Windows” that “dumps permissions (DACLs) and audit settings (SACLs) for the file system, registry, printers, and network shares in a concise, readable format, so that holes in system security are readily apparent. DumpSec also dumps user, group, and replication information.”

Figure 3. SystemTools DumpSec - *www.systemtools.com/somarsoft/*

c. Databases – find them and dump data from them

The next objective of a hacker who is seeking databases is to find the database servers. One common way to do this is to use a Port Scanning tool to look for open network ports that are associated with database servers. Did the Target breach hackers use this method? While we can’t say for certain, not having been involved in the investigation, it is certainly a likely scenario, especially given the identification of another file reported by iSight Partners.

The hash 6c1bcf0b1297689c8c4c12cc70996a75 is used by the file for “Angry IP Scanner version 2.21”, which is a very aggressive and nicely configurable port scanner. Using this tool, it would be easy for a remote hacker to scan the network to identify any other database servers on the network.

After the database servers have been identified, the hacker would likely attempt to connect using various administrative passwords and/or passwords that have been recovered from other portions of the Windows domain. We see evidence that the hacker has prepared to connect and query Microsoft SQL server in four of the files identified in the iSight Partners report. Three of the files are actually from Microsoft, two versions of “osql.exe” (a command line SQL query tool) and “osql.dll” (required for osql.exe to function) were found by matching the MD5s. BCP.exe, a Microsoft tool for doing “Bulk

Copy" of data out of an SQL server was also found, as was Albahari's Query Express. The advantage of Query Express is that a single 100kb file can be used to query and retrieve data from SQL servers without having to run any installations or having administrative privileges.

d. Exfiltrate the Data

Assuming that the intruder is able to identify and dump data from databases, the next challenge is exfiltrating the data not just currently but on an on-going basis. Among the files found were a Microsoft System Center Orchestrator (sometimes known as SCORCH) file used for scheduling and launching services. From the ThreatExpert report (see below) we know that malware was being executed AS A WINDOWS SERVICE. It is possible that SCORCH was used to invoke the actual malware and ensure that the malware would be regularly re-executed.

The presence of "PortForward.exe" is also sometimes used to get past local Access Control Lists. If a server or service is only allowed to write to places on the local network, by using PortForward, an exposed local port can serve as a proxy to a remote location. It is also possible that PortForward worked in the opposite direction, allowing an exposed port that had firewall permissions to allow inbound access to be used to redirect traffic to an internal resource that lacks permission to pass through the firewall.

III. Threat Expert Report

Threat Expert is an online resource where malware files can be uploaded and automatically analyzed by the Threat Expert system. From a privacy perspective, it can be problematic if a ThreatExpert report is generated on malware that has been designed for a custom attack. This seems to be what occurred in this situation, as the ThreatExpert report has been removed from their website. Security researcher and journalist Brian Krebs retrieved a copy from Google Cache prior to it being fully purged and has saved a copy on his website.

The malware was uploaded to ThreatExpert on December 18th, 2013 to be analyzed. The MD5 for the malware was: CE0296E2D77EC3BB112E270FC260F274 and the file was 270,336 bytes. The malware launches a new service, called POSWDS, which is quite likely the search term Krebs used to find it. A hard-coded internal network address, 10.116.240.31 port 80, was found in the malware. The malware also mounts a shared network drive, 10.116.240.31, mapping the administrative file share "C$" and the path \Windows\Twain_32. The network drive map connects using a domain name "TTCOPSCLI3ACS" with the hard-coded userid and password "Best1_user" and "BackupU$r".

It is possible that, similar to the HeartlandPayments system, the administrative share of the entire drive had been previously established to allow backup software to backup that server's full hard drive. The latter is hinted at by the fact that the password is "BackupU$r" – "Backup User?"

Many of the additional files that were listed in the iSight Partners report remain unknown and unknowable for us as outsiders to this investigation, but we believe understanding the set of files laid

out above may be helpful for investigators who are trying to determine whether similar vulnerabilities exist on their own networks.

IV. Lessons Learned

Let's review some of the things that went wrong at Heartland Payments and that may have gone wrong at Target, as examples of questions we could ask about our own networks. Again, Malcovery is not involved in the Target investigation, and we certainly defer to iSight Partners, who played an active role for a fuller set of recommendations. However, some observations are worth noting as general best practices questions, regardless of the specific malware in the Target case.

Q1. Do you have a process for identifying "Administrative Tools"that are stored in the wrong places?

Many of the tools used above are NOT DETECTED AS MALWARE because they are not malware. They are network or database administrative tools that have legitimate uses as well. Searching for these tools on YOUR network may help to identify users who have tools that do not match their roles. Tools such as Bit9, or really any good network inventory system, could help to identify the location of network and database administrative tools that are on a machine owned by a user who does not have those responsibilities. Even version matches are important. The OSQL.exe files are from 1998. No one should have that version present in a production environment.

Q2. Are your servers restricted from accessing the Internet?

In the Heartland Payments breach, a server was used to execute a download of NetCat, and then that same server was used to make outbound network connections. Server class machines should never have firewall rules that allow "general network use." The argument that servers need to have internet access to retrieve patches is doubly negated. First, patches should be distributed from an INTERNAL patch management system, after appropriate testing. Secondly, if direct network access is needed, the DESTINATIONS THAT SERVE UP PATCHES should be hardcoded as the only legitimate outbound traffic allowed from these devices. In both the Target breach and the Heartland Payment breach, the method of data exfiltration was an FTP command. Again, there should be no circumstance where the firewall would allow an FTP command from a server class machine. In addition, because FTP is an insecure protocol that passes userid and password information in plain text on the wire, any internal machine originating an outbound FTP should be reviewed to determine if there is a business reason for the FTP to be occurring.

Q3. Is there an inventory of "authorized services" for your servers?

In the Target case, we know that the malware was running as a Windows Service named POSWDS. While it is unknown to this analyst if this was a customary service name for the Point Of Sale service within Target, it certainly would have been a wrong binary to be running as a service.

Q4: Are system folders being checked for new or changed files?

In the Target case, the data collector was storing information in the Windows folder under the subdirectory "\twain_32" with the stolen information being disguised as a ".DLL" file.

Q5: Is there a way to check for unusual protocols and ports being used for internal LAN communications?

In the Target case, many of the binaries, according to iSight’s report, used ICMP to communicate with one another. ICMP is most commonly associated with the PING command, and is very rarely logged in standard network logging. A periodic check for atypical network protocols being used on the network may be worth considering.

Q6: Are internal network honeypots in place to look for port-scanning and other signs of an intruder?

In all of the data breaches mentioned in this paper, a key aspect of the criminal’s reconnaissance was identifying and mining the internal SQL servers. It would certainly be a worthwhile practice to deploy “honeypot” versions of internal servers. A criminal intruder port-scanning for databases would certainly encounter your honeypot if it were listening on standard ports. This could be an early warning system that someone is hunting inside your network.

