Interested in learning more about security?

SANS Institute InfoSec Reading Room

A Guide on How to Find Cardholder Data without Automated Tools for PCI Assessors

The PCI Data Security Standard requires organizations to determine the scope of their compliance obligation accurately. A critical aspect of PCI DSS scope definition is identifying all the locations where cardholder data is stored. During the course of an assessment, PCI Assessors must validate that the perceived compliance scope is in fact accurately defined and documented. Automated discovery tools, while effective to find cardholder data, sometimes are not an option due to the negative impact they may have in a prod...

A Guide on How to Find Cardholder Data without Automated Tools for PCI Assessors

GIAC GSNA Gold Certification

Author: Christian J. Moldes, christian.moldes@hotmail.com
Advisor: Richard Carbone

Accepted: September 15, 2014

Abstract

The PCI Data Security Standard requires organizations to determine the scope of their compliance obligation accurately. A critical aspect of PCI DSS scope definition is identifying all the locations where cardholder data is stored. During the course of an assessment, PCI Assessors must validate that the perceived compliance scope is in fact accurately defined and documented. Automated discovery tools, while effective to find cardholder data, sometimes are not an option due to the negative impact they may have in a production environment. In this paper, the author discusses audit techniques and tips on how to find cardholder data without using automated tools.

1. Introduction

It has been seven years since the PCI Security Standards Council (PCI SSC) was founded and the PCI Data Security Standard (PCI DSS) published to provide a minimum set of required security controls to protect cardholder data.

PCI DSS, now on its third version, requires organizations to determine the scope of their compliance obligation accurately. A critical aspect of PCI DSS scope definition is identifying all the locations where cardholder data is stored. During the course of an assessment, PCI Assessors must validate that the perceived compliance scope is in fact accurately defined and documented (PCI SSC, 2013)

Automated discovery tools, while effective to find cardholder data, sometimes are not an option due to the negative impact they may have in a production environment. Therefore, it is essential for PCI assessors to improve their audit and discovery skills to perform manual cardholder data discoveries.

In this paper, I share audit techniques and tips I learned throughout my career so PCI assessors can improve their audit and discovery skills to find cardholder data without using automated tools.

2. About the Author

For the past seven years, I have been performing PCI compliance assessments and gap analysis for organizations in several different vertical markets. Throughout my career as a QSA (Qualified Security Assessor), I have been part of at least 100 PCI assessments in several countries.

I worked for five years for Verizon Business as a Senior Security Consultant specialized in PCI DSS assessments, led the PCI compliance practice at IOActive for about two years, and recently joined IBM in a similar role.

I wrote several papers regarding PCI compliance in topics such as incident handling, contracting, security policies, and auditing ASP.NET applications for PCI DSS compliance. These papers are available on the SANS Reading Room website.

Christian J. Moldes, Christian_moldes@hotmail.com

All the case studies in this paper are based on my personal experience auditing organizations as a PCI QSA. The case details have been generalized as much as possible and slightly modified to preserve the confidentiality of my clients.

3. Key Skills to Find Cardholder Data Manually

Four basic skills are needed to finding cardholder data manually:

- Professional skepticism;
- Understanding of the organization's business processes;
- Understanding of the interviewee's roles and responsibilities;
- Understanding of the technology used by the organization.

3.1. Professional Skepticism

"Trust but verify" should be the mantra of any PCI assessor. The main reason why cardholder data may remain uncovered is that sometimes assessors take their interviewee's assertions at face value. Professional skepticism is an essential attitude.

Auditees are not deliberately trying to hide unprotected cardholder data from the assessor. In almost all cases where I found undisclosed cardholder data, the organizations falls into one of the following categories:

a) They performed a poor data discovery process by not identifying all the systems in scope and all locations where cardholder data could potentially be stored.

b) They relied on the statements made by other business units, departments, or people without performing any validation themselves.

c) They did not have a clear understanding of how the scope for PCI DSS compliance is defined and left some areas or system components out of the discovery process.

d) They did not fully understand the technology they are using, and how it may cause cardholder data to be stored.

e) They did not understand the PCI DSS requirement to render cardholder data unreadable and the acceptable technologies to do so.

Therefore, remaining skeptical of any unverified claims is not only a healthy habit for an assessor but a key factor to find any unknown or undisclosed locations where cardholder data is being stored.

3.2. Understanding the Organization's Business Processes

One of the major challenges of being an external assessor for the payment card industry, or QSA, is that most of the time external assessors are expected to understand an organization's business processes better than the organization's staff themselves. While it is impossible for that to occur by only spending a couple of weeks or less reviewing an organization's compliance status, a seasoned assessor would have been previously exposed to different types of organizations, business processes, and technologies and should be able to extrapolate these experiences to the environment under review.

There are common business and payment processes that most assessors will quickly identify and appropriately review such as authorization, capture, settlement, and chargebacks processes. However, unless assessors cover all the business processes applicable to an organization; potentially, cardholder data could be left uncovered.

The following cases demonstrate how understanding the auditee's business processes leads to finding undisclosed cardholder data.

3.2.1. Revenue Streams and Sales Channels

Assessors must do their best to understand all the organization's revenue streams and sales channels.

Merchants' revenue streams are typically generated by selling products and services through different sales channels. For merchants, it is obvious that any type of sale, regardless of the channel, may potentially involve sending, receiving, or accessing previously stored cardholder data.

The following table lists channel types and typical areas where cardholder data could be stored, processed, and/or transmitted:

Table 1 - Sales Channel Types - Typical Locations for Cardholder Data

Channel Type	Typical Locations for Cardholder Data
Sales Force	CRM Solutions, call recordings, call center systems, hardcopy forms, faxes, and e-mails
E-commerce	E-commerce applications and databases, and web servers
Mail Orders	Mail orders and imaging systems
Stores	Points of Sale registers, self-service kiosks, backoffice servers, receipts, imprint forms, and registration forms
Special Sales	Systems used for occasional events, tradeshows, seasonal stores, charity events, etc.
Partners	Whole sale processes B2B applications and databases, hardcopy forms, faxes, and e-mails

Organizations other than merchants may have similar channels. For example, charities and non-profit organizations may not have a "Sales force" per se but instead a "Fundraising team" that uses processes and technologies similar to for-profit organizations.

Failing to identify all sales channels may cause cardholder not to be found.

Real Case Study: PCI DSS only requires encryption during transmission when transmitting cardholder data over open, public networks. Many organizations sometimes do not understand that while cardholder can be sent in the clear over the internal network, using e-mail to do so expands the compliance scope unnecessarily. While e-mails can be deleted immediately once read, there is no assurance that users will consistently do so. They may potentially archive them in their local e-mail database or forward it to other users who are not aware of the deletion policy for e-mails containing cardholder data. The chargebacks and settlement team of a merchant I assessed was sending files containing cardholder data via e-mail internally. During the interviews, the staff assured that they delete these e-mails as soon as they are received. We agree to review the workstation later that day. When I inspected the interviewee's e-mail inbox, we did not find any e-mail containing cardholder data. However; I asked to see the deleted e-mails folder and we found deleted e-mails containing cardholder data spanning several weeks. What I found interesting was that the deletion time and date for all those e-emails was a few minutes after our meeting ended earlier that day. (Moldes, 2014)

Christian J. Moldes, Christian_moldes@hotmail.com

3.2.2. Customer Segments

Merchants may handle sales to specific customer segments differently. For example, luxury brands usually assigns sales executives to high-networth clients. This usually occurs on markets where these types of clients live in affluent areas such as Beverly Hills, New York City or San Francisco.

> *Real Case Study: While assessing a luxury brand merchant, I found that the sales executives were keeping cardholder data for their VIP clients on their own personal phone books. For this type of clients, the sales executive was shipping merchandise that meets their VIP clients' known preferences. The VIP clients keep what they like and return unwanted items. The sales executive then charge the VIP client's credit card on record.*
>
> *These phone books were extremely valuable to the executives; however, a huge risk to the organization if they were not protected adequately. (Moldes, 2014)*

An assessor have to think about cases where salespeople may keep cardholder data to facilitate transactions for themselves or their clients. For example, seniors, children, students, VIP clients, etc.

3.3. Understanding the Interviewee's Role and Responsibilities

Many users may view security as affecting business processes negatively by making processes more difficult, slower, and unnecessarily complex. It should not be a surprise that if users can bypass a security control, they will. They will always adopt the principle of least effort.

Assessors must put themselves in their interviewee's shoes and ask themselves: "If I would have to do his/her work, what would be the easiest way to do it, not taking into consideration security at all?"

The following cases illustrates examples of how understanding roles and responsibilities leads to find undisclosed cardholder data.

3.3.1. DBAs

DBAs are privileged users and because of the nature of their work, most of the time they have unrestricted access to data. DBAs are often tasked to manipulate and transform data, export it, and to obtain backups in different formats.

The easiest way for a DBA to work with data is to use the database tools to select, order, and extract data. During this process, it is common for them to create temporary or working tables within the same database. Such temporary tables may potentially remain on the system if not purged after the task has been completed. Assessors should look for this type of tables, usually; they may have names that reflect their temporary purpose.

> *Real Case Study: A DBA was tasked with the responsibility to rotate the cryptographic keys used to encrypt cardholder data. Having no automated process in place, the DBA had to manually decrypt and re-encrypt the data with the new cryptographic key. A temporary table was created to host the decrypted data. A script took each entry in the temporary table, re-encrypted it, and copied the encrypted data to the original table. The production team had to verify that the process has been successful so the data in clear text could not be removed immediately.*
>
> *When the process was completed, the DBA forgot to delete the temporary table and nobody found it until the annual PCI DSS assessment was conducted. (Moldes, 2014)*

In addition to the database itself, DBAs can also store temporary data files in directories or folders on the local database server. Assessors must review these directories or folders. Since a tool may not be available to review all directories, assessors should select directories based on an educated guess. The following are common locations were I found data files containing cardholder data in the past:

- The DBA's home directory;
- The database administrator account's home directory (e.g. oracle and informix);
- Developers' home directory;

- The root directory;
- Backup directories;
- Application code directories;
- Script directories.

Real Case Study: When reviewing a SQL database server, I spent a few minutes inspecting the local directories. A compressed .rar file was found in one of the folders, and after inspecting the contents, I found a database export containing payment card transactions. The transactions were more than three years old, and most cards were already expired. However, it is interesting that such a file could have remained on the local server for years without anybody noticing. The DBA was unable to provide a business justification for that file. (Moldes, 2014)

When finding files containing cardholder data, assessors have to take into consideration the possibility that these files could be traces of a security breach. The incident response team may need to get involved to determine whether that is the case.

Most organizations that initiate remediation plans to achieve PCI DSS compliance opt for encrypting data repositories that contain cardholder data. Applications are modified and data begins to be stored in an encrypted format. Assessors have to verify that remediation plans included both data before and after such a change was implemented.

Real Case Study: A level-2 merchant recently had updated their application to encrypt payment card transactions. During the database review, the DBA selected the most recent entries in the database. When asked to select the oldest entries, it was found that previous data was not encrypted. While transactions were now being encrypted, the organization did not take into consideration encrypting legacy data. (Moldes, 2014)

Real Case Study: During an assessment for a large organization, my partner and I were interviewing a business process owner. He was explaining how encryption was recently implemented and showed us a column named PAN containing encrypted data. My partner deemed this sufficient for him to move on to a different subject and he was about to end the meeting. However, I asked the interviewee to show us the rest of the table columns. To the surprise of my partner and the interviewee, after browsing a few columns, a column containing PANs in clear text appeared. The interviewee was speechless as it was his understanding that his team had implemented encryption and removed any existent data in clear text from the table. (Moldes, 2014)

These two cases show that assessors should verify both recent and old data, and all the columns of tables that contains cardholder data.

3.3.2. Developers

Developers need data to test their applications. Some processes may require complete data that cannot be easily recreated. The least amount of effort for developers is to copy production data, especially, if they already have access to the production environment to support production processes. In the worst-case scenario, developers would not only copy data but also store it in insecure locations.

Real Case Study: I was interviewing a developer who had responsibilities to support the production environment. I asked him about the tools used to connect to the database to support it remotely. I asked him to demonstrate how he connects to the database and he double-clicked on a desktop link. When asked where on his workstation the data is stored when it is downloaded using the tool, the developer mentioned a specific directory. We reviewed the directory and found database export files containing cardholder data. The developer mentioned that once a year data is downloaded to create sanitized data for testing purposes and that he forgot to delete the source file after the last download. (Moldes, 2014)

3.3.3. General Users

For issuing entities such as merchant banks, debit and credit cards that have yet to be delivered to the cardholder are assets that are carefully guarded. It is not unusual for custodians to be required to maintain a log of all issued cards currently in custody and have cardholders acknowledge reception of their cards in writing when received.

> *Real Case Study: At a merchant bank, the procedure at the branches required cards to be inventoried daily and a chain of custody maintained. I found that the staff was using a book to record the full card numbers, cardholder name, and other sensitive details to inventory cards. They were also asking cardholders to acknowledge reception of their cards in writing. In other cases, I found that similar records were maintained in an Excel spreadsheet stored on the custodian's workstation. (Moldes, 2014)*

The above scenario can occur in retail environments as well. Because of excessive due care and the sensitivity of cards, employees may carry unofficial procedures to handle physical cards.

> *Real Case Study: At retail stores, it is not unusual for customers to forget their cards at the cashier desk. At one of the stores of a large retailer, one employee was assigned custody of any cards that customers forget at the premises. This individual, out of his sense of due care created and maintained a book recording all the forgotten cards in his custody in a book. He was recording the full card number, name, and expiration date in the book. (Moldes, 2014)*

3.4. Understanding the technology used by the organization

Assessors often fail to identify undisclosed locations for cardholder data because they are not familiar with specific technologies used by the organization they are assessing. The following is not an exhaustive list but a sample that will illustrate this point.

3.4.1. GET vs POST

Web based applications and even fat-client applications using HTML for communication can post data using either GET or POST HTML commands. Assessors not familiar with web-based applications may not inquire about how cardholder data is being transmitted to the web server.

Posting data with GET makes data transmitted with this command much more insecure than posting data with POST commands. Diffen.com published an article listing all the differences between these commands (Jasuja, Sehgal, & Balram). GET presents the following security risks that may affect PCI DSS scope:

Table 2 - GET vs POST

Category	Description	Security Risk
History	Parameters remain in browser history because they are part of the URL	Data stored in the workstation browser history
Bookmarks	Can be bookmarked	Bookmarks may potentially include cardholder data
Parameters	Data sent is part of the URL	Data stored in the local browser history and server logs in clear text
Cache	Can be cached	Data stored in the workstation cache data

The following figure shows an excerpt from an IIS web log that shows how sensitive information such as passwords could be recorded in web logs because GET is used to post data.

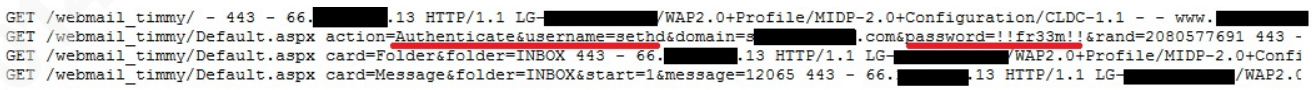


Figure 1 - Microsoft ISS web log. Obtained from the Internet after searching on Google for "filetype:log GET". Redacted to preserve the confidentiality of the misconfigured website.

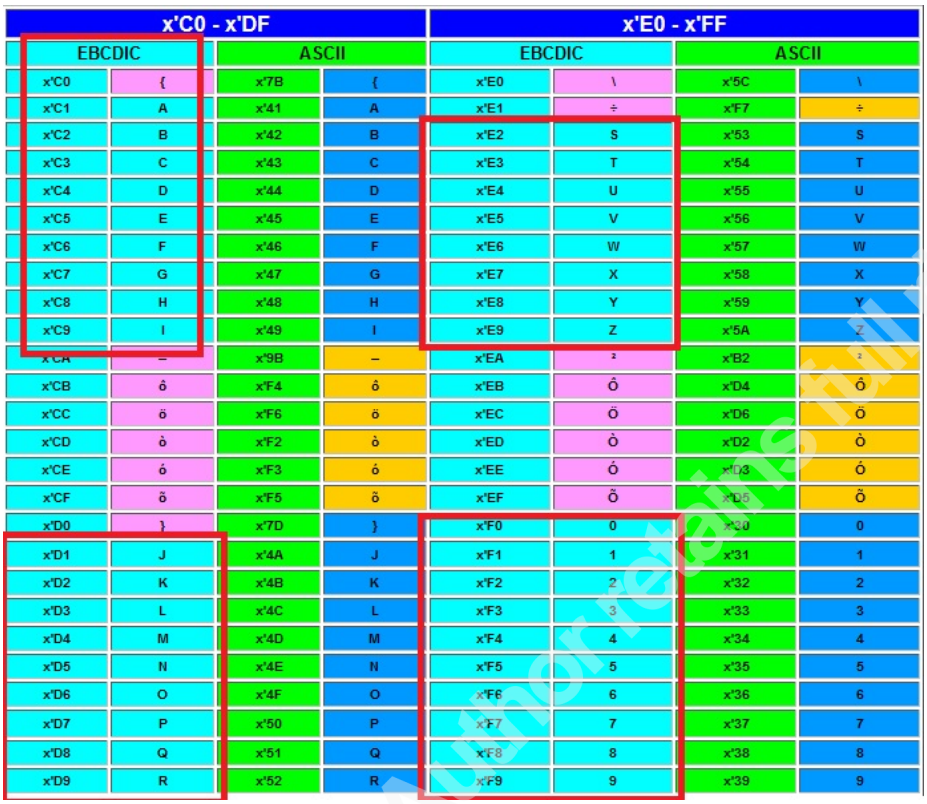
Real Case Study: A payment processor successfully underwent a couple of assessments led by a QSA. During their last assessment, cardholder data in clear text was found in the Microsoft IIS web logs. It was evident that previous assessors did not inspect the IIS log

contents. After researching why cardholder data was being recorded in this log, it was determined that different versions of the application were still in production. Some versions were using POST and others GET to post data to the web server. Whenever GET was used, form fields containing cardholder data were appended to the URL and therefore recorded in the access logs. (Moldes, 2014)

3.4.2. Lost in Translation: EBCDIC

EBCDIC (Extended Binary Coded Decimal Interchange Code) is a data-encoding system, developed by IBM that uses a unique eight-bit binary code for each number and alphabetic character as well as punctuation marks and accented letters and non-alphabetic characters. EBCDIC differs in several respects from ASCII, the most widely used system of encoding text, dividing the eight bits for each character into two four-bit zones, with one zone indicating the type of character, digit, punctuation mark, lowercase letter, capital letter, and so on, and the other zone indicating the value (that is, the specific character within this type). (Encyclopædia Britannica, 2014). EBCDIC was developed in the late 1950s and early 1960s. (Bemer)

When dealing with EBCDIC most organization convert data to Hexadecimal to facilitate data operations. In addition, in organizations where mainframes and open systems intercommunicate, a translation from EBCDIC to ASCII characters have to be performed for these systems to communicate. The following figure shows a mapping table that converts data from EBCDIC to ASCII including a Hexadecimal mapping for each character set. This table was obtained from http://www.flounder.com/ebcdictoascii2.htm.



x'C0 - x'DF				x'E0 - x'FF			
EBCDIC		ASCII		EBCDIC		ASCII	
x'C0	{	x'7B	{	x'E0	\	x'5C	\
x'C1	A	x'41	A	x'E1	÷	x'F7	÷
x'C2	B	x'42	B	x'E2	S	x'53	S
x'C3	C	x'43	C	x'E3	T	x'54	T
x'C4	D	x'44	D	x'E4	U	x'55	U
x'C5	E	x'45	E	x'E5	V	x'56	V
x'C6	F	x'46	F	x'E6	W	x'57	W
x'C7	G	x'47	G	x'E7	X	x'58	X
x'C8	H	x'48	H	x'E8	Y	x'59	Y
x'C9	I	x'49	I	x'E9	Z	x'5A	Z
x'CA	–	x'9B	–	x'EA	²	x'B2	²
x'CB	ô	x'F4	ô	x'EB	Ô	x'D4	Ô
x'CC	ö	x'F6	ö	x'EC	Ö	x'D6	Ö
x'CD	ò	x'F2	ò	x'ED	Ò	x'D2	Ò
x'CE	ó	x'F3	ó	x'EE	Ó	x'D3	Ó
x'CF	õ	x'F5	õ	x'EF	Õ	x'D5	Õ
x'D0	}	x'7D	}	x'F0	0	x'30	0
x'D1	J	x'4A	J	x'F1	1	x'31	1
x'D2	K	x'4B	K	x'F2	2	x'32	2
x'D3	L	x'4C	L	x'F3	3	x'33	3
x'D4	M	x'4D	M	x'F4	4	x'34	4
x'D5	N	x'4E	N	x'F5	5	x'35	5
x'D6	O	x'4F	O	x'F6	6	x'36	6
x'D7	P	x'50	P	x'F7	7	x'37	7
x'D8	Q	x'51	Q	x'F8	8	x'38	8
x'D9	R	x'52	R	x'F9	9	x'39	9

Figure 2 - EBCDIC / ASCII Conversion Table (from flounders.com)

Note that numbers 0 thru 9 in EBCDIC converted to Hexadecimal are represented as F0 – F9, which is easier to identify when compared to text data. For example, "2014" in EBCDIC Hexadecimal is represented as "F2F0F1F4". Note that you may be able to read the number if you remove all "F"s. On the other hand, "Bach" in EBCDIC Hexadecimal is represented as "C2C1C3C8". The following case study illustrates how translation from different character systems may lead to unknown repositories of cardholder data

Real Case Study: A large broker organization processes transactions in real time using mainframes. These transactions come as messages from business partners' systems. These partners are charged for each message and the details of these messages are important to solve disputes with their business partners' clients. While data was processed in the mainframe, it also flows to open systems. This requires transactional switches that convert data from EBCDIC to ASCII so the open systems can work with the data. During the interviews, the team responsible for the transactional switch informed

me that their transactional switch log contained all information received in Hexadecimal EBCDIC and ASCII format so data could be reconstructed if necessary. They also mentioned that they identified cardholder data in the logs previously, and that they removed it by replacing digits to the character X and leaving only the last four original digits. I asked for a sample log entry and I received something like this:

John Smith|XXXXXXXXXXXX1149|1015
d196889540e29489a3886af4f7f1f6f5f3f7f4f8f6f2f7f1f1f4f96af1f0f1f5

I asked for the sample because I was assuming that the team only truncated what they can easily see and left the hexadecimal EBCDIC representation untouched. My suspicion was confirmed after looking at the mapping tables between EBCDIC and ASCII and finding the mapping for numbers. The following table shows this relation clearly (Moldes, 2014):

Table 3 - Cardholder data in ASCII, Hexadecimal ASCI, and Hexadecimal EBCDIC

Format	Data String
Original ASCII data stream before truncation	John Smith|4716537486271149|1015
Hexadecimal ASCII	4a6f686e20536d6974687c34373136353337343836237313134397c31303135
Hexadecimal EBCDIC	d196889540e29489a3886af4f7f1f6f5f3f7f4f8f6f2f7f1f1f4f96af1f0f1f5

3.4.3. Internet Temporary Files

A browser creates temporary Internet files to store web site data for the web pages visited. When web page files are sent to the browser, they are stored so that they can be retrieved the next time that web page is visited. The next time the same web page is visited, the data is taken from the temporary file. This helps the browser display web pages faster. They are stored in the Temporary Internet Files directory and comprise Java scripts, style sheets, cookies, etc. (Oak, 2013)

Real Case Study: A processor developed an internal web-based application for the settlement and disputes department. This department generated Excel files containing transactions including full card numbers. The generated files were stored on a network share that has all required PCI DSS security controls including encryption. Technically,

cardholder data was only passing through the workstations used by the staff to their final destination. However, if the staff selected "open" instead of "save file" either intentionally or by mistake they were leaving a temporary version of the file on their local workstations. Even though they had been deemed compliant for a couple of years, previous assessors and security staff from this organization missed this unwanted repository of data. When workstations were reviewed, hundreds of files containing cardholder data were found. (Moldes, 2014)

3.5. Validating Card Numbers

Once card numbers are found, the assessor should be able to quickly validate whether the numbers are valid cards. Assessors can use the following techniques to validate whether what they found is a card number.

3.5.1. Card Number Structure

Card length can be between 13 and 16 digits. Card numbers are composed of several pieces of data. The first digit identifies the major industry and payment card brand. The following five digits identify the issuer bank, the remaining digits not including the last one are the account number, and the final digit is the check digit using the LUHN algorithm. (ComputerSolving.com, 2010)

Figure 3 - Card number structure for a Visa card (from Moldes (2014b))

Computer Solving.com has published an article describing this card number structure and other useful information to identify cards for the major card brands (ComputerSolving.com, 2010). The following table lists the identifiers for the card brands that formed the PCI SSC:

Table 4 - Card identifier for the Major Card Brands

Issuer	Identifier	Card Number Length
American Express	34, 37	15
Visa	4	13, 16
MasterCard	51-55	16
Discover	6011	16
JCB	3528-3529	16

3.5.2. Validate IINs (Issuer Identification Numbers)

Some numbers that are not actually card numbers may potentially pass the LUHN check. An additional test that can be conducted is to validate whether the IINs (previously known as BINs, Bank Identification Numbers) match any of the known IINs.

Ribbon.com has functionality that assessors may use to validate IINs. Just enter the first six digits into this webpage: http://bins.ribbon.co/.

Figure 4 - BIN validated using functionality provided by http://bins.ribbon.co/

3.6. Unusual Methods to Obfuscate Card Numbers

It is important to understand that developers usually find "creative" methods to obfuscate cardholder data, for example:

- Card digits shifted so the value stored is not the actual number;
- Card numbers obfuscated by breaking it into different pieces and storing it in separate fields or locations;
- Card numbers obfuscated by using encoding.

3.6.1. Digit Shifting

Digit shifting may make identifying card numbers more difficult. However, most of the time the first digits are not shifted. Assessors should pay attention to traces of the card number structure and especially to the card brand identifiers and IINs.

3.6.2. Breaking It into Pieces

Assessors should pay attention to column names, table descriptions, and field types to identify whether the card number has been broken into pieces. The most common pieces are the first six digits (major industry identifier and the bank identifier number) and the remaining digits (account number and check digit).

If the assessor finds fields containing the IINs, it is highly likely that the account number may be stored as well.

3.6.3. Encoding

Card numbers can also be stored encoded. While not in the clear, encoded data can be easily decoded. As a rule of thumb, all encoded data should be decoded to verify that card numbers are not only being encoded but encrypted as well.

For additional guidance on how to visually infer whether cardholder data is encrypted or encoded, refer to a paper I published entitled "How to Audit ASP.NET Applications for PCI DSS Compliance", Section 7.3 (Moldes, 2011).

4. Conclusion

While using automated discovery tools to find cardholder data may be ideal, an assessor' audit and discovery skills to perform manual discoveries could be highly valuable for an organization.

A quick review by a skilled assessor may help determine whether the organization should spend more resources reviewing their systems for cardholder data, and whether an automated tool would be needed to perform an exhaustive search.

In order to find cardholder data manually, assessors should maintain professional skepticism; and understand the organization's business processes, interviewees' roles and responsibilities, and the technology used by the organization.

Christian J. Moldes, Christian_moldes@hotmail.com

5. References

PCI SSC. (2013). "Requirements and Security Assessment Procedures". Retrieved September 3, 2014 from www.pcisecuritystandards.org website: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3.pdf, page 10.

Moldes, C. (2014), Case studies based on personal recollection of past assessments.

Jasuja, Sehgal, & Balram."GET vs. POST". Retrieved September 3, 2014 from www.diffen.com website: http://www.diffen.com/difference/GET_%28HTTP%29_vs_POST_%28HTTP%29

ComputerSolving.com. (2013). "What your credit card numbers mean". Retrieved September 3, 2014 from www.computersolving.com website: http://www.computersolving.com/computer-tips-tricks/what-your-credit-card-numbers-mean/

Bemer, B. "EBCDIC and the P-Bit". Retrieved September 3, 2014 from www.bobbemer.com website: http://www.bobbemer.com/P-BIT.HTM

Encyclopædia Britannica (2011). "EBCDIC (Extended Binary Coded Decimal Interchange Code)". Retrieved September 3, 2014 from www.britannica.com website: http://www.britannica.com/EBchecked/topic/198825/EBCDIC

Moldes, C. (2014b), "Card number structure" diagram

Moldes, C. (2011). "How to Audit ASP.NET Applications for PCI DSS Compliance". Retrieved September 3, 2014 from www.sans.org website: http://www.sans.org/reading-room/whitepapers/application/auditing-aspnet-applications-pci-dss-compliance-33869, pages 11-15.

Oak, M (2013), "What are Temporary Files ". Retrieved September 3, 2014 from www.buzzle.com website: http://www.buzzle.com/articles/what-are-temporary-files.html

6. Acknowledgments

Special thanks to the following members of the IBM security services practice who graciously offered themselves to proofread this paper and suggest additional content.

- Lance R. Mueller, Senior Incident Response Analyst
- Greg Tkaczyk, Senior Managing Consultant

I would also like to thank Richard Carbone who in his role as advisor suggested additional content for this paper.

Last Updated: October 21st, 2017

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS San Diego 2017	San Diego, CAUS	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Seattle 2017	Seattle, WAUS	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Gulf Region 2017	Dubai, AE	Nov 04, 2017 - Nov 16, 2017	Live Event
SANS Milan November 2017	Milan, IT	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Amsterdam 2017	Amsterdam, NL	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Miami 2017	Miami, FLUS	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Paris November 2017	Paris, FR	Nov 13, 2017 - Nov 18, 2017	Live Event
Pen Test Hackfest Summit & Training 2017	Bethesda, MDUS	Nov 13, 2017 - Nov 20, 2017	Live Event
SANS Sydney 2017	Sydney, AU	Nov 13, 2017 - Nov 25, 2017	Live Event
GridEx IV 2017	Online,	Nov 15, 2017 - Nov 16, 2017	Live Event
SANS San Francisco Winter 2017	San Francisco, CAUS	Nov 27, 2017 - Dec 02, 2017	Live Event
SANS London November 2017	London, GB	Nov 27, 2017 - Dec 02, 2017	Live Event
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZUS	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Khobar 2017	Khobar, SA	Dec 02, 2017 - Dec 07, 2017	Live Event
SANS Austin Winter 2017	Austin, TXUS	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Munich December 2017	Munich, DE	Dec 04, 2017 - Dec 09, 2017	Live Event
European Security Awareness Summit & Training 2017	London, GB	Dec 04, 2017 - Dec 07, 2017	Live Event
SANS Bangalore 2017	Bangalore, IN	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Frankfurt 2017	Frankfurt, DE	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DCUS	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS SEC460: Enterprise Threat Beta	San Diego, CAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Amsterdam January 2018	Amsterdam, NL	Jan 15, 2018 - Jan 20, 2018	Live Event
Northern VA Winter - Reston 2018	Reston, VAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SEC599: Defeat Advanced Adversaries	San Francisco, CAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Berlin 2017	OnlineDE	Oct 23, 2017 - Oct 28, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced