PCI Resources

Volume 3

Building a PCI DSS Information Security Program

Yves B. Desharnais, MBA, CISSP, PCIP

PCI Resources

PCI DSS Book Series (PCI DSS 3.1 edition)

by Yves B. Desharnais, MBA, CISSP, PCIP

Book Series Introduction

Welcome to this book series on PCI DSS. If you're reading this book, then you must have either an interest (in the field of PCI DSS compliance) or a need (your organization must become compliant, or currently has issues with PCI DSS compliance) to gain a better understanding of PCI DSS.

My goal is to provide a common understanding for business and technical people alike, and to provide a way for those people to communicate better about PCI DSS compliance, and information security in general. This is not a book for dummies. I believe that PCI DSS can be explained to laymen if properly presented. Some clients have even hinted that I'm pretty good at explaining it in a language that everyone can easily understand.

This book has been divided and broken up in 3 volumes that address the following ideas:

1. The Business Case for PCI DSS - What PCI DSS is and why it matters
2. PCI DSS Scoping - How scope is defined and documented
3. Building a PCI DSS Information Security Program - How organizations should approach the standard effectively and efficiently, and apply it to their in-scope environment (people, processes, and technology)

This book is for anyone who wants to better understand PCI DSS and its implications. I come from a strong technical background but I have also worked with many who do not. I tried to explain everything clearly without dumbing anything down while remaining true to my understanding of the standard. Some technical items are still present but will be highlighted accordingly so that the non-technical reader who wishes to do so can skip those sections (although I do hope that less technical readers might learn a few things from them should they look into these aspects).

My goal in describing PCI DSS is that a reasonable and knowledgeable person would arrive at a very similar conclusion to mine on most issues. While this book is published, it is by no means complete. The PCI SSC continues to release information based on new questions that come up and on changes in business and technology. Every such change will be documented on the associated website (www.pciresources.com) and I will issue reviews as warranted.

I believe the general approach and description in this book will stand the test of time. Links on the web however, since they are out of my control, may be more subject to change. For that reason, all links will be placed on the website for this book and updated as the standard evolves (including new information that I come accross). A PDF version of the references for printing will also be available from the website.

About the author

I started doing information security work in 2001, a time when there was limited resources still out there for those learning how get started in the field of information security.

At the time, there were mostly two ways of starting in information security. The first was through administrative studies, and focused on governance and policy. The second was network and system administrators involved in the technical aspects of the work. I came more from the latter side and my technical background was helpful in learning the ropes. My background was more related to application and system development, and I had decent system administrator skills, mostly self-taught, on Linux, Windows and OS/2. And I had also done work on two Unixes: Solaris during my undergrad years, and AIX for an internship.

The mid-90's undergrad course in computer engineering I took really prepared us well for what was to come. Through reading on a myriad of topics, practicing my craft, discussing with colleagues, I grew as a professional.

I was a QSA for a bit over a year while I lived in Chicago, and I now perform this work for organisations of all sizes, from the large and complex to the small and simple. I've helped many clients understand, scope and assess their PCI DSS compliance.

I wrote this book because, while there are many very good but disparate sources of information online (from the PCI SSC, blogs, etc. - see www.pciresources.com for a complete list of the sources I followed and used during the writing of this book), I have not found one document (physical or online) that presents things the way I think they should be presented. I felt a need to document my own thinking. The work I did for one PCI client led me to a deep reflexion on how I should present this information. This book is the result of this process.

This book is geared towards the business side of dealing with PCI DSS but also includes technical elements required for completeness (the PCI DSS has a more technical bent itself). Technical sections are identified as such and can be skipped by the non-technical reader. My hope is that having both technical and non-technical sections in one document will help both business and technical staff have the same vocabulary and understanding, thereby helping organisations reach (achieve) and sustain over time (maintain) their PCI DSS compliance.

Throughout this book I'll spell out PCI DSS to ensure no confusion exists with other PCI norms such as PA DSS and PCI PIN PTS. PA DSS will only be discussed briefly; PCI PIN PTS even less so.

Disclaimers

This book is the result of my experience and only represents my understanding, and is not endorsed by anyone other than myself, including previous or future employers, the PCI SSC or the card brands.

Mention of any product in the text should not be construed as an endorsement of any specific product, but only seen as examples, unless otherwise specifically mentioned.

What this book is and is not

This work is an interpretation of the standard based on my experience with it, various client experiences, conversations with peers and information security in general. I've read all that I could find on the subject including most documents from the PCI SSC and every Internet post I could find.

Please confirm with your assessor (QSA or otherwise) and document any interpretation you may use within your network. Your assessor, internal or otherwise, is the ultimate arbiter in the compliance world.

So, without further ado, let's dig in.

Book Series Acknowledgments

Writing acknowledgements is always a tricky thing since we always forget someone. So if you were deserving of such recognition and did not get it here, my bad.

I want to thank all of those of which I have learned over the years: family, friends, colleagues, teachers… all of which were teachers in some way, shape or form. Your support and constructive criticism have helped me learn and improve. I want to call out a few individuals who were instrumental in my professional career.

I would have probably not felt ready to write a book had it not been for the detailed and exhaustive review of the proposals, reports and other communications that Jan Hertzberg, my boss for over 2 years, took time to review, helping me improve my communication skills in the process.

The text you're reading also reads better thanks to the help of my brother Francois, a literary mind to my technical and business one. Francois performed the role of editor and made sure that this text was proper English.

Finally, I want to mention a few of the colleagues with whom I've discussed information security and PCI DSS over last few the years: Dan W., Jamison R., Josh B., Vanya O., Hakim A., and Tom B.

PCI Resources

Volume 3 - Building a PCI DSS Information Security Program

by Yves B. Desharnais, MBA, CISSP, PCIP

3.1 Volume Introduction

Welcome to volume 3 of this PCI book which leverages the work of volume 2 on how to determine what falls under PCI DSS scope. For details on how the standard came to be, who it applies to, and how and why you should care, please see volume 1.

This volume outlines an approach to compliance of all PCI DSS requirements using a standardized Information Security Program based on industry best practices, including the use of compensating controls when requirements cannot be met *“as stated"* .

The goal of information security should never be to block anything outright, but only to enable users to perform their legitimate business tasks in a secure fashion.

The goal of PCI DSS is to protect cardholder data from theft or unauthorized disclosure. This is our gold standard, the lens through which we will look at the PCI DSS requirements. And while the goal is to protect data, it is accomplished through measures on people, processes and technologies.

Note: Throughout this book/section, you will see me use many acronyms (including the already mentioned CHD, PAN, SAD). These are the most relevant ones for this section:

- CHD = Acronym for “Cardholder Data”; consists of the PAN, cardholder name, card expiration date, and sometimes service code
- PAN = Acronym for “Primary Account Number”; the card number printed on the front of the card.
- SAD = Acronym for “Sensitive Authentication Data”, it includes the magnetic track information, the PIN or PIN block, as well as the Card-not-present authorization value which we will refer to as CVV2 but can take any of the following acronyms: CAV2/CVC2/CVV2/CID.
- SPT = An acronym for “Store, Process, or Transmit”, meaning that a system or process comes into contact with CHD and/or SAD and is therefore automatically in scope.
- CDE = Acronym for “Cardholder Data Environment”, basically what we are trying to protect, which starts with the systems that SPT CHD or SAD but is not limited to these.

- Isolation = There is no possible access between systems.
- Controlled Access = There are limited (restricted) communications possible between systems.
- RoC = Report on Compliance
- Policy = a high-level document identifying the problem addressed by the document, the goals (or objectives), the position of the organization, and assigning responsibilities (technical detail is to be found in procedures) - this document must provide the ‘spirit’ (as in ‘spirit of the law’) that individuals will use to ensure that they are meeting the objectives of the organization
- Procedure = these are the ordered steps that are to be followed for any given process (e.g. some form of checklist) - when followed, procedures allow for consistent operations (consistent, not necessarily adequate, complete or optimized)
- Standard = a model that defines how (versus the procedures that address the ‘what’) things must be done - typically used for configuration standards (i.e. which IP range to use) and device hardening standards
- 'Untrusted' networks = networks not under the control of the organization, often also called “open, public networks” such as the internet
- Issuer identification number (IIN) = previously called the 'Bank Identification Number' (BIN), the full first six digits of the PAN that represent the financial institution
- Identification Number' (BIN) = See Issuer identification number (IIN)
- DMZ (demilitarized zone) = a buffer zone between the internet and the internal network of an organization
- Designated Entities Special Validation (DESV) = PCI DSS Designated Entities Supplemental Validation for PCI DSS 3.1 (DESV) - A new set of requirements to increase assurance that an organization maintains compliance with PCI DSS over time, and that non-compliance is detected by a continuous (if not automated) audit process; this set of requirements applies to entities designated by the card brands or acquirers that are at a high risk level for the industry

A full glossary is provided at the end of the book and on the companion website.

The “PCI DSS Scoping Model and Approach” presented in volume 2 (and published on the www.pciresources.com website) is also required, as I reference the different categories.

3.2 The High-Level PCI DSS requirements

The PCI DSS version 3.1 [1] standard released in April 2015 is used going forward in the volume.

Once our PCI DSS scope has been properly defined and hopefully reduced, the next step is to ensure that all 12 PCI DSS requirements and 200+ sub-requirements are met. Some of those requirements may be better met at either system, network or documentation level. I will describe the most appropriate scenarios when discussing each of those requirements.

Within the standard, the 12 PCI DSS high-level requirements are grouped into 6 different objectives that are not numbered. Most experienced professionals in PCI DSS refer to the 12 high-level requirements using a short-name (or description) that I have added to the following table:

	Objective / # Requirement	**Short Name**
	Build and Maintain a Secure Network	
1	Install and maintain a firewall configuration to protect cardholder data	*Firewall*
2	Do not use vendor-supplied defaults for system passwords and other security parameters	*Hardening*
	Protect Cardholder Data	
3	Protect stored cardholder data	*Storage*
4	Encrypt transmission of cardholder data across open, public networks	*Transmission*
	Maintain a Vulnerability Management Program	
5	Use and regularly update anti-virus software	*Antivirus*
6	Develop and maintain secure systems and applications	
	Implement Strong Access Control Measures	
7	Restrict access to cardholder data by business need-to-know	*Need to know*
8	Assign a unique ID to each person with computer access	*Authentication*
9	Restrict physical access to cardholder data	*Physical Security*
	Regularly Monitor and Test Networks	

10	Test and monitor all access to network resources and cardholder data	*Logging and Monitoring*
11	Regularly test security systems and processes	*Testing*
	Maintain an Information Security Policy	
12	Maintain a policy that addresses information security	*Policies*

Table 1 - PCI DSS High Level Overview [2]

The 12 high-level requirements organized into these 6 categories provides one approach to structuring an Information Security Program. While this method can work, I prefer a slightly more granular approach.

I will go through all PCI DSS 3.0 requirements in later sections grouping them along related themes, mostly following the high-level requirements, but with small ordering changes to categories and requirements as needed to present a more methodical approach. Those themes are how I would create a PCI DSS Information Security Program for an organization were there none in place.

3.3 Building a PCI DSS Information Security Program

3.3.1 Where you come from matters

I have worked with and within organizations big and small, and similar patterns often emerge in how they approach and manage security. Challenges differ depending on the organization type and size. Universities, for example, generally have a decentralized power structure, while big organizations are more top-down in their decision making processes.

If you look at any single information security individual, the path (experience) that brought him to his role has a tremendous impact on how he will initially approach security, although this is changing as information security training is more and more incorporated in college level programs.

Some get into information security from a policy governance side, often through a career or studies in administration or in management of information systems (MIS). It should be no surprise then that those individuals often start with crafting the governance structures, and then the policies. They choose fashion over form.

Others, myself included, come in with a more technical background. This used to be more someone who came up the networking ranks, or the system administrator route. My background was more in application development and system administration, and starting out, practical or technical controls were more my concern: form over fashion. In my first job as an Information Security Officer back in 2001, I taught myself what I needed to know and then configured the Cisco firewall (initially it provided no security), installed a proxy to manage internet traffic, wrote scripts to review (monitor) who went where on the web, etc. All these tasks were, at the time, more important than the policies.

Now neither approach is necessarily favored, as holistic security requires both the governance head and the procedural/technical body to achieve security.

3.3.2 Information Security Programs are meant to address Risks

With larger organizations or as smaller ones grow and more people get involved with information security, the need for greater/better structure becomes a necessity for coordination purposes. This is where policies and procedures become a means of aligning people with repeatable processes and a shared outcome.

The goal of an Information Security Program should be to protect information (addressing 'confidentiality'). We should also include the protection of system and infrastructure, which can extend to 'integrity' and 'availability', and include anything that can disrupt a business' activities (vandalism, disgruntled employees, etc).

The most critical thing for the success of an Information Security Program is what we generally refer to as the “tone at the top”. Basically, we need the backing and support of the top brass; they must be convinced that protecting this information is important, or we run the risk of having our Information Security Program that reads like a “check the box” type that does not sufficiently address risks. Or as noted leadership trainer John E. Jones said: *“What gets measured gets done, what gets measured and fed back gets done well, what gets rewarded gets repeated”* [3] .

The information security risks include what is often referred to as 'cyber security risks' (the technology aspect of information security), which I consider to be a subset of information security since 'cyber security risks' do not include the people and process areas of information security. But ultimately information security risks will be a subset of the risks faced by an organization, which generally include :

- Strategic – risks that would prevent an organization from accomplishing its objectives (meeting its goals).
- Financial – risks that could result in a negative financial impact for the organization (waste or loss of assets).
- Regulatory (compliance) – risks that could expose the organization to fines and penalties from a regulatory agency due to non-compliance with laws and regulations.
- Reputational – risks that could expose the organization to negative publicity.
- Operational – risks that could prevent the organization from operating in the most effective and efficient manner or that could be disruptive to other operations. [4]

Compliance with PCI DSS addresses a regulatory risk, but the controls it requires to be put in place help address many of the other risks faced by the organization as well.

Ultimately, information security is about managing risk. The PCI DSS standard is just more specific about mandatory minimal control requirements. Section 3.5.2 will cover my understanding of what PCI DSS requires in a risk assessment.

3.3.3 Information Security Frameworks

Most comprehensive information security frameworks should be broad enough to support the PCI DSS requirements, though some specific controls and concepts may need to be addressed in the implementation detail.

Several comprehensive frameworks and standards may be used as the basis of an Information Security Program, or to review its completeness. Some of the most common information security frameworks include:

- ISO/IEC 27001/2 [5] - The international standards has gone through many iterations and were initially derived from British standards derived themselves from UK public sector experience; these standards are often preferred by information security professionals and referenced in section 3.12.
- ITIL [6] (Information Technology Infrastructure Library) - is a set of practices for IT service management (ITSM) that focuses on aligning IT services with the needs of businesses (also derived from British government work); these standards are often preferred by IT professionals.
- COBIT [7] (Control Objectives for Information and Related Technology) - a framework created by ISACA for information technology (IT) management and IT governance; ISACA (Information Systems Audit and Control Association) is a nonprofit, independent association that advocates for professionals involved in information security, assurance, risk management and governance; these standards are often preferred by auditors (IT auditors, internal and external auditors).
- NIST 800 [8] series publications - a series of technical of publications from the NIST (National Institute of Standards and Technology) which are mandatory for most US federal institutions, and often referred to by HIPAA [9] SOX (Sarbanes Oxley [10] and other US based regulations.

In section 3.12, I will map the PCI DSS high-level requirements onto the ISO/IEC 27001/2 framework and discuss the differences between both. All individual controls provided by PCI DSS and other information security frameworks can be classified as:

- preventative
- detective
- corrective

Those three classifications are often referred to as the control triad, a term much used in all types of audits, including financial ones. This is to say that all requirements in PCI DSS (and any decent Information Security Program) will attempt to either:

1. prevent non-acceptable behavior (internal or external)

2. detect this non-acceptable behavior
3. and correct this non-acceptable behavior over time.

3.4 The PCI DSS Information Security Program Structure

The governance of the program, addressed next in this volume, will be key for us to achieve our goals of protecting information. An organization's structure can have drastic impact on the value assigned to protecting information versus other organizational goals. Whatever the reporting structure however, a clear distribution of tasks between the different people involved, internal and external, is required. We'll get back to governance and organizational structure in the next section.

While PCI DSS is divided in 12 high-level requirements, I prefer to start from this basic question, “what are we trying to protect?” and move forward from there (the same approach taken in volume 2 on defining PCI DSS scope). And while I will outline this for CHD and SAD, this approach should work with any type of data.

We start first by identifying the types of data, which form our data classification. PCI DSS does not call for a data classification outright since it has already defined what information requires special care (p.7). It does however implicitly allude to it in requirements over data retention and disposal (3.1) and media classification (9.6.*).

3.4.1 Recapping the PCI DSS data elements

The PCI DSS standard and its requirements cover two types of data.

The first type of data in scope is the Cardholder Data (CHD) and it is the one most often mentioned. It includes the Primary Account Number (PAN), which is the 15 or 16 digit payment card number (credit or debit [11]), the cardholder name, card expiration date and service code (a number rarely mentioned anywhere else). The last 3 elements (name, expiration date, service code) are only in scope if the ('complete') PAN is present (more on this in section 2.6.3 of volume 2).

The other set of data is called Sensitive Authentication Data (SAD) and consists of the information on the magnetic strip (also often called magnetic track), the card-not-present authorization code (3 or 4 digit code at the back of the card - except for American Express where it is on the front - and that can bear any of the following names or acronyms: CAV2/CVC2/CVV2/CID), and the PIN or PIN block (if present). SAD must be even more carefully protected than the PAN and other CHD, and that fact is often sadly forgotten (pun intended).

	Data Elements	**Storage Permitted**	**Protection Required**	**Render Unreadable**
Cardholder Data (CHD)	Primary Account Number (PAN)	Yes	Yes	Yes
	Cardholder Name	Yes	Yes	No

	Service Code	Yes	Yes	No
	Expiration date	Yes	Yes	No
Sensitive Authentication Data (SAD)	Full Magnetic Stripe Data	No	N/A	N/A
	CAV2/CVC2/CVV2/CID	No	N/A	N/A
	PIN / PIN Block	No	N/A	N/A

Table 2 - PCI DSS data [12]

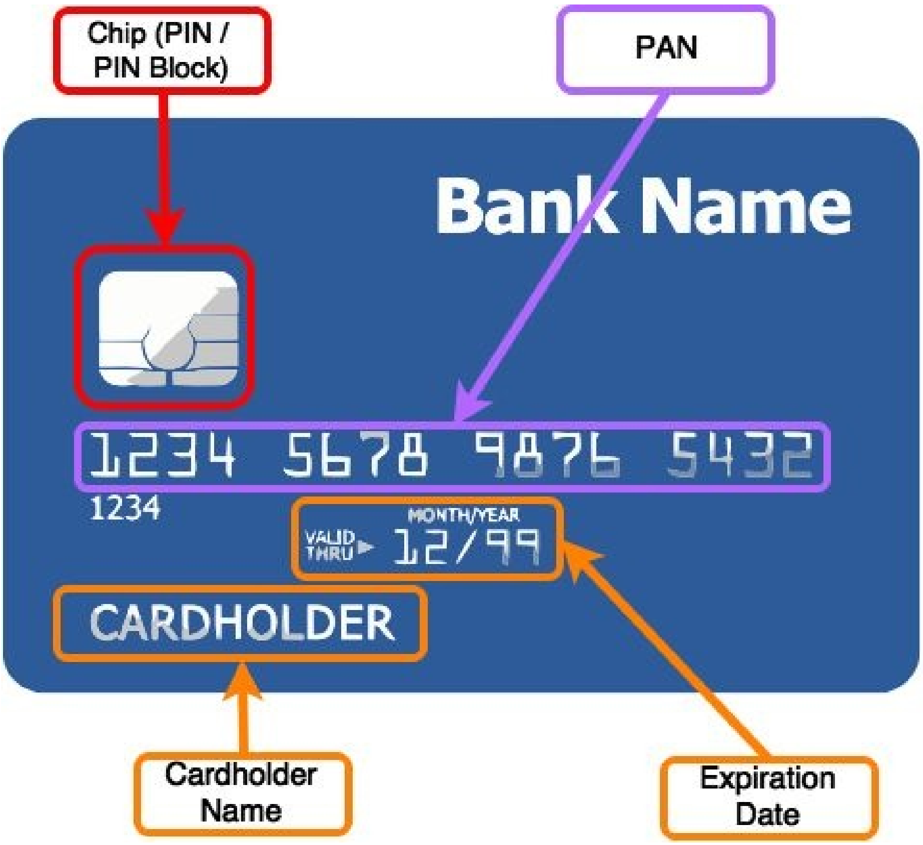


Figure 1 - Rendering of Credit Card (Front)

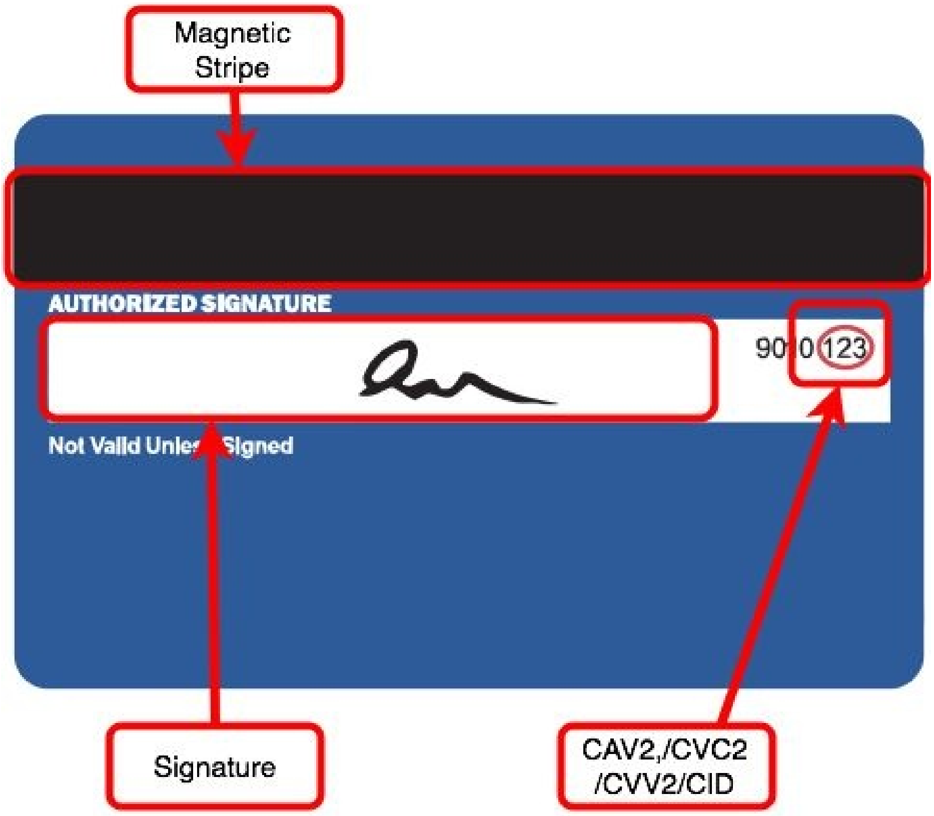


Figure 2 - Rendering of Credit Card (Back)

3.4.2 Data Classification

Most regulatory frameworks identify and classify information much like PCI DSS does. HIPAA [13] (Health Information Privacy Accountability Act) defines PHI (Patient Health Information) that must be protected. Many privacy laws (state-based in the USA, PIPEDA [14] in Canada, the European privacy directives [15]) define Personally identifiable information (PII), or Sensitive Personal Information (SPI) that must also be protected.

NIST Special Publication 800-122 [16] provides guidance on PII, and references a 2008 GAO (US Government Accountability Office) report to define PII as:

> *any information about an individual maintained by an agency, including (1) any information that can be used to distinguish or trace an individual's identity, such as name, social security number, date and place of birth, mother's maiden name, or biometric records; and (2) any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information.*

It then goes out to give multiple examples of what this data may include [17] :

- Name, such as full name, maiden name, mother's maiden name, or alias
- Personal identification number, such as social security number (SSN), passport number, driver's license number, taxpayer identification number, patient identification number, and financial account or credit card number
- Address information, such as street address or email address
- Asset information, such as Internet Protocol (IP) or Media Access Control (MAC)

address or other host-specific persistent static identifier that consistently links to a particular person or small, well-defined group of people

- Telephone numbers, including mobile, business, and personal numbers
- Personal characteristics, including photographic image (especially of face or other distinguishing characteristics), x-rays, fingerprints, or other biometric image or template data (e.g., retinal scan, voice signature, facial geometry)
- Information identifying personally owned property, such as a vehicle registration number or title number and related information
- Information about an individual that is linked or linkable to one of the above (e.g., date of birth, place of birth, race, religion, weight, activities, geographical indicators, employment information, medical information, education information, financial information).

Now the disclosure or theft of any information from the previous list does not always bring about the same level of risk, or impact to the affected individuals. For example, names, telephone and email addresses are generally considered less sensitive information than bank information or health information. Different types of data will thus have different levels of requirements for 'confidentiality', 'integrity' and 'availability' (often referred to as the C.I.A. triad of information security, with no relation to the 3-letter US agency that shares the same acronym).

And there are countless other examples of data that also requires protection in the myriads of regulated industries out there. Most organizations are subject not just to one, but to multiple of these regulations. It explains why most organizations develop a data classification that will be used to create policies and standards regarding the protection of information identified by these laws and regulations.

3.4.3 Examples of data classification

The number of categories and level of granularity found in data classifications is generally based on what is required by an organization. The adage to make things as simple as can be but never simpler (attributed by some to Albert Einstein [18]) is a good one to follow here.

Military data classification, portrayed in news, books and films, should be familiar to most people, and generally include categories such as 'Top Secret', 'Secret', 'Confidential', etc. [19]

It is very typical to see at least 3 major categories for all organizations: Classified or Restricted, Private and Public. Let's look at these basic ones in more detail.

'Restricted' is information, that if disclosed would cause significant harm to the organization through the risks identified in section 3.3.2. This category can include CHD and SAD (PCI data), Patient Health Information (PHI), more sensitive PII such as Social Security Numbers. It would also include trade secrets (think of the Coca Cola formula or proprietary source code).

'Private' is generally comprised of the internal work products that could have a limited negative impact on the organization if disclosed. This would generally include financial statements, client lists, and less sensitive PII data.

'Public' is information that is widely known and for which disclosure would have little impact on the organization. You may ask why we need a category for this type of information if it does not need protection. The reason is exactly so that you can inform people as to what not to focus energy on protecting (being that all organizations have limited resources, people and money). Already released financial information and press releases are all examples of publicly known information.

Often we'll see these top-line categories further divided; for example, we could have 'Restricted-PCI', 'Restricted-Health', 'Restricted-PII' (including bank data), 'Private-PII' (including emails, telephone) etc. This can allow an organization to define more granular controls that must be put in place for such a category.

To achieve PCI DSS compliance, we need to be able to match CHD and SAD to specific organization data classification categories (which could be simply a category called 'PCI data') all the way to the requirements mandated in policies (see section 3.5.3).

3.5 Governance

The Merriam-Webster dictionary defines governance as *"the way that a city, company, etc., is controlled by the people who run it"* [20] .

Any organization with limited resources (pretty much all of them) must make trade-offs to balance between different internal departmental goals (sales vs production). Thus, no matter what area we look at, be that information security, PCI compliance, sales vs production, etc., to whom responsibilities are mandated (what level is this person at in the organization) and what authority this person has demonstrates the value an organization places on that particular area. This is also the case in information security where the role and position of the ultimate person in charge makes a huge difference.

There are many ways that organizations can and have assigned information security responsibilities. Here are a few common ones, starting from the highest level of importance assigned by the organization:

- As a C-level executive responding to the CEO, often under the term CISO or CSO
- As a director/manager responding to a non-IT C-level executive (the CFO or Chief-Risk Officer (CRO), Compliance chief, etc.)
- As a director/manager responding to the CIO or IT director
- As a manager with limited authority within a convoluted IT department

Obviously the higher the person stands in the organization, then the more visibility senior management, and likely board members, will have into the information security posture. In much the same way, the level of authority given to that individual will be key to the approach taken by the organization to possibly integrate security within all processes (which PCI DSS included under the term Business-as-usual since version 3.0 [21]). The number of staff dedicated to security functions and their reporting structure is also telling of the importance assigned to this area.

Another item to consider is the department where that function is located. When the person responsible falls under the IT department, there can be some frictions with the rest of IT and Information Security. This kind of friction is inherent in any organization since different departments and roles have different responsibilities and are judged on different things. This is typical of sales (wanting to increase sales) vs production (trying to ensure they can actually produce what is sold) or purchasing, or even finance which may insist on certain levels of profit margins on products. This is normal, and as long as all perspectives are considered appropriately this should not be an issue. This type of friction explains why sometimes Information Security is placed with compliance or risk (not IT) as a 'check' (from checks and balances) to IT. This case can also address issues of separation of duties.

As a personal example, I have worked through conflicts with IT (telecom) in early portions of my career. The telecom engineer's goal was to provide connectivity (focus on availability) while mine was in protecting information through limiting accesses (focus on confidentiality). We both had the interest of the organization at heart, but also had different objectives. The role of our common boss was to be an arbiter when we could not compromise or resolve differences of opinions.

Sometimes the qualities and experience of the person in charge will have an impact on what level that person is placed at: the higher up, the more good communication skills are required (including explaining technical concepts to non-technical people without dumbing them down).

All of these possible role structures have pros and cons and should be considered based on needs, risk appetite, and skillset by organizations when they decide how to structure their organizations and where to assign responsibilities.

3.5.1 Responsibilities for the program

While PCI DSS compliance should not be addressed as an IT problem, it is still very technical (IT) in nature and many responsibilities will fall to technical staff. I generally recommend that one (non-IT) person be in charge of compliance with PCI DSS. If you have a chief compliance function, that would be a likely choice. If not, I would recommend looking at who has the relationship with the entity you need to report your compliance to. For merchants, this entity is your acquirer. For issuers, acquirers and service providers, reporting is made to the card brands (often multiple ones). In a merchant's case, that relationship is often held by the treasury department. So assigning the CFO, the treasury director or manager may work well. This individual does not need to be technically savvy, but would interact with individuals in charge of IT and Information Security (which depending on the organization can be one and the same) and serve as primary point of contact with the entity imposing compliance.

A very small committee may also be employed if assigning a single individual is not feasible, but I still recommend the task be given a single person if possible. Whatever the case, this relationship is better borne on the business than on the IT side. Remember PCI DSS is a legal, contractual and compliance requirement, not an IT one.

Requirements 12.5.* of PCI DSS mandate assigning information security responsibilities. We also recommend that these fall to a single individual, generally the CISO or CIO. Some of the responsibilities in the sub-requirements can then be delegated, but ultimate accountability should rest with the identified individual. Amongst the responsibilities are:

- developing and maintaining (updating at least annually) information security policies and procedures (12.5.1)
- ensuring monitoring of security alerts (12.5.2)
- implementing security incident response processes (12.5.3)
- administering user accounts (12.5.4), including controls over the addition and termination of users
- monitoring and controlling all access to data (cardholder) (12.5.5)

All of these responsibilities must be documented clearly and approved by management (12.4). Again, while these requirements cover cardholder data, they should still apply in reasonably the same way to all information held by the organization.

3.5.2 It's all about risk

The PCI DSS standard states it that it *“comprises a minimum set of requirements for protecting account data”* [22] and implies that it may not be sufficient to ensure security. This claim is the reason for requirement 12.2 to implement a risk assessment process to ensure that all risks are identified, assessed and addressed. The standard provides examples of risk-assessment methodologies:

- OCTAVE [23] : a methodology developed by the Software Engineering Institute (SEI) at Carnegie Mellon University (CMU) and used as part of the CERT Coordination Center (CERT-CC) division of CMU-SEI ('Computer Emergency Response Team', CERT)
- ISO/IEC 27005 [24] : a part of the ISO/IEC 27000 set of standards (including ISO/IEC 27002) that covers Information security risk management
- NIST SP 800-30 [25] : The Guide for Conducting Risk Assessments by the National Institute of Standards and Technology (NIST) aligns well with the other NIST 800 publications.

Still, any methodology that covers the following requirements should be adequate:

- Identifies critical assets, threats, and vulnerabilities
- Results in a formal, documented analysis of risk

Let's investigate those two requirements:

3.5.2.1 Risk Assessment: Identifies critical assets, threats, and vulnerabilities

How do we identify all assets? The simplest way is through a thorough scope definition as outlined in volume 2. This includes PCI data flow diagrams (1.1.3) and network diagrams (1.1.2), but also a complete inventory of all elements within the in-scope environment (2.4). The RoC reporting template [26] also provides us with more detail about the type of information that we must provide to ensure that everything has been identified. Section 3.6 covers scoping briefly, and volume 2 adds details on how to identify what assets are in scope and how to document this.

How do we identify threats? This is where many risk assessments fail in my humble opinion, and where further guidance from the council should be provided. The solution is the use of a discipline called 'Threat modeling', which the Open Web Application Security Project (OWASP) defines for applications (the same can be extended to IT systems and entire network environments) as:

> *Threat modeling is an approach for analyzing the security of an application. It is a structured approach that enables you to identify, quantify, and address the security risks associated with an application. Threat modeling is not an approach to reviewing code, but it does complement the security code review process.* [27]

OWASP even recommends Microsoft's approach to threat modeling [28] which Microsoft sees as a *“key activity in their Secure Development Lifecycle (SDL)”* [29] . Note that many other organizations provide guidance on threat modeling.

Threat modeling basically has information security professionals get in an attacker's mindset and try to uncover attack vectors, and then look at which controls (preventive, detective, corrective) are required to eliminate or mitigate risks to a level acceptable by an organization. This acceptable level of risk, called risk appetite, will vary with each organization but is generally influenced by the regulatory environment as well as other business factors.

How do we identify vulnerabilities? This is done through a vulnerability management process which is described in detail in section 3.7.11. This program will include most requirements of 11.*, but also tie back to requirements 6.1 (risk ranking) and 6.2 (patching). Although this type of testing has proven to be an issue for organizations (see section 1.9.2 of volume 1), it is well understood and described in section 3.7.11.

3.5.2.2 Risk Assessment: Results in a formal, documented analysis of risk

This simply means that everything needs to be documented so that an independent review (for example, your trusted QSA) can review the risk assessment that was performed.

3.5.3 (Information Security) Policies (Requirement 12)

The cornerstone of any Information Security Program is proper policies which lead to implementations of procedures and standards. This is why I'm presenting it early in this volume, to show its importance. Policies tell the organization what rules they need to follow. Note that policies, procedures and standards may be found under different names within different organizations. To align with the PCI DSS standard, we will use the same terminology. One blogger has outlined his own guidance [30] with which I agree. The following are short definitions that explain what each represents in the context of this book:

- Policy: a high-level document identifying the problem addressed by the document, the goals (or objectives), the position of the organization, and assigning responsibilities (technical detail is to be found in procedures) - this document must provide the 'spirit' (as in 'spirit of the law') that individuals will use to ensure that they are meeting the objectives of the organization
- Procedure: these are the ordered steps that are to be followed for any given process (e.g. some form of checklist) - when followed, procedures allow for consistent operations (consistent, not necessarily adequate, complete or optimized)
- Standard: a model that defines how (versus the procedures that address the 'what') things must be done - typically used for configuration standards (i.e. which IP range to use) and device hardening standards

Your policies may however be the last thing you address as it should reflect the current state of what you are actually doing as an organization. The order used is represented by the typical top-down vs bottom-up approach debate. Ultimately, as long as we arrive at policies, the process to get to them is irrelevant. And obviously, from a risk perspective, it is better to have a consistently followed approach (aka procedure or process) that meets the requirement and addresses the risk, so tackling that first may make more sense. However, you may, as you create or review your information policies and procedures, realize that you have forgotten something in your policies. This would be a good time to review them.

A review of policies is an area where the compliance or internal audit functions of your organization (which can be outsourced if you do not have such a role) can help perform a check function on your information security program.

Policies and their associated/derived procedures, while not as glamourous to IT professionals as the technical aspect of the work, are nonetheless critical elements. They help with personnel changes, from onboarding to people simply going on vacation (I like vacations and prefer this analogy to the *“hit-by-the-bus rule”* which is often mentioned to demonstrate the need for documentation in case an employee does not make it in one day), and they tell us what we should be looking for when assessing the organization.

We often see that issues identified are direct effects of breakdown in regularly (or not) performed processes. For example, when performing vulnerability scanning on client systems, I often found old vulnerabilities (2 or more year old) that would be addressed by existing patches; often, the affected system had not been properly decommissioned or was not covered by the organizational patch management process.

Since PCI DSS 3.0 and through 3.1, policies and procedures have been distributed amongst each of the 12 high-level requirements (they were previously all within 12.1.1). These specific requirements could still all be included in one or multiple documents, whatever the organization feels fits its needs best, as long as all requirements are covered. Many organizations have a PCI policy that they can update more frequently than other policies.

At a minimum, PCI DSS compliant Information Security Policies (12.1) and Procedures (P&P) should cover assigning responsibilities for :

- PCI compliance - an implied requirement of PCI DSS, but made mandatory in requirement DE.1.* [31] for designated entities [32] (and likely to be covered in future versions of PCI DSS)
- Information security (12.4, 12.5.*) - already covered in section 3.5.1
- Managing the firewall type devices (which can include routers and switches) (1.5) a requirement linked to the change control management process
- Managing vendor defaults and other security parameters (2.5) - also known as Hardening
- Change control management (6.4, 6.7) including testing and approvals
- Data classification (implied) and data retention (3.1, 3.7)
- Cryptographic key-management policy, processes and procedures (3.5, 3.6, 3.7, 4.3)
- Protecting the transmission of cardholder data (and likely other sensitive data) over networks not under the organization's control (4.3)
- Protecting systems against malware (5.4)
- Vulnerability identification (6.1, 6.7) from vendor sources
- Risk ranking of vulnerabilities (6.1, 6.7)
- Patch management (6.2, 6.7)
- Software Development Life Cycle (SDLC, 6.3, 6.7) including Secure Coding Guidelines and Training (6.5, 6.7)
- Access control, including the use of Role-Based Access Control (7.3)
- Identification and authentication of individual users (8.1.*, 8.2.*, 8.4, 8.5, 8.6, 8.7, 8.8) including user authentication policy for password changes
- Ensuring visitor identification and authorization (9.4.*, 9.10)
- Media (physical and electronic) classification (9.6.*) and management (9.7.*) including media storage (9.5.*) and destruction (9.8.*) (all within 9.10)
- Protecting payment card devices from tampering (9.9.*, 9.10)
- Logging and monitoring of relevant events (10.*, 10.8)
- Wireless network testing (11.1.*, 11.6)
- Vulnerability testing (11.2.*, 11.6) - aka performing vulnerability scans

- Network and application penetration testing (11.3.*, 11.6) including network segmentation testing (11.3.4) and corrections of identified vulnerabilities (11.3.3)
- Intrusion detection management (11.4, 11.6)
- Critical changes detection (11.5.*, 11.6)
- Performing risk assessment as required (12.2) - covered in section 3.5.2
- Developing and maintaining usage policies for critical technologies (12.3) that pose a high-risk, such as:
  - Remote access and wireless technologies (8.3)
  - Acceptable devices (12.3.3 / 4)
  - Mobile devices (laptops, tablets, phones) including BYOD if in-use
  - Removable electronic media, email usage and Internet usage.
  - Never sending unprotected PANs by end-user messaging technologies (4.2)
- Ensuring formal security awareness training (12.6.*)
- Personnel screening (HR) (12.7)
- Managing PCI Service Providers (12.8.*, 12.9)
- Incident response management (12.10.8)

These policies should be reviewed at least annually, updated when the environment changes (12.1.1) and approved by appropriate level staff in the organization. We will review the specific requirements that must be covered by the policies in section 3.7.

3.6 Documenting usage of card information

In order to define and validate scope, as well as assess compliance, we need to maintain basic information. I dedicated a complete volume (volume 2 in this series) to this very important topic and I recommend you review it prior to reading on, if needed.

I generally start with business process flows that show how people in an organization interact with cardholder data (while business process flows are not required by PCI DSS, I recommend that organizations maintain them nonetheless). This is the easiest way to work when initially interacting with non-technical personnel. Those processes often include hardcopy (i.e. paper) as well as electronic information.

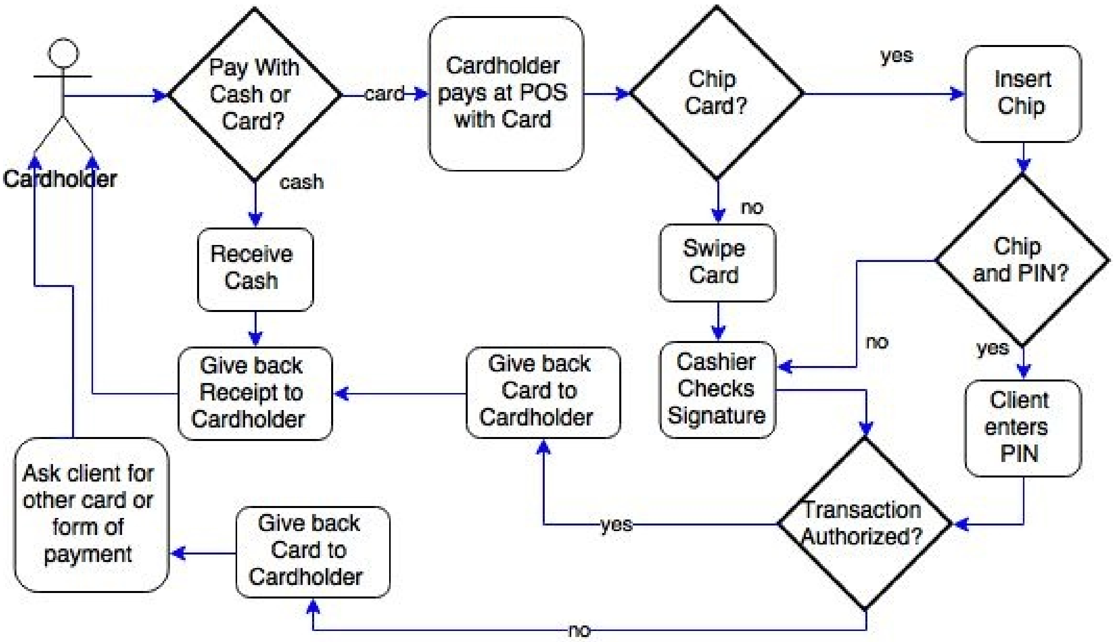


Figure 3 - Sample business process diagram

Once we have defined processes, we need to map these onto network diagrams into what is referred to as cardholder data flow processes across systems and networks (1.1.3). We obviously also need network diagrams (1.1.2) that provide sufficient levels of detail of what is in-scope.

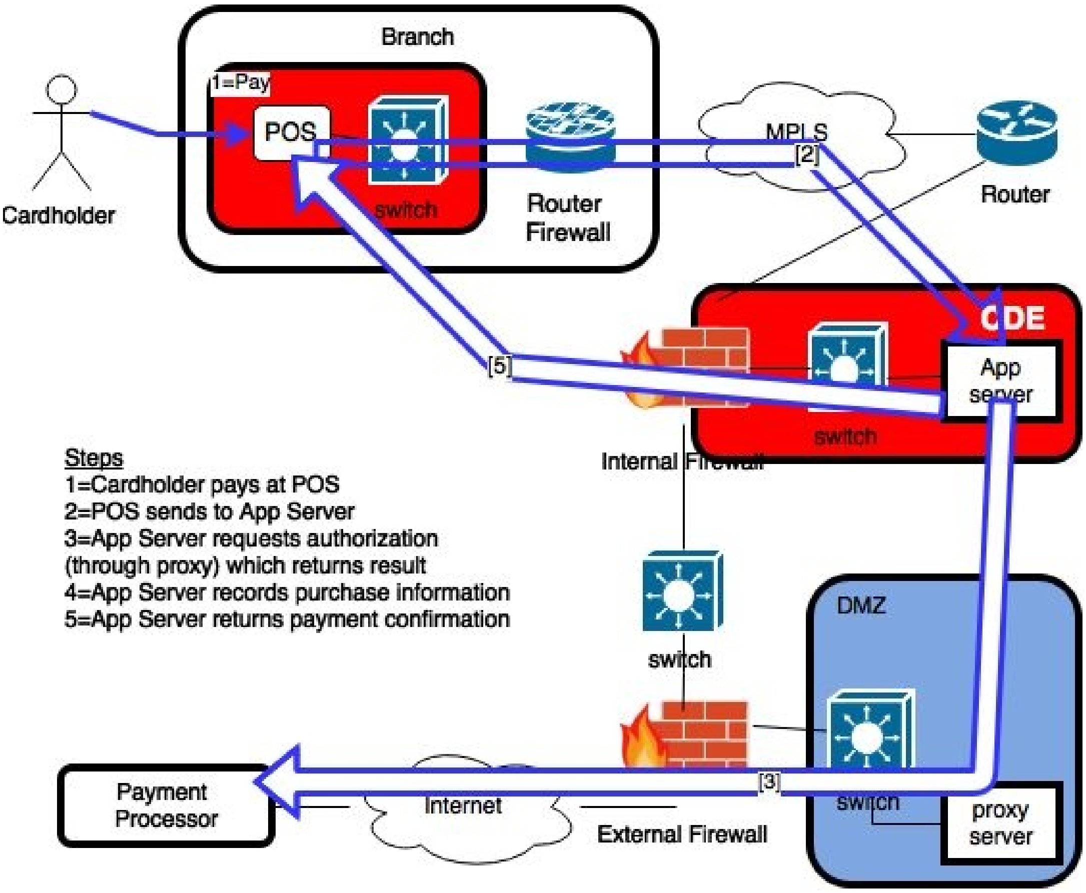


Figure 4 - Sample cardholder dataflow diagram

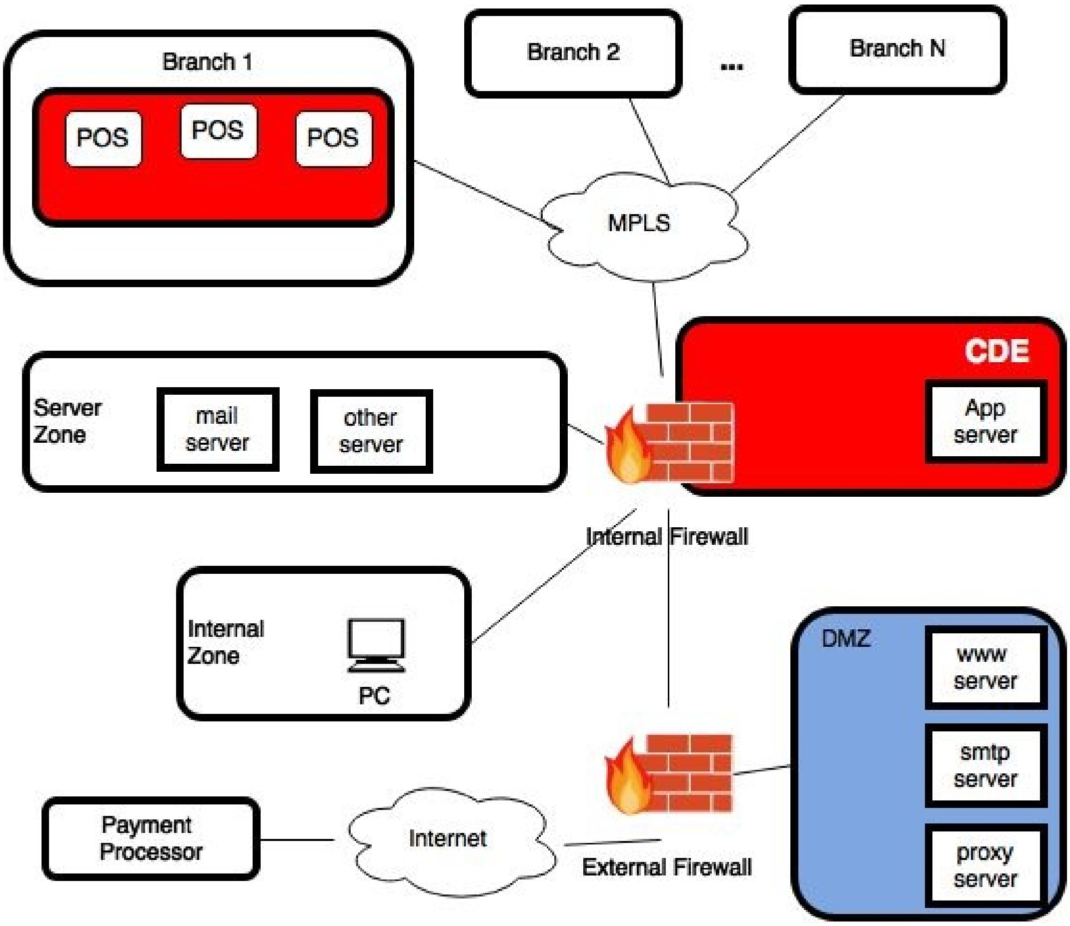


Figure 5 - Sample high-level network diagram (store chain)

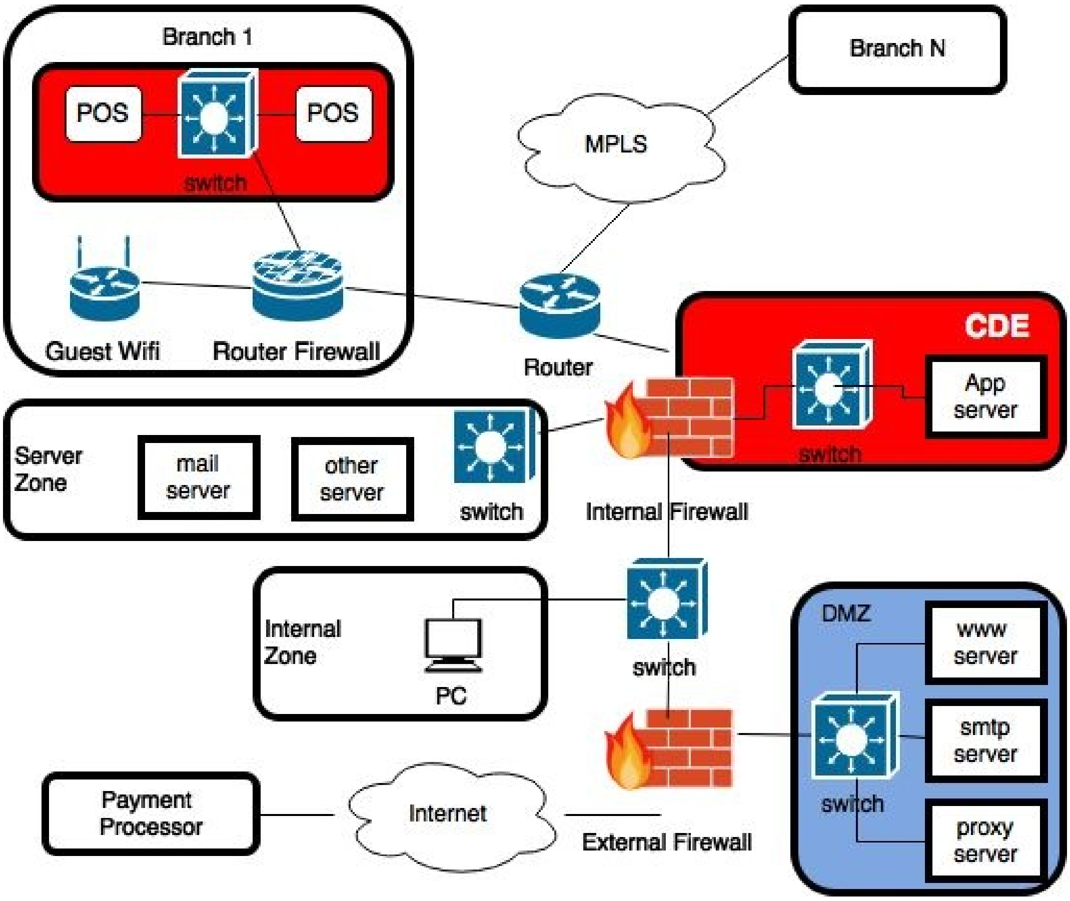


Figure 6 - Sample detailed network diagram (individual store)

The RoC reporting template [33] gives us the minimal information that must be produced and maintained. Please see section 2.5 of volume 2 for more detail.

These diagrams should be kept up to date as changes occur. One simple way to make this happen is to ensure that one of the items of the change management processes (for applications, systems and the network) includes the obligation to document the changes affecting PCI DSS, as one requirement for the change to be approved.

PCI DSS 3.0 introduced two new requirements calling for the maintenance of an inventory of all in-scope system components (2.4), and all wireless access-points (11.1) if any wireless networks are in use, regardless of if in-scope. Maintaining an inventory, often called 'asset management', is an area where many organizations fail. But asset management is a key control since you cannot protect what you do not know you have. This is why asset management is required by most regulatory, as well as all information security, frameworks.

The Designated Entity Special Validation (DESV) [34] requirements of June 2015 add further guidance about what an organization must do to 'Document and validate PCI DSS scope' in requirements DE.2.*. DE.2.1 formalizes the scoping requirement from p.10 of the standard. DE.2.2.* mandate that upon changes to the environment, and through the change control process, that documentation (network diagrams, cardholder data flows) and controls must be put in place, including performing necessary risk assessments for significant changes (please see section 3.7.11.2 for more detail on what PCI DSS considers significant changes).

3.7 - The body of the program

The policies and procedures implemented by the governance arm of the program must meet the PCI DSS requirements. These requirements are explained within the next sub-sections.

3.7.1 - Requirement 1 Firewall - Isolating the Cardholder Data Environment (CDE)

The firewall requirement comes first since the first technical layer of information security is generally at the network level, by preventing *“unauthorized access from untrusted networks”* [35] . Firewall functionality can be provided by multiple types of devices, from firewalls themselves to routers and switches, all of which can be physical devices or even virtual ones. The term 'network footprint' is used to define the limited set of protocols allowed in or out (both are important).

Once scope has been reduced, and systems consolidated in the smallest number of (network) areas, we must then work to protect these systems by initially isolating them fully from the rest of the network (also known as 'default deny-all', per requirement 1.2.1) and restricting traffic to only the systems and protocols/ports that are required for business (1.2.1). The business justification of all those protocols/ports open must be documented (1.1.6).

If the organization uses any insecure protocols, then countermeasures must be put in place to protect them and be documented as well (1.1.6). Insecure protocols may include, but are not limited to, FTP, Telnet, POP3, IMAP, and SNMP v1 and v2. Many of these allow the sending of credentials (e.g. usernames and passwords) in clear-text over an unencrypted connection that could allow a well-positioned attacker to intercept the traffic and gain access to these valuable credentials. A 'compensating control' for such a case (ftp, telnet, etc.) would be to run the network connection over a VPN tunnel.

The firewall rules employed should not be generic or apply to all systems. The only exception is where this simplifies implementation for generic services. For example, all CDE systems can send their log information (one-way only) to the centralized log collector system over the syslog port. I also recommend the use of name groups (instead of IP addresses and ranges) within rules to aid in reading the rules.

Here's one simple example of what this documentation could look like.

Source	Destination	Protocols	Action	Business Justification
any	CDE	any	deny	Deny everything not explicitly authorized
CDE	log_server	udp/syslog	allow	Allow CDE systems to send their logs to centralized server
IT_net	CDE_jump	tcp/ssh	allow	Allow IT network systems SSH to the jump server in the CDE

Table 3 - Example of business justification of firewall rules (requirement 1.1.6)

This documentation must tie in to the diagrams described in section 3.6 and 2.5 (of volume 2).

Remember that systems that are connected to 'CDE' systems to via open protocols/ports are 'connected' systems and considered in-scope.

The list of firewall rules must be reviewed at least every six months (1.1.7) to ensure that all rules are still required (which explains why we need to maintain documentation on those rules). Network diagrams (1.1.2) and PCI data flows (1.1.3) will generally also be involved during this review. For larger organizations, tools may be available that tie-in with your network devices (including firewalls) and allow you to meet the objectives of the documentation (1.1.6) and rules review (1.1.7) requirements.

We generally see at least a few different network segments within the network of a PCI compliant organization. At a minimum, we see an externally-facing demilitarized zone (DMZ) , the internal network, and an internal PCI zone (called the Cardholder Data Environment, CDE, in PCI terminology). A firewall must be present at each Internet connection and between any DMZ and the Internal network zone (1.1.4).

3.7.1.1 Internet-facing systems in the DMZ

Any internet-facing system should be placed within a special zone usually referred to as a DMZ (1.3.1). The term DMZ comes from the military; it defines a buffer zone between different nations or groups, famously still present between the Koreas (North and South). The DMZ is a less secure zone than the internal network since some of its services are exposed to external attackers (more on internal threats later). This zone generally has a small number of systems performing limited functions. The goal of this intermediate zone is to make an attacker's job more difficult by having them need to subvert a first set of systems with limited access to the internal network.

Only required protocols/ports should be open from the Internet to the DMZ (1.3.2) for both incoming and outgoing traffic (to make exfiltration harder should an attacker ever manage to gain access to this system). There should be no direct connection from the internet to the CDE (1.3.3).

Note that no CHD should ever be stored in the DMZ: it should all be in the CDE (1.3.7), the internal PCI zone. The CDE should not have DIRECT access to the Internet (1.3.5) nor should it be accessible from the internet (1.3). In fact, for security's sake, standard best practices mandates that most systems in an organization should never access the internet directly, but should go through filtering systems that may restrict access to undesirable sites (undesirable is to be defined by the organization) including filtering for malware or illegal sites. Any filtering system used by in-scope systems is likely contaminated by CHD. In some cases, the filtering solution will allow, through the use of a master certificate, to inspect all traffic that flows through it, looking for malware of even exfiltration of data. In such a case, the filtering system should be extremely well-protected and monitored since it will likely be considered a 'CDE/CHD' system.

An organization could have more than one DMZ if they wanted to split zones that come in contact with CHD from others that do not, although this is not a PCI DSS requirement. For example, they could implement a standard DMZ for smtp email gateway and web servers, and another DMZ for proxy systems, and one more for web-facing payment services/systems.

The three remaining 1.3.* requirements are interrelated and a bit more technical, so let me explain them along with some basic networking information. All three seek to protect the organization from Internet-based attacks.

I will use the IPv4 examples as they are simpler to understand than IPv6 (which is slowly replacing IPv4) but the same general concepts apply. A simplified networking primer is available in section 2.8 of volume 2.

Requirement 1.3.6 refers to *“stateful inspection, also known as 'dynamic packet filtering'”* [36] . This serves to protect against an attacker that tries to insert himself into a communication channel that was opened by someone else (say an application). The firewall maintains the 'state' of the connection to ensure this occurs. Most firewalls now meet this standard out-of-the-box. Validation would require the assessor to look at the manufacturer, make and model to confirm this.

Technical description: *This state validation generally occurs at level 3 (network) of the OSI model, usually in the IP (Internet Protocol) implementation of the firewall. Most firewalls perform some type of Network Address Translation (NAT) basically mapping between an external IP address and an Internal IP address. In section 1.9.5 of volume 1, I mention the recommendation of the Verizon 2015 PCI Compliance Report that 'stateful inspection' is not considered strong enough by many information security professionals, at least for external-facing firewalls. The recommendation is to use 'application-aware' firewalls which provide greater protection.* [37]

Requirement 1.3.4 (anti-spoofing measures to detect and block forged source IP) is related to 1.3.6 and generally automatically offered by most firewall devices. An attacker will often attempt to disguise himself as coming from somewhere else to bypass security defenses. We also see forged IP addresses during Denial of Service (DoS) attack. We should ensure that security degrades gracefully during a DoS attack as attackers have managed to hide their tracks during such attacks.

Requirement 1.3.8 asks us to never disclose private IP addresses and routing information to unauthorized parties. This is most often accomplished using Network Address Translation (NAT) and through the use of network ranges reserved for internal networks (and thus not routable over the general internet).

RFC 1918 has reserved 3 IPv4 ranges reserved for internal networks. 10.*.*.*, 192.168.*.*, and 172.16.*.* to 172.31.*.* (where * means a number from 0 to 255, or 8 bits).

3.7.1.2 Wireless

If any wireless networks are in use within the organization, then firewalls must be in-place between the wireless networks and the rest of the internal network (1.2.3). Wireless networks are at greater risk since an attacker need not be physically present onsite to access them. In fact, wireless access to networks using specialized antennas can be performed from far larger distances [38] . Only authorized users should be able to get access to the cardholder data environment from the wireless network. A safer approach (not mandatory, but something I would recommend) is to have wireless users perform standard remote access (e.g. VPN) into the network in order to access the CDE.

3.7.1.3 Firewall Configuration Standards

Again, on the subject of firewalls (and routers), we mean whichever device is used to provide firewall and network segmentation services, which we would categorize as 'CDE/segmenting' devices.

The organization should have defined a firewall and router standard (1.1) that provides a change process for any network change (1.1.1) including testing the change. This change process can be the generic one used within the organization (and covered in requirement 6.4), but if the required changes affect the PCI DSS scope then security (and compliance, if such a group exists) should have to review and approve the changes, so as to not risk reducing PCI compliance and increasing security risks unknowingly. The standard, which could be part of another policy (such as the information security policy), should include descriptions of groups, roles, and responsibilities for management of network components (1.1.5).

Using that standard, the organization should create firewall and router configurations that restrict connections between any in-scope zone and 'untrusted' networks (1.2). The default 'deny-all' should be in there (1.2.1). 'Untrusted' networks are those not controlled by the organization. The firewall and router configurations should be synchronized (1.2.2), meaning that changes made to them are actually saved and used when the device reboots. Network devices are notoriously not restarted very often and changes made to them can be in memory only (for testing purposes). Requirement 1.2.2 calls for securing and synchronising configurations. Many times, changes are made in memory on firewall devices but are only active as of the time they are entered. If they are not explicitly saved, then the changes may be lost during a reboot.

3.7.1.4 Changes to the CDE

Any change or extension / opening of the PCI network (the CDE) must ensure that security is not degraded. DESV requirement 2.2 mandates that those changes be approved, reviewed to ensure risk is managed, required controls are put in place, and that the relevant documentation be updated.

Such an extension is exactly what happens when a remote device (i.e. not on the organization network) accesses the CDE. Certain specific additional requirements, described below, apply.

3.7.1.5 Remote Access - Workstations, Desktops, Laptops

Requirement 1.4 mandates a personal firewall for mobile device (not in a fixed location) that may connect remotely to the network or to a network not controlled by the organization (1.4), also called an 'untrusted' network. This would include laptops, tablets, phones, etc., whether employee owned or organization provided.

The goal of this requirement is to protect such devices when they may be connected to a more hostile network environment not controlled by the organization, such as an cafe or airport (or even some home networks). In such networks, malware is often lurking, just waiting for targets to exploit.

It is a good general practice to mandate this on all individual devices, whether or not they are permanently on the network.

While as information security professionals we may debate the risk/reward of employee provided devices (commonly referred to as 'Bring Your Own Device' or BYOD), there is no doubt that this is a trend that is unlikely to recede, and thus it becomes our obligation to protect the organization's information everywhere it is held. A personal firewall is likely not sufficient to protect from threats in such cases, and I would strongly advise looking at security solutions for all mobile devices (employee and organization provided). Those solutions are often categorized as 'Mobile Device Management' (MDM) solutions.

Two-factor authentication (8.3) is included in requirement 8 (authentication), but it makes sense to tie it to requirement 1. Any remote access (user, administrator, vendor, etc.) that can interact with the CDE in some way, shape or form, can be seen as 'breaching' the CDE 'bubble' (or isolation). This added level in risk is compensated by that second factor which means that two of the following must be used to confirm the user's identity:

- *Something you know, such as a password or passphrase*
- *Something you have, such as a token device or smart card*
- *Something you are, such as a biometric.* [39]

Remember that you must use 2 different categories, as two of the same category (say a password and a PIN) are still considered a single factor (used twice).

The goal of authentication is to tie every action back to an individual user; any factor (password, token, certificate, etc.) must be tied to an individual and CANNOT be shared between multiple users (8.6)

3.7.2 - Requirement 2 - Hardening

Hardening seems to be new to many organizations but is a basic building block of any Information Security Program. It basically means building default secure configurations for all devices at the offset. This is the systems equivalent to the 'deny all' rule of firewalls, and requires only allowing functions strictly required for business operations. It includes disabling (or removing) all default settings and accounts (2.1) and in-addition for wireless networks, changing network passwords, keys and SNMP strings (2.1.1). SNMP, or Simple Network Management Protocol, is a protocol that may return configuration and status of network devices. Once again, this is more dangerous in a wireless environment where an attacker does not need to be physically present.

The way to ensure that all of these default settings are changed is to develop secure configuration standards (2.2). Industry-accepted standards have been developed by a number of organizations, including but not limited to:

- Center for Internet Security (CIS) [40]
- International Organization for Standardization (ISO) [41]
- SysAdmin Audit Network Security (SANS) Institute [42]
- National Institute of Standards Technology (NIST) [43]

The standard an organization builds should be based on a trusted industry-accepted standard, or at least validated against them. For many of my clients, I've recommended that they use one of those source organizations, adopt the standards as-is, and document the differences with the external standard and what reasons justify the deviation. That way, it simplifies the maintenance of those standards over time. For example, an Active Directory standard may call for not reusing 24 last passwords, but due to a specific constraint the organization cannot support more than 12 (which still exceeds requirement 8.2.5). Documentation could look like this:

> Configuration Standard: Microsoft Windows Server 2012
>
> SOURCE: CIS Microsoft Windows Server 2012
>
> Exceptions:

Source Section	Recommended Value	Implemented Value	Deviation Justification
1.1.1.5	24	12	System X limitation only supports 12. PCI DSS calls for 4, 12 exceeds this.

The hardening standards include having only one primary function per server (2.2.1). The role can be DNS, file storage, database, application, web front end, etc. Note that an individual virtual machine (VM) is seen as one server. The hypervisor running the VM is seen as a server having the hypervisor role. [44]

Only necessary services should be enabled (2.2.2) which means that for insecure services, additional security features must be implemented to address the unsecured portion (2.2.3). For example, telnet and ftp send authentication credential (username and password) through clear-text. Running those services over a VPN connection would address the clear-text credential issue. Requirement 2.2.4 mandates setting secure default values for all configuration setting. The organization must also remove all unnecessary functionalities (2.2.5). Often, an attacker can use those default (and vulnerable) scripts to nefarious results. One such attack I have myself performed as part of a penetration test is gaining shell (command line) access on a database server using default scripts that came with the software. Had those files been removed, I would have had a harder time gaining access to the machine.

The hardening standards should be reviewed at least annually, and changes should be applied retroactively to all systems currently in production.

Finally, any administrative non-console (i.e. not physical, and by console, we mean the physical console of a system, often found in the data center) access must be encrypted (2.3). This type of access is generally done through protocols such as Remote Desktop (RDP), ICA or VNC. Encryption is explained in the crypto primer found in section 3.13.

So for all types of systems (windows web servers, Linux DNS, etc.) we should include (in one or multiple documents):

- a configuration standard
- a build or installation guide (to meet the requirement)

Often the build guide implies that the organization will create a base hardened operating system (windows, linux, etc.) image that must be used for all new system implementations.

An assessor will review the configuration standards versus the build guide, and industry best practices or manufacturer recommendations, and then sample in-scope systems to see if the proper configuration was applied.

3.7.3 - Requirement 3 - Storage of Cardholder Data

As mentioned in section 3.5.3 (policies) earlier, PCI mandates data retention and disposal policies and procedures (3.1). The retention policy is often included within the data classification policy, or at least references it. The(se) policy(ies) should define time limits for retention with proper justification (mostly laws and regulations), and specifically cover retention requirements for cardholder data. They must also include a process which could be manual or automated, that runs at least quarterly (every 3 months) to identify and delete cardholder data that has passed retention time. Remember that cardholder data can exist in many places including files, databases, and logs (it shouldn't be logged but it could be, etc.). DESV requirements 2.5.* [45] asks us to implement a *“data-discovery methodology to confirm PCI DSS scope and to locate all sources and locations of clear- text PAN at least* *quarterly”* [46] (the same frequency as destruction of data) . This discovery could be performed manually, but is better performed using specialized tools that can look for patterns of PAN, including Data Loss Prevention (DLP) or data identification tools. Section 2.5.5. covers this in more detail.

Requirement 3.2 and its sub-requirements cover SAD after authorization. The only entity that can keep SAD is the issuer for its own cards, and those must be adequately protected. SAD pre-authorization data (see section 2.3 and FAQ 1154 [47]) can be kept, but it should be encrypted securely as defined for the PAN in requirement 3.4. SAD includes track data from the magnetic stripe (3.2.1), card verification codes or values (three-digit or four-digit number printed on the front or back of a payment card) (3.2.2), the PIN or PIN-block (3.2.3).

The next two requirements cover presentation and storage of the PAN. When displaying the PAN, unless you absolutely need the full number (and can justify this as a documented business need), it should be masked (3.3) displaying, at a maximum, the first 6 digits and last 4 digits (something like 4444 44** **** 1234). Look at any payment receipt and you'll see that it only shows the last 4, often so that you can identify within your multiple cards, which one you used. Remember that if you capture a screen containing a full PAN, then it must be stored following requirement 3.4.

The PAN's format is 16 digits (15 for AMEX) like the following: 4012 8888 8888 1881

The first digit identifies the brand; generally, 4=Visa, 5=MasterCard, 6=Discover, 3=JCB or AMEX.

The full first six digits represent the financial institution, and are called 'Issuer identification number (IIN)' which was previously called the 'Bank Identification Number' (BIN).

The last digit is a calculation to validate that the full number is valid, a 'check digit' using an algorithm called LUHN.

Requirement 3.4 requires that we *“render the PAN unreadable”*, or not valuable to the attacker. Four ways are identified as acceptable:

- One-way hashes (see section 3.13.3.3 in the Encryption Primer) - an option that I think should no longer be used as it can likely be brute-forced.
- Truncation (a screen capture of a masked PAN per 3.3 becomes truncated) - truncation is non-reversible.
- Index tokens (tokenisation) - where tokens with unpredictable values replace the PAN - see section 2.6.3.2 of volume 2 for more detail on tokenisation.
- Strong cryptography - encryption requirements described further in section 3.7.3.1 “Encryption of Stored Data” below.

In all cases, for the method to be acceptable, it must be impossible for an attacker to reconstitute the PAN.

3.7.3.1 - Encryption of Stored Data

Encryption is a complex process; and it is very easy to make mistakes during its implementation. I provide an high-level version of how encryption works in section 3.13. If you do not have a good understanding of encryption, I suggest you review it before reading this section.

If stored cardholder data (PAN or SAD) is to be stored encrypted, then multiple requirements must be met. Encryption could be performed at the database table or field level (recommended), file level, or the media can also be fully encrypted, as is the case with some hard drive or tape backup encryption. In the case of encryption performed at the media level (disk or tape), keys must not be associated or managed by the operating system authentication (which would make breaking that layer a single point of failure) but must be managed separately and independently from the local authentication. This means that the user would likely have to enter a passphrase (a type of key-encrypting key) manually after logon to get access to the encryption media. See requirement 3.5.2 below for more detail.

Encryption is only as secure as the protection given to its encryption keys, which is why requirement 3.5 mandates developing (and documenting) procedures to protect encryption keys by, amongst other things, restricting access to the smallest number of key custodians necessary (3.5.1). Although that exact number of custodians is not defined, it should be kept to a minimum. The keys must also be stored in the fewest possible locations (3.5.3).

The other 3.5.* requirement requires a bit more technical knowledge, which is covered in the encryption primer of section 3.13. Data may be encrypted with either a symmetric cipher such as AES, using a shared-key, or using an asymmetric cipher such as PGP, using a public and a private-key pair (for details see encryption primer). For symmetric ciphers the shared-key must be protected, while for asymmetric ciphers the private-key (which is used to decrypt) must be protected while the public key may be known. Whichever key must be protected, requirement 3.5.2 mandates that secret and private keys be stored in one of the following ways:

- *Encrypted with a key-encrypting key that is at least as strong as the data-encrypting key, and that is stored separately from the data-encrypting key*
- *Within a secure cryptographic device (such as a hardware (host) security module (HSM) or PTS- approved point-of-interaction device)*
- *Or as at least two full-length key components or key shares, in accordance with an industry-accepted method* [48]

A key-encrypting key is just another encryption cipher. Those key-encrypting keys do not need to be encrypted, just stored securely. A key sharing system used to share these is also often called a key distribution system.

Requirement 3.6 mandates development and documentation of all the relevant key-management processes and procedures for cryptographic keys used for encryption of cardholder data. The processes and their documentation must cover, at the very least, generation of strong cryptographic keys (3.6.1), secure cryptographic key distribution (3.6.2), secure cryptographic key storage (3.6.3), prevention of unauthorized substitution of cryptographic keys (3.6.7), and a requirement for cryptographic key custodians to formally acknowledge that they understand and accept their key-custodian responsibilities (3.6.8). No individual user should have access to clear-text versions of the keys, and in such cases, operations must be managed using split knowledge and dual control (3.6.7), meaning that the key is split between two or more individuals.

Organizations must also provide for the *“retirement or replacement (for example, archiving, destruction, and/or revocation) of keys as deemed necessary when the integrity of the key has been weakened”* or *” keys are suspected of being compromised”* (3.6.5), which also includes defining a 'cryptoperiod' for each cipher and key *“(for example, after a defined period of time has passed and/or after a certain amount of cipher-text has been produced by a given key)”* that will force retirement or replacement (3.6.4).

3.7.4 - Requirement 4 - Transmission of Cardholder Data

Requirement 4.1 mandates that we must use strong encryption (see section 3.13 for the encryption primer) when transmitting cardholder data over open, public networks ('untrusted' networks). Depending on the encryption protocol and algorithm (cipher), some variations of requirements covered in section 3.7.3.1 will need to be in place (see section 3.7.3.1 and the encryption primer). Some of the main requirements are to validate trusted keys and certificates. The protocol version and encryption strength must also be secure. For example all versions of SSL and SSH version 1.0 are no longer considered secure, but more recent versions of those protocols are usable (e.g. TLS 1.1 and later, SSH 2.0). And 40-bit key lengths are definitely no longer considered sufficient. The primer describes the current understanding on the most secure protocols but will defer to the NIST standards, as PCI DSS does, for acceptable strong ciphers.

In version 3.1 of PCI DSS, some protocols were depreciated and organizations still using those need to move to newer secure ones by July 1, 2016 at the latest.

Since open public networks are outside the control of the organization and a well-placed attacker may be able to intercept and eavesdrop on the communication, we need to secure the communications on networks where we have no control. This can be done by using encrypted communication channels such as VPN (site-to-site or point-to-point) or using dedicated private links.

Open, public networks include, but are not limited to the Internet, wireless networks, and bluetooth connections. For Multiprotocol Label Switching (MPLS) networks, which are often used to provide connectivity between various physical sites (data centers, branches, etc.), the details of the implementation determine whether the network is considered public or private. FAQ #1045 [49] addresses this issue when responding to the question: *“Is MPLS considered a private or public network when transmitting cardholder data?”* It basically asks us if there is any connection to (or entry point from) the open internet:

> *If the MPLS network contains publically-accessible IP addresses or otherwise provides exposure to the Internet (for example, if an edge router has an Internet port), then it may need to be considered an “untrusted” or a public network.*

So, if there is a connection to the internet, then it will be considered an 'open, public network' whereas if no internet connection exists, then it will be considered a 'closed network'.

Wireless networks are at greater risk since an attacker need not be physically present onsite to access them, and must therefore also use strong encryption (4.1.1), which generally means using WPA/WPA2 protocols (also see section 3.7.1.2). The primer goes into more detail on the industry best practices for secure protocols for wireless networks.

The PAN (and SAD) should also never be sent through email, instant messaging, chats and other applications of that nature (4.2). This mandate needs to be placed in a policy somewhere (a logical place is the usage policy described in 12.3.*, but could be any other that is viewed by all users of the organization). DESV DE.2.6.* [50] adds the obligation *"implement mechanisms for detecting and preventing clear-text PAN from leaving the CDE via an unauthorized channel, method, or process, including generation of audit logs and alerts"*, which means this has to go above the policy level to add a technical detective control through some sort of filtering system, which may include Data Loss Prevention (DLP) solutions.

3.7.5 - Requirement 5 - Antivirus / Antimalware

Malicious software, or malware, includes but is not limited to viruses, worms, trojans and rootkits. Malicious software has been with us for almost as long as computers have, but their effect has been compounded in a fully networked world (aka the Internet). The Morris worm in 1988, was the first worm identified. And malware has evolved to be not only generic, but at times more targeted at specific organizations with *"70-90% of malware samples are unique to an organization"* (variations in virus families) [51] .

Still, the question of whether there really is a need for an antivirus (more anti-malware nowadays) keeps popping up. The Verizon 2015 PCI compliance report confirms that some form of malware is used in the first steps of most successful attacks.

Thus any system vulnerable to malicious software, or malware, needs to have protective software installed (5.1). The software selected should be one recognized by the industry as effective in being able to remove all known malware (5.1.1). These should be centrally managed and end-users should not be able to disable them as a general rule (5.3) (only when a technical reason requires it, authorized by management and then only for as short a period of time as necessary). The software must be kept current (updated so it can detect new malware), scan the systems periodically and generate logs (5.2). Any and all logs generated by the software need to be collected and monitored alongside all other organizational logs as demanded by requirement 10.7 (see section 3.7.10). As an alternative to anti-malware, application whitelisting solutions, which allows only vetted applications (generally because they are signed using cryptographic keys) to run, thus preventing (unsigned) malware from running. Application whitelisting may also be less resource intensive on systems. As always, no technology is a perfect solution, which is why we have to maintain multiple layers of controls.

The one part of requirement 5 that may get a different interpretation is of *"commonly affected by malicious software"* (5.1.2). Generally, this has been taken to mean any end-user general purpose operating system such as all versions of Microsoft Windows, Apple's

Mac OSX and some desktop usage of Linux. Windows is the one most people think about as it has been targeted more than others since it represents the standard in the business world. Whatever definition you decide to use internally, a new requirement introduced in PCI DSS 3.0 requires that the organization re-evaluate periodically (at least annually) whether these excluded systems warrant the use of anti-malware software (5.1.2) (Windows cannot be considered not affected). For example, an organization that uses Linux or OSX as a desktop may (not necessarily should) consider these to not be commonly affected by malware. It would still need to review whether that claim stands up. Remember that humans (who will use these computers) are often the weakest link in the security chain. The one exception everyone generally agrees about is not requiring antivirus on mainframe and midrange similar types of systems: IBM Z series, IBM P series (AIX), IBM I series (OS/400), HP Non Stop (Guardian), HPUX, etc.

3.7.6 - Requirement 6 - Vulnerabilities, Patching, Change Control and Software and Web Development

As the section title implies, requirement 6 is a hodgepodge of different but related requirements for securing systems and applications.

3.7.6.1 Vulnerability Management

The first portion of requirement 6.1 mandates the creation of a process to identify vulnerabilities using reputable outside sources. For organizations that use mostly components (hardware and software) from very few vendors, this may mean subscribing to mailing lists, newsgroups or RSS feeds (for example, from vendors such Microsoft, Cisco, etc.). The second portion of requirement 6.1 is assigning a risk ranking to all vulnerabilities (from a vendor list, external or internal testing). The latter part of this requirement will also apply to vulnerabilities identified in internally developed software (see 6.3, described shortly). The risk ranking methodology should be based on industry standards and must include a risk level consisting of, at a minimum, high, medium and low rankings. Any vulnerability with a risk level of 'high' or above, should be remediated within one month. We often see risk ranking methodologies using two distinct axes: impact (what could happen if someone exploited this vulnerability) and probability (likelihood). Unless you are an experienced risk professional, please do not create your own methodology but adopt an existing one. There are many industry standards, from CVSS (Common Vulnerability Scoring System, a free and open industry standard for assessing the severity of computer system security vulnerabilities in use since 2004- CVSS is also mandated for external vulnerability scans of requirement 11.2.2), to those provided by SANS, OWASP (Risk Rating [52]) and others. Whichever methodology you use, make sure that your methodology is properly documented and used consistently.

For example, the OWASP Risk Rating methodology uses underlying factors for both likelihood and impact to create an overall risk rating, as described in the table below.

		Overall Risk Severity		
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium

	LOW	MEDIUM	HIGH
	Likelihood		

Table 4 - OWASP Overall Risk Severity Rating from Likelihood and Impact Factors

The methodology considers 8 likelihood factors and 8 impact factors.

Likelihood is further divided into threat agents (related to the attacker) and vulnerability (regarding the vulnerability). Threat agents (attack) factors include skill level, motive, opportunity and size (of group of potential attackers). Vulnerability factors include ease of discovery, ease of exploitation, awareness, and intrusion detection (whether we are well equipped to detect, log and react to an exploitation attempt).

Impact is further divided into technical and business impact. Technical impact factors include loss of confidentiality, loss of integrity, loss of availability, and loss of accountability. Business impact include financial damage (effect on revenue and profit), reputation damage (often to the 'goodwill' effect on the balance sheet), non-compliance (which can have a financial impact), and privacy violation.

Now that you have identified vulnerabilities, you need to address (or remediate) them. Requirement 6.2 (often called *'patching'*) mandates installation of critical patches within one month. Critical patches are those with a high probability of exploitation on a vulnerable system, and high impact if exploited. Patching is an area where I've seen many organizations struggle, especially with a 30 day (month) timeline. This is why a good risk ranking methodology is so important to ensure that adequate patching can be performed in a timely manner. Other less risky vulnerabilities also need to be addressed, but the timeline for these is left to the organization. I've often used 3 months for high vulnerabilities, 6 months for medium vulnerabilities and a year for low vulnerabilities, but each organization needs to make that determination for themselves based on their risk appetite.

3.7.6.2 Change control

Any change to any component must go through a formal change process (6.4) regardless of whether the change is a patch (6.2) identified during the vulnerability identification process (6.1), a configuration or software change, or the testing of systems (11.*). The change control process serves as a check against both insider threats as well as the law of unintended consequences. The law of unintended consequences, which can be compared to the adage that “the road to hell is paved with good intentions”, is also augmented by the complexity of systems and applications. Test environments must be different from production ones (6.4.1) and include separation of duties (6.4.2) so that, for example, a developer cannot put code into a production environment (an independent system administrator will generally put this new code into production). In smaller organizations where such separation of duty is not possible due to limited staff, compensating controls should be put in place (see section 3.10). For change control management, compensating controls that could be used might include automatic logging of all changes performed and a review (matching file changes to change control requests) by other departments of the organization (even by non-technical staff). In no case should production data which contains CHD and full PANs be used in testing or development environments (6.4.3).

Some test numbers are generally available from payment processors or acquirers. Visa Europe also provided 2 BIN's series reserved for internal use [53] , much in the same way that RFC 1918 provides internal IP ranges.

The organization can choose to use random test data or implement a process to sanitize CHD from live data before its use in other environments. Before code is put in production, any test accounts and data must be removed (6.4.4). Hardcoding values within code should never ever be done (for PCI DSS or any other environments). You should always use configuration files (or the registry) both of which are easier to modify.

The documented change control procedures (6.4.5) must include documentation of impact (6.4.5.1), approval by authorized parties (6.4.5.2), functional testing of security impacts (6.4.5.3) and back-out procedures (6.4.5.4) or how to revert back if unforeseen negative impacts occur.

Often emergency change control processes exist which may allow for a verbal authorization first, but mandate proper documentation following the standard/regular process within a very short timeframe (days, not weeks).

3.7.6.3 Software Development Requirements

In an information world, custom developed software is often a business differentiator for organizations. But the focus is generally on functionality and often does not take into account security until much later in the process. If organizations the size of Microsoft and Adobe end up with vulnerabilities, how improbable is it that smaller organizations will not face the same issues? Most pentesters will attest that insecurely coded applications (as well as misconfigured systems) are often the way we manage to penetrate networks and systems. So how does the PCI SSC recommend we address this?

First, an organization must have a Software Development Life Cycle policy and process (6.3) that is based on industry best practices. This can include the standard waterfall process where each phase (requirements analysis, design, implementation, testing, promotion to production) must be finished and approved before moving on to the next one, or even agile development processes (which release to production much faster and often) such as Extreme Programming (XP), Scrum, etc. Information security should be included in all phases of the process (requirements analysis, design, implementation, testing and promotion to production). Note that this applies to all organizations whose software (purchased or internally developed) is used in a PCI DSS environment. All such applications must also meet other PCI DSS requirements such logging, authentication, etc.

Removal of development, tests accounts and data must be done prior to release or promotion to production (6.3.1) which is similar to requirement 6.4.4. Code reviews (by an application security expert, or at least a different developer than the one who wrote the code) must be performed to identify potential coding vulnerabilities (6.3.2); code reviews may include automated and manual portions. To identify potential coding vulnerabilities, you must :

- ensure that code reviews are performed by someone other than the author (a qualified internal or external person, knowledgeable about secure coding - also see requirement 6.5.*)

- verify that secure coding guidelines are followed (also see requirement 6.5.*)
- verify that recommended corrections are implemented before release
- have code review results reviewed by management as part of the change control process (which is defined in requirements 6.4.*)

This is where buying PA-DSS certified software can help reduce some of those controls. PCI PA-DSS software is software that has gone through an evaluation by a PA-QSA. A PA-QSA is a like a QSA for Software Applications used in a PCI DSS environment. The organization implementing a PA-DSS validated application must follow the implementation guide that comes with the application and place it in a PCI DSS compliant environment. All other 6.3.* and 6.5.* requirements (and possibly 6.6) are taken care of by the PA-DSS certification, simplifying the organization's compliance efforts.

Requirement 6.5 covers basic common web-application coding vulnerabilities. It mandates the development of secure coding guidelines (also required in 6.3.2) and the training of developers on those topics. Sub-requirements closely align to the OWASP top 10 (updated in 2013, just prior to the release of PCI DSS 3.0) and one could say they were at the very least inspired by that list. Other industry standards could be used for the organization guidelines, such as the SANS/CWE top 25 (Common Weakness Enumeration, the top 25 software errors list produced conjunctly between the SANS institute and MITRE Corporation, a not-for-profit company that operates multiple federally funded research and development centers). These requirements cover typical mistakes made by developers that cause easily exploitable vulnerabilities:

- Injection flaws, including SQL injection and others (6.5.1). The flaws are generally caused by non-validated parameters that are sent directly to a subsystem, such as a database or an operating system. In SQL injection for example, this can allow us to bypass authentication or retrieve database information [54] .
- Buffer overflows (6.5.2) - typical in compiled languages such as C, C++, Objective-C, Assembler but often not seen in Java and .Net. A buffer overflow is another type of improper validation ('bounds checking'). A buffer (or reserved memory space) of a fixed size is allocated to the application, then a function (such as strcpy in C/C++) is used to copy data that is longer than the buffer and overwrites a part of the memory where code lies.
- Insecure cryptographic storage (6.5.3). Cryptography is complex and implementation mistakes are common. This requirement will cover these issues.
- Insecure communications (6.5.4). This requirement covers usage of encryption (cryptography) for communication to prevent the disclosure of clear-text credentials (username, passwords), session keys, as well as CHD and other sensitive information

- Improper error handling (6.5.5). This requirements covers what I would call 'degrading gracefully' of 'soft fail' errors. Too often, error screens provide debugging information (including file path information) which can be useful to an attacker. A typical example is when logging, you should never tell a user whether the username or password is incorrect (which can tell an attacker that a user account does exist), but that one of the two is incorrect, for example: “Invalid username and/or password. Please try again.”.

Requirement 6.5.6 ties back to the vulnerability identification process (6.1) to ensure that all 'high risk' vulnerabilities identified are addressed (within 30 days as required by 6.1). This can also be a feedback loop that allows improvements of secure coding guidelines.

We then find vulnerabilities in web-applications and application interfaces (for example web-services). These include:

- Cross-site scripting (XSS, 6.5.7). XSS happens when a web-application includes code from different domains and code injected by a malicious user in one domain can access information in another. For example, a company order website integrates an iFrame component for third-party secure payment so that CHD never enters the organization's network. An attacker manages to modify the organization's website and inject XSS code. When the user puts in his payment information, malicious XSS code manages to grab that information and send it to a site controlled by the attacker. Note that the PCI SSC has produced guidance for e-commerce clarification [55] that you should consult.
- Improper Access Control (6.5.8) can be seen as one form of security through obscurity; it generally means that some form of permission validation (object reference, URL access) was not performed; this could also cover some insecure web server configuration such as permitting file directory listings.
- Cross-site request forgery, or CSRF (6.5.9) builds upon XSS but targets a comprised authenticated user (a client of the application) to make a request to a vulnerable application.
- Broken authentication and session management (6.5.10) is a new requirement introduced in PCI DSS 3.0 that must be in place as of July 1, 2015. This requirement was added to the OWASP Top 10 2013 version [56] . This requirement just means that the authentication and session system can be easily targeted by an attacker. Attacks of this type can include:
  - brute-force guessing of credentials if no account locking is present (8.1.6, 8.1.7).

- capture (eavesdropping) of credentials not protected by encryption (6.5.4, 8.2.1).
- capture of leaked session identifiers (6.5.4) often included in the URL, reused or never timed-out.
- other> attacks are also likely possible.

As you can probably see, improper validation of input values is one source that leads to many of these issues (and directly 6.5.1, 6.5.2).

These are the minimal required checks to be performed. An organization should review the current threat landscape and identify whether other types of vulnerabilities should be covered within its secure coding guidelines and training.

Since externally-facing web-applications are more and more targeted by attackers, any public-facing (externally outside the organization, connected to the full Internet or just to a limited subset through a network not fully controlled by the organization) are especially at risk. PCI DSS requirement 6.6 gives us two options to address this requirement. The first option is to perform an annual security assessment of the application. This is not simply setting up automated vulnerability scans as required by 11.2.*, but more specialized tools with at least some human intervention. I often recommend calling this option web application penetration testing to differentiate it from tools-based vulnerability testing (11.2.*). The second option is to install some form of automated solution that detects and prevents web-attacks. This can mean Web-Application Firewalls, reverse proxies, or other such tools. It goes without saying that if a technological solution is selected, the solution must be kept up to date (6.1, 6.2).

In volume 1 (section 1.10.7), I argued that organizations should probably be using both approaches, and not just one of the two. This is still my personal recommendation.

3.7.7 - Requirement 7 - Need to know

In requirement 3.1 we were asked to limit retention of CHD. In 7.1, we are asked to limit access to CHD to only those who absolutely need it. This will include proper separation of duties to prevent collusion, and includes the concept of least privilege (7.1.2), i.e. granting only the minimum level required to perform a function.

Roles must be defined for specific business and IT functions (7.1.1) that specify which system access and which level of access (user, reviewer, administrator, etc.) is required for each role. Often this will come with job description functions and system/application roles assigned to those functions (7.1.3). Granting of roles and permissions must be documented and approved by authorized individuals (7.1.4).

Requirement 7.2 requires implementing a Role-Based-Access-Control (RBAC) system with a default of *“deny-all”* (7.2.3). A RBAC system simply means that we assign permissions to roles, and roles to users (not permissions to users directly), often through group membership. This reduces the risk that individual permissions will be given that do not belong to an individual, or if that individual changes functions that some permissions not be removed. The RBAC system must cover all components (7.2.1) and assign privileges to individuals (7.2.2) with no shared account used (8.5). Traceability of action is a key objective of PCI DSS (necessary for an investigation should there ever be any form of incident or breach) and requires that roles be assigned to individual accounts, and that no shared accounts be used.

A very good common practice for administrative users is to have dual accounts. Those users will have a regular user account for most functions (email, internet browsing) and an administrative account that is used only for tasks requiring this level of access (not for logging on to their individual workstation). For example, John Doe has a regular 'jdoe' Active Directory (AD) account which he uses to log on to the network and check email, as well as a 'jdoe-a' administrator account which he uses for changes that require administrator privilege. This separation helps reduce risks in the usage of the administrative account. Of course, this implies that some monitoring needs to be in place to ensure that administrators are using administrative accounts only when necessary.

3.7.8 - Requirement 8 - Authentication

In PCI DSS 3.0, multiple sub-requirements of 8.* were moved around to come up with a more logical presentation which I believe helps everyone. This remains unchanged in PCI DSS 3.1.

Requirements 8.1.* now cover user identification, while 8.2.* cover user authentication requirements.

Authentication procedures must be documented and communicated to all users (8.4) and must include the following:

- Guidance on selecting strong authentication credentials, for example choosing hard-to-guess passwords including no dictionary words or words related to known hobbies (favorite sports team, pastimes)
- Guidance for how users should protect their authentication credentials, for example not writing passwords down (on paper or in a file on the computer)
- Instructions not to reuse previously used passwords (or use the same password for organization and personal accounts)
- Instructions to change passwords if there is any suspicion the password could be compromised, such as who to report this to and the requirement to change your password even if compromise is not confirmed

3.7.8.1 User Identificaion and Accounts (ensuring traceability)

All users must have a unique identifier (or account) in each in-scope system (8.1.1). No generic or shared accounts are allowed and existing ones must be removed or disabled (8.5). If shared accounts are required due to technical or business constraints, then proper compensating controls must be put in place (covered in section 3.10) to ensure traceability to the individual.

For example, a 'sudo to root' mechanism (where a user logs on as an individual user and then changes to the root account to perform management tasks) with adequate logging and review (of the usage of this shared account by an individual) may be one such compensating control. Specific requirements for accounts used by shared service providers (8.5.1) are described in section 3.8.2.

Procedures must be in place to add, delete and modify user accounts (8.1.2). Terminated users must have their accounts removed immediately (8.1.3). This is an area where I see organizations struggling. To me, terminations are even more critical than granting access to a system, especially if the user is terminated with cause (versus having resigned). Inactive (unused) user accounts must be removed at least every 90 days (8.1.4). If you find that there are many unused accounts at every review then you should likely review which user access roles and membership actually need access (per requirement 7) and uncover why these were not removed more timely. This may have to be dealt with as an incident (12.10.*).

Vendor accounts are to be enabled only when needed and in use, and should be monitored when used (8.1.5). For remote vendors, monitoring access (and possibly recording) via a jump box is one obvious, but not the only, way to accomplish this. Another could be an administrator initiating a screen sharing session, granting control to the vendor and monitoring what the vendor does (which requires some level of technical understanding by the administrator).

All accounts must be locked after at most 6 failed login attempts (8.1.6) which are potential attacks on the systems and should be investigated as possible incidents (12.10.*, see section 3.8.3). These accounts must be then locked-out out for at least 30 minutes (8.1.7), unless a user whose identity was validated (8.2.2) calls the help desk to reset it (for example, in case of a forgotten password). If a user does not use a system for 15 minutes then the system should be locked and require the user reauthenticate himself to reactivate (8.1.8) (this can be done at the OS level, for example at the Windows lock screen).

3.7.8.2 User Authentication (confirming the identity)

To authenticate the user, the account must be matched with at least one of the following authentifying factors (8.2):

- something you know - a password, passphrase or Personal Identification Number (PIN, in certain cases only)
- something you have - a token (e.g. RSA), a smart card, a smart phone, a certificate installed on a user-assigned computer
- something you are (biometrics) such as fingerprints, iris scans, etc.

If using something you have, the token, card or other device must be tied to an individual account and must not be shared amongst users, both from a procedural and technical standpoint (8.6). There have been reported cases of sharing of an RSA token between support staff by using a webcam to stream the numbers directly to the internet [57] . Convenient yes; insecure and stupid, most certainly.

The credentials (the username/password or other information) must be both stored and transmitted securely in an unreadable fashion (8.2.1). For storage, we're generally looking at a secure and salted hashing function (unless there is a need, reversible encryption should not be used for password storing; in Active Directory this means the option called 'Store passwords using reversible encryption' is not checked). For transmission, this is generally through some transmission encryption such as SSL/TLS (4.1). See crypto primer in section 3.13 for more details. If passwords or passphrases are used, they must be 'complex' for passwords, this is generally interpreted at being least seven characters long and containing both numeric and alphanumeric characters (8.2.3) (for Active Directory, there is a setting referred to as 'complex passwords'). For passphrases, a similar level of complexity is required, which generally means longer phrases with spaces, punctuation or other special characters and numbers.

Password or passphrases should be changed every 90 days (8.2.4) and the last four passwords or passphrases employed should not be reused (8.2.5). When an account is created, an initial unique value should be set that must immediately be changed the first time the user logs on (8.2.6). That initial value should be communicated securely to the user (which means by a different communication channel, often on paper or over the phone). Again, validating the user's identity is required before providing him this information or changing his credentials (8.2.2).

Special care must be taken with user account with access to databases containing CHD (8.7) to protect and ensure traceability of access to CHD (as required by 10.2.1). Note that on some systems such as mainframes, files may be considered databases and this requirement might apply. Direct access to databases with CHD (per requirement 6.4.3, real PANs are not allowed on test systems) must be restricted to database administrators (DBAs). Application access to databases must be made through special single purpose accounts for the application. End-user must never have direct access to the database. All non-DBA accesses must be through programmatic methods (for example stored procedures, views or specific libraries) to properly control access, ensure adequate logging (10.2.1), and prevent attacks (for example injection attacks as defined in requirement 6.5.1).

3.7.9 - Requirement 9 - Physical security

This requirement is generally the best understood one in all of PCI DSS 3.1. This requirement applies to sensitive areas where CHD is transmitted, stored or processed on paper and electronic format. Sensitive areas include data centers, server rooms, call centers, etc., but do not include public-facing (e.g. cashier in store) areas.

All of those sensitive areas require entry controls (e.g. keys, electronic badges) to limit and monitor access physical (9.1). For sensitive areas, we should use video cameras or other access control mechanisms (9.1.1). The goal is to, yet again, ensure traceability. Video recordings and access logs must be kept for at least 3 months and must be immediately available for review in the event of an incident. I would recommend that physical access logs be centralized along with other logs, as mentioned in section 3.7.10.

Access to network-jacks (which provide connections to the internal network) must be protected (9.1.2). This can be through logical controls (for example, Network Access Control or NAC, which authenticates a device before allowing it to connect to other devices in the network) or physical controls (for example, network-jacks are disconnected by default in a network room, where modification to connections requires physical access that is restricted to authorized personnel as per 9.1 and 9.1.1). Physical access to other network equipment, including to wireless access points, must also be similarly restricted (9.1.3).

Physical access to sensitive areas must be authorized based on job function (as in 7.1.3), with access immediately removed upon termination (like in 8.1.3), and ensuring that access mechanisms (keys, badges) are returned or disabled (9.3).

3.7.9.1 Visitors

Procedures must be put in place to identify and authorize visitors (9.4). Visitors must be authorized and accompanied at all times (9.4.1) when in sensitive areas (defined in section 3.7.9). Visitors must be easily identifiable (for example, using a different badge type) and their access must be limited (9.2, 9.4.2). Visitors must surrender their badge (or if electronic, expiration may be programmed) at the end of the authorized period (9.4.3). A visitor log must be maintained for access to sensitive areas (9.4.4) which contains, at a minimum, the visitor's name and firm, and the organization individual authorizing physical access, as well as relevant dates and times. This visitor log must be kept for at least 3 months.

3.7.9.2 Media Management

For PCI DSS, media can include, but is not limited to, physical media such as paper, as well as electronic media such as CDs/DVDs, hard drives, USB keys, and tape backups. The organization must maintain strict control over all media potentially containing CHD (9.7) with adequate inventory logs (9.7.1) and annual (or more often) inventories. Sensitivity of the data held on the media must be classified (9.6.1), generally based on the organization's data classification policy (3.1), so that strict control of distribution (9.6) can be maintained. This does not mean that a label must be placed on the media identifying *“this media has valuable data”*, which would only help an attacker in determining its value. Labeling means being able to know from the label identifier, often looking at some internal management system, what type of data is on the device. Thus, if a tape backup is lost, we should be able to know what was backed up to that tape (and if it was encrypted or not) to see if an incident must be declared (following requirements 12.10.*).

Media should be sent via secure and traceable means such as a secured courier (9.6.2) but only when approved by someone with appropriate authority (9.6.3), which may be the person performing this task as defined in the procedure. Media should be stored in a physically secure location (9.5), preferably in a secure off-site facility (9.5.1). The organization must perform an annual verification of the site's security (which for a third-party may include a physical visit and/or reviewing of external audit reports). Finally, media, like any data identified in 3.1, should be destroyed when no longer required for business or legal reasons (9.8). Hardcopy (paper) should be shredded, incinerated or made into pulp (9.8.1), and electronic media should be made unrecoverable in an appropriate way (9.8.2). NIST SP 800-88, Guidelines for Media Sanitization, provides useful information for disposing of electronic media.

3.7.9.3 Protection of Point-of-Sale (POS) and other payment devices

Requirements 9.9.* are new requirements introduced in PCI DSS 3.0 (although I had previously instructed my clients to implement such procedures years before the standard came out). These requirements became mandatory as of July 1, 2015.

These requirements apply to card-present transactions, that is when a user presents a physical payment card to a device of some kind (Points-of-sale, kiosks, ATMs, etc.). Those devices must be protected from tampering and substitution (9.9). Payment card skimmers have a long history, especially in more automated places such as ATMs, gas payments, isolated kiosks. Brian Krebs has documented very interesting examples on his blog [58] .

Organizations must keep an up-to-date list of all such payment devices, including make and model, location, device serial number or other unique identifiers (9.9.1). Those devices must be manually inspected for tampering (9.9.2) periodically (I would recommend between daily and weekly depending on how often they are left alone) by personnel who have been trained in what to look for (9.9.3) and their review must be logged somewhere (at its simplest form, on a form like the ones often used to note when bathrooms have been cleaned). This includes validating the identity of any repair person before granting them access to the devices, not installing updates without prior verification, and reporting suspicious behavior to appropriate personnel as a potential incident (12.10.*). No guidance is provided on how long to keep the review logs, but I would recommend to keep those at least 3 months, the same length as physical access logs (9.1.1).

3.7.10 - Requirement 10 - Logging & Monitoring (audit trails)

In the (hopefully very unlikely) event of a breach, we need to be able to identify what happened when, and what was done and by whom, to reconstruct the events that occurred. Logs are critical in that function, and requirement 10.1 mandates audit trails (another term for logs) to link all access to system components to each individual user (traceability), which means that all relevant events must be recorded. Requirement 10.2 is more specific that this must be automated and, at a minimum, cover the following events:

- All individual user accesses to cardholder data (10.2.1) - if access is through programmatic methods (8.7) then the best place to log access may be within that method (without forgetting information required by 10.3.* and described below).
- All actions taken by any individual with root or administrative privileges (10.2.2) (remember no shared accounts!).
- Use of and changes to identification and authentication mechanisms. Any creation, deletion, or change to authentication configuration and any changes to accounts, with a special emphasis on accounts with root or administrative privileges (10.2.5). Multiple failed attempts at logging in (10.2.4) are often tell-tale signs of an ongoing attack (trying to guess or brute-force account credentials).
- Any access to audit trails (logs, 10.2.3) or Initialization, stopping, or pausing of the audit logs (10.2.6); preventing logging is one of an attackers standard first steps, and

erasing them is one of the last (note that adding information to logs and log rotation are common functions).
- Creation and deletion of system level objects (10.2.7); system level objects are those running by the operating system and not an end-user; malware often modifies operating system files so it can take hold on a system.

All logs must include the following level of detail (10.3):

- User name or identifier (10.3.1)
- Type of event (10.3.2)
- Date and time (10.3.3)
- Event action success or failure (10.3.4)
- Source or origination of event (10.3.5)
- Identity or name of affected data, system component, or resource (10.3.6).

Audit trails (logs) should be secured so they cannot be altered (10.5). This generally means that logs are thus sent to an independent and internal centralized log server (10.5.3) for both internal and externally facing servers (10.5.4). Separation of duties from standard system administration functions is generally key to protecting audit trail files from unauthorized modifications (10.5.2), and we often see this through monitoring, centralized logging and incident management functions completely split from system administration functions. Viewing access to logs, since it can contain sensitive information (although should not include any CHD, including no full PAN) must be restricted only to those who require it (10.5.1). On the centralized log server(s), the organization must use either file integrity monitoring or some other change-detection software to detect log data changes (such as pruning) that generate an alert, since an attacker will often perform log destruction in an attempt to hide their tracks (10.5.5).

Logs must be retained for one full year (10.7) with the last 3 months immediately available (more time than physical access logs like camera recordings in 9.1.1). Immediately available can be online, archived or restorable from backups. In other words, immediately means readily available in a few hours, but not days or longer.

Logs must be reviewed (this is the monitoring function) to identify anomalies or suspicious activity (10.6) and the use of tools is not only permitted, but encouraged. This is generally done using SIEM (Security Information and Event Management) tools as manual review of logs is generally too much time-consuming. The proper configuration of those tools (to adjust for false positives and negatives) should be an ongoing periodic task (that periodicity should be defined by the organization). Required reviews (minimum daily) include (10.6.1):

- All security events
- Logs of all system components that store, process, or transmit CHD and/or SAD, or

that could impact the security of CHD and/or SAD (generally CDE/CHD, CDE/segmenting and connected/security system)

- Logs of all critical system components (it is up to the organization to define what 'critical' means, but I would include at least all CDE/CHD, CDE/segmenting and connected/security system)
- Logs of all servers and system components that perform security functions (for example, firewalls, intrusion-detection systems/intrusion-prevention systems (IDS/IPS), authentication servers, e-commerce redirection servers, etc.) (generally CDE/segmenting and connected/security system)

Requirement 10.6.2 calls for the periodical review of other logs based on *“the organization's annual risk assessment”* (see section 3.5.2 for the risk assessment). A well known blogger requested clarification [59] through the FAQ process; he was answered in FAQ 1304 [60] . The FAQ states that it *” allows the organization to determine the log review frequency for all other in-scope events and systems that do not fall into those categories”* (those in 10.6.1), so this gives flexibility to the organization. They also clarify that this requirement applies only to in-scope systems. See volume 2 for what constitutes in-scope systems versus out-of-scope ones.

Finally, the standard mentions that any any anomaly or suspicious activity detected must be adequately investigated (10.6.3), potentially instigating the incident management process (12.10.*).

Requirements 10.4.* mandate use of organizational time servers (using the Network Time Protocol, NTP) to ensure that log dates can easily be compared. An organization should maintain a few (but at least two for redundancy) central time servers that are synchronized from industry-accepted time sources (10.4.3) with their time data protected (10.4.2). These servers are sometimes core network switches, routers or Active Directory servers. All critical systems within the organization should be synchronized with these central servers (10.4.1). I would recommend that all (not just in-scope PCI DSS ones) organizational systems be synchronized as well using the same internal sources.

3.7.11 - Requirement 11 - Testing

Do you prefer finding that hole in your system yourself or would you prefer an attacker to do so? I certainly hope you prefer the former, and this is why testing is crucial.

Requirement 11 is all about proactively looking for vulnerabilities that often stem from a failure in IT processes. For example, did you forget to check a server that is also running XYZ software (which should be patched) and may have vulnerabilities? Your policies do mention that you can't connect an unauthorized device to the network right? Could somebody not have gotten that memo? Or not cared enough to read it?

3.7.11.1 Testing wireless networks

The first thing the standard asks us to test for is whether an unauthorized wireless network is connected to your network (11.1). This requires identifying all wireless networks and access points (AP) on a quarterly basis (I would recommend a more timely timeframe). Those wireless networks and APs are then compared to the list of authorized AP and networks that you must maintain (11.1.1). This applies even if there is no direct access from the wireless network to the CDE as we're also looking for networks that a user has connected to the internal network. In heavily populated areas, there can be many wireless networks that are not originating from the premises, but from across the street or another floor. Certain tools will help you pinpoint the location of the APs using signal strength so you can rule out false positives (wireless networks present but physically outside your premises and thus not connected to your network). Should you identify an unauthorized network, you should treat this as an incident (11.1.2) and follow your incident response plan (12.10.*). Note that if you implemented technical controls to prevent connection of unauthorized devices to the network, such as NAC also described in section 3.7.9, you could use this as a compensating control that is stronger than what PCI DSS requires.

3.7.11.2 Vulnerability testing

How about that system or application which you forgot about? Requirement 11.2 is here to the rescue. It mandates that we perform internal and external network vulnerability scans on all in-scope systems, at least quarterly and after any significant change. This means that we need to have a process in place to manage these scans. FAQ 1317 [61] provides the following guidance about 'significant changes':

> *Generally, changes affecting access to cardholder data or the security of the cardholder data environment could be considered significant. Examples of a significant change may include network upgrades, additions or updates to firewalls or routing devices, upgrades to servers, etc.*

Thus, a significant change can include: network topology change, a new major change to a system involved in the storage, processing or transmission of CHD, changes in critical technologies such as segmentation of providing security services, etc. One blogger has also provided a more detailed list [62] . Those changes will be covered in the change control process (6.4) and this list should be reviewed by the assessor to determine whether significant changes have occurred, and warrant more testing.

The vulnerability scanning process must produce four (plus those for 'significant changes') 'clean' scans per year (clean means with no vulnerabilities identified, or all remediated) for all in-scope systems. This can be achieved by combining multiple scans during the quarterly period (11.2). For example, say an organization has three systems: A, B, and C. During the January 1st scan, A experiences vulnerabilities but B and C do not. The organization remediates the vulnerabilities in A, but when they run the scan on February 1st, systems B and C show new vulnerabilities. While these new vulnerabilities need to be addressed within the applicable timeframe defined in requirement 6.1, the January scan (for systems B and C) and February scan (for system A) can be combined (with proper documentation) to show a 'clean' scan for the period.

Quarterly Internal scans (11.2.1) can be performed by internal qualified individuals using industry recognized tools. All 'high' or higher ranked vulnerabilities (see 6.1) must be remediated within a month. Rescans must be executed to confirm the vulnerabilities were remediated.

Quarterly External scans (11.2.2) must be performed by an Approved Scanning Vendor (ASV) [63] . An ASV is a vendor approved by the PCI council (like for QSA companies) to perform this task. ASVs are more of a commodity service so that they can easily be replaced by another vendor from the list maintained by the PCI SSC [64] . Some ASVs offer a fully automated solution with little involvement from the ASV staff (unless Compensating Controls are needed). Some will also allow for multiple rescans at a flat fee (based generally on the number of IP addresses in-scope). Rescans must be executed to confirm the vulnerabilities were remediated within the appropriate timeframe.

Just in case you forgot, after any significant change to the environment, you must rescan the network (internally and externally) (11.2.3). If such a change occurs, this should also require additional penetration testing (described in the next section).

3.7.11.3 Penetration testing

An organization performing penetration testing must have a well-defined methodology based on industry-accepted standards (11.3). If this task is outsourced to a vendor, that vendor should document its methodology with references to the industry standard and provide it to the assessor validating PCI DSS compliance. The methodology needs to cover all in-scope networks and systems, both externally facing as well as on the internal network. It needs to cover both network testing as well as application testing. Obviously, it needs to be conducted by qualified personnel. These changes to the requirements introduced in PCI DSS 3.0 must be in place since July 1st, 2015.

Vulnerability scans are mostly automated tools. They are generally one of the first steps performed during penetration testing. But penetration testing takes it further by using the tester's experience as well as many other specialized tools.

External (11.3.1) and internal (11.3.2) penetration testing must be performed at least annually, or after significant changes are made (see definition in the previous section). Exploitable vulnerabilities must be corrected and then re-tested to confirm their resolution (11.3.3). A new requirement introduced in PCI DSS 3.0 is that if network segmentation is used (see volume 2 sections 2.5.1.3 and 2.6.2), testing of the effectiveness of the segmentation must be performed (11.3.4) to ensure isolation and adequate access-controls restrictions.

My recommendation would be for an organization to have an internal vulnerability scan tool that is used to scan regularly (daily or weekly) all systems (internal and external) to address vulnerabilities in as timely a fashion as possible, based on the level of risk. 3 months is a long time for a vulnerability to be present, especially for systems exposed to the internet (external). Also, please ensure that you keep all relevant documentation demonstrating the work performed.

3.7.11.4 Other detective controls

Another detective control is the requirement for Intrusion Detection Systems (IDS) or Intrusion Prevention Systems (IPS) at the perimeter (Internet and CDE entry points) as well as other critical points in the network infrastructure (11.4). An IPS is an IDS that can also instruct some equipment to automatically block traffic that match a certain network pattern or signature (attacks). Obviously these IDS/IPS systems must be kept up-to-date and the events they generate must be logged and monitored.

A final detective control is the use of change-detection mechanism of modification to critical files (11.5) (often of the Operating System, but also of key applications), which in previous PCI DSS versions was limited to the use of File Integrity Management (FIM) tools. As of version 3.0 of PCI DSS, added flexibility has been provided to use other types of tools, as long as they can be setup to alert appropriate personnel to changes to critical files or configurations. Any alert (11.5.1) must be handled through the incident response process (12.10.*) which will confirm whether we are actually dealing with an incident.

3.8 Other Requirements

3.8.1 Third-party service providers (TPSP)

Outsourcing functions to other organizations can be an efficient way for organizations to fulfill business functions it cannot or does not want to perform in-house, whether for costs or capacity reasons.

Now, one cannot simply use any third-party service provider (TPSP). If that was not obvious before, it is made abundantly clear in the information supplement provided by the PCI SSC in August of 2014. In figure 2 of the information supplement, the due diligence process is presented in the decision tree. If you follow this process, it becomes clear that unless a service provider has either (1) validated and provided evidence of PCI DSS compliance, (2) provided evidence so that the entity has validated that it is compliant, or (3) provided a reasonable plan to achieve compliance, then the entity should select another TPSP. Indeed, the supplement also adds:

> *The use of a TPSP, however, does not relieve the entity of ultimate responsibility for its own PCI DSS compliance, or exempt the entity from accountability and obligation for ensuring that its cardholder data (CHD) and CDE are secure.* [65]

Essentially, you can delegate responsibility to a third-party for tasks, but you cannot outsource your accountability for compliance.

So an organization retains is the obligation to ensure that the third-party service providers it hires are PCI DSS compliant and maintain their compliance with PCI DSS through a program consisting of policies and procedures (12.8), including performing proper due diligence prior to engaging a TPSP (12.8.3). The program must cover maintaining a list of PCI DSS service providers (12.8.1) and monitoring (i.e. validate) the service providers' PCI DSS compliance status at least annually (12.8.4).

When engaging the TPSP and when renewing the contracts, the organization must ensure it has a written agreement from the TPSP that includes *“an acknowledgement that the service providers are responsible for the security of cardholder data the service providers possess or otherwise store, process or transmit on behalf of the customer, or to the extent that they could impact the security of the customer's cardholder data environment”* [66] (12.8.2). PCI DSS 3.0 added a new requirement that ensure that *” information about which PCI DSS requirements are managed by each service provider, and which are managed by the entity”* [67] is properly documented and agreed upon by the organization (the entity) and the TPSP (12.8.5).

PCI DSS 3.0 also introduced requirement 12.9 for service providers (not other entities), which must be in place since July 1, 2015, that ties back to requirement 12.8.2 of the client. The requirement mandates the same written acknowledgement that an entity requires in 12.8.2, this time from the service provider.

3.8.2 - Shared service providers requirements

For PCI DSS, 'shared service providers' are PCI service providers who must comply to PCI DSS and that provide services to more than one client. PCI DSS has included requirements for these service providers since version 1.2 released in 2009. Those requirements are not under a number but under appendix 'A' and are mandated within requirement 2.6 (which was moved from 2.4 to 2.6 in the move from PCI DSS 2.0 to 3.0) which requires performing testing of requirements A.1.1 to A.1.4 of the existing Appendix 'A'.

PCI DSS 3.0 introduced a new requirement outside the appendix that also applies only to shared service providers and not other PCI DSS covered entities. Requirement 8.5.1 mandates that shared service providers with remote access to customer's premises must ensure that individual users use different authentication credentials (username and passwords) for different customers. This requirement tries to prevent that if attackers manage to get the credentials for one customer, these cannot be used to attack another customer. A note clarifies that this does not apply for access to infrastructure managed by the shared service provider and that hosts multiple customers. There, one set of credentials for the complete infrastructure may be adequate.

Requirement A.1 simply asks us to protect each hosted environment and data by meeting the four next requirements. The first two requirements cover logical segmentation (it does not have to be physical) between the different entities (organizations) by using different user account or user IDs (A.1.1), and ensuring that an organization does not have privileges that allow it to access another organization's environment (A.1.2). The language of these requirement still appears written for web hosting providers, but will apply equally to all shared service providers. We are then asked to ensure that logging of each environment meets all of the obligations of requirement 10 (see section 3.7.10), including that logs are available to the client and ensuring that they are reviewed per requirement 10.6.* (who reviews the logs must be agreed upon between the shared service provider and its client, but the logs must be available to the client) (A.1.3). Finally, and linked to the previous requirement, the shared service provider must have defined *"processes to provide for timely forensic investigation in the event of a compromise to any hosted merchant or service provider"* (A.1.4), in other words, logs must be readily available as per requirement 10.7 and the shared service provider must have the necessary resources to assist an investigation in case they are needed.

3.8.3 Incident Management

Incident management is a corrective control invoked by a detective control. Sadly, organizations too often learn of most breaches (confirmed incidents) *"when they receive notification from a law enforcement agency, the card brands, or another third party"* [68] and not through the organization's own monitoring .

Requirement 12.10 asks us to create a an incident response plan that is ready *“to respond immediately to a system breach”*. The plan must cover, at a minimum:

- Assigning roles, responsibilities, and communication and contact strategies in the event of a compromise including notification of the payment brands
- Defining specific incident response procedures
- Covering business recovery and continuity procedures (this is the only mention of BC/DR within the whole standard)
- Data backup processes (since backed up data may contain CHD)
- Analysis of legal requirements for reporting compromises (many countries and states have different breach reporting requirements that organizations must adhere to)
- Reference or inclusion of incident response procedures from the payment brands (provided by acquirers to merchants, or from the payment brands themselves for service providers, issuers and acquirers)
- Coverage and responses of all critical system components (it is up to the organization to define what 'critical' means, but I would include at least all CDE/CHD, CDE/segmenting and connected/security system) and further expanded by requirement 12.10.5 to *“Include alerts from security monitoring systems, including but not limited to intrusion-detection, intrusion- prevention, firewalls, and file-integrity monitoring systems”*

The plan must be tested at least annually (12.10.2); it can be done as a tabletop exercise. Specific personnel must be assigned and available at all times (24/7) to respond to alerts (12.10.3) and this personnel must be trained at least annually (12.10.4). Finally, since we know that systems and attacks are not static, the organization must be able to update, evolve and improve *“the incident response plan according to lessons learned and to incorporate industry developments”* (12.10.6). This is often done through post-mortem analysis of events.

3.9 - Addressing compliance gaps – prioritization

So, you've done (or had someone do) a readiness assessment or just realized that you are not compliant with certain requirements of the PCI DSS. What are you to do? I mean, can anyone expect you to remediate everything overnight? Do you stop operating until then? Of course, not. No business would accept something so drastic. There's an understanding that, since not all things are created equal, that some risks are actually greater than others and the controls required to address those risks should follow the risk level. The PCI SSC also understands this. This is why, with version 2.0 of the PCI DSS standard, they started distributing alongside the current version of the standard a prioritized approach to compliance.

The PCI DSS Prioritized Approach [69] recognizes that not all issues are equal in terms of risk and that some need to be addressed before others. The PCI SSC has divided the PCI DSS requirements into six different milestones, numbered from 1 to 6. Requirements grouped in Milestone 1 are the ones that reduce risk the most and should be addressed first, while Milestone 6 requirements reduce risk the least and might be addressed later.

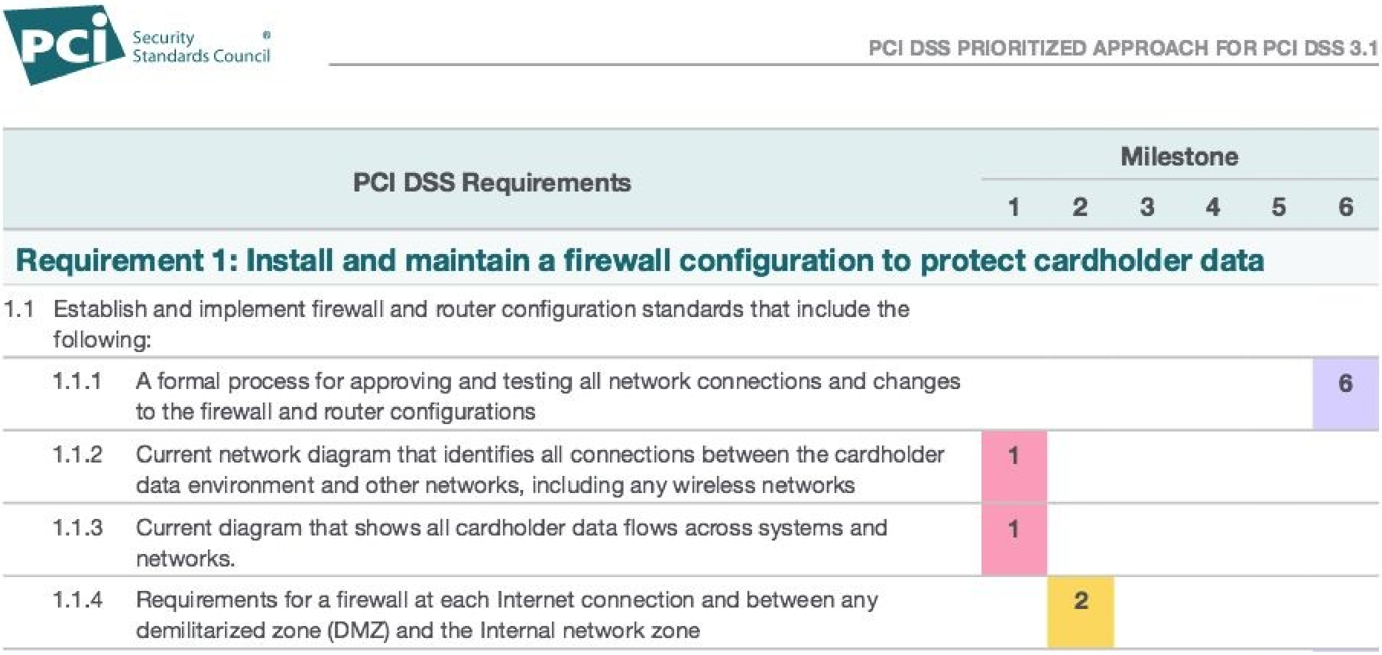
The PCI DSS prioritized approach has not generally changed from version 2.0 to 3.0 and 3.1.

Table 5 below summarizes the high-level actions and goals of each milestone. The PCI DSS prioritized approach document maps the milestones to each of all twelve PCI DSS requirements and their sub-requirements [link].

Milestone	Title	Goals
1	Remove sensitive authentication data and limit data retention.	This milestone targets a key area of risk for entities that have been compromised. Remember – if sensitive authentication data and other cardholder data are not stored, the effects of a compromise will be greatly reduced. If you don't need it, don't store it.
2	Protect systems and networks, and be prepared to respond to a system breach.	This milestone targets controls for points of access to most compromises, and the processes for responding.
3	Secure payment card applications.	This milestone targets controls for applications, application processes, and application servers. Weaknesses in these areas offer easy prey for compromising systems and obtaining access to cardholder data.
4	Monitor and control access to your systems.	Controls for this milestone allow you to detect the who, what, when, and how concerning who is accessing your network and cardholder data environment
5	Protect stored cardholder data.	For those organizations that have analyzed their business processes and determined that they must store Primary Account Numbers, Milestone Five targets key protection mechanisms for that stored data.
6	Finalize remaining compliance efforts, and ensure all	The intent of Milestone Six is to complete PCI DSS requirements, and to finalize all remaining related policies, procedures, and processes needed to protect the cardholder

	controls are in place.	data environment.

Table 5 - PCI DSS Prioritized Approach Milestones



PCI Security Standards Council

PCI DSS PRIORITIZED APPROACH FOR PCI DSS 3.1

PCI DSS Requirements		Milestone					
		1	2	3	4	5	6
Requirement 1: Install and maintain a firewall configuration to protect cardholder data							
1.1	Establish and implement firewall and router configuration standards that include the following:						
1.1.1	A formal process for approving and testing all network connections and changes to the firewall and router configurations						6
1.1.2	Current network diagram that identifies all connections between the cardholder data environment and other networks, including any wireless networks	1					
1.1.3	Current diagram that shows all cardholder data flows across systems and networks.	1					
1.1.4	Requirements for a firewall at each Internet connection and between any demilitarized zone (DMZ) and the Internal network zone		2				

Figure 7 - Screenshot of Prioritized Approach document

The PCI DSS prioritized approach gives us a good idea of which approach to take in achieving PCI DSS compliance. Milestone one is to reduce the amount of information we have and keep on our systems. You don't have to protect what you do not have. This falls in-line with the general recommendation of reducing scope. If you don't need it, don't store it (or collect it in the first place). When we mention that process changes are often the best approach, that would fall directly within this milestone. Note that if there is a business need for the information, you are allowed to keep parts of it (except SAD, covered more in depth in section 3.7.3) provided it is adequately protected. You should however never take the approach *“well I may need it in the future”*. You will almost never need this information (CHD) for this purpose, especially if you are a merchant. Issuers often have to keep more information, but that too should be as limited as possible. The same applies to acquirers and service providers. Note that this is a general information security recommendation I would provide to all my clients regarding not storing sensitive data (and not just cardholder data) unless you absolutely must. Within milestone one, critical requirements include maintaining network diagrams (1.1.2) and PCI data flow diagrams (1.1.3), managing data retention (3.1, 3.2, 9.8) and performing risk assessments (12.2).

Milestone two relates to protecting the in-scope systems, and ensuring that if ever there was a breach, that you could have enough information so that an investigation would allow for identifying how the breach occurred and who may have been involved. Prevent then detect and investigate/react. Within milestone two, we find isolating the CDE (1.*), hardening devices (2.*), securing transmissions (4.*, 8.3), ensuring physical security (9.*), vulnerability testing (11.2, 11.3), intrusion detection (11.4), managing service providers (12.*, 12.9) and incident response (12.10.*).

Milestone three covers payment applications. While operating system vulnerabilities are still prevalent, the proportion of vulnerabilities from applications (with web application being especially targeted) has risen to often become the main avenues that attackers target. Many web apps are also often custom-built within organizations by developers that may not have adequate understanding of secure application development. The Open Web Application Security Project (OWASP, found on the web at www.owasp.org) has risen in prominence in large part due to that fact. Within milestone three, we find system hardening (2.2.*), patching (6.2), secure application development (6.3, 6.5.*), change management (6.4), and web application security (6.6).

Milestone four adds monitoring and access control. Monitoring requires logging which is critical to the incident response function of milestone two. Milestone four covers Role-Based Access Control (RBAC, 7.*), User Management (most of 8) including Identification (8.1) and Authentication (8.2), Logging (10.*) including synchronizing time (10.4, NTP) and monitoring events (10.6), as well as detecting critical changes (11.5)

Milestone five targets stored in-scope data. It covers protecting the PAN (3.3, 3.4, 9.6, 9.7), encryption of stored data (3.5, 3.6) and managing visitors (9.2, 9.4).

Finally, milestone six covers all remaining requirements. It covers mostly policies and procedures.

Just because some items are of a lower risk level does not mean that they should not be addressed immediately. Simple changes, often called low-hanging fruit (e.g. easy fixes), should likely be dealt with quickly.

3.10 - Compensating Controls

PCI DSS 3.1 covers Compensating Controls (CC) briefly on page 16 and then in Appendix B and C of pages 112 to 114.

On page 16, we have confirmation that all compensating controls must be documented (using the compensating control worksheet of Appendix C), reviewed by the organization, and validated annually by an organization's assessor (internally for self-assessment, or externally by a QSA). The PCI SSC is clear in FAQ 1046 [70] that this validation is a responsibility of the assessor (QSA) and not the PCI SSC itself.

Appendix B is where we get more information on what is required to constitute a compensating control.

Almost all PCI DSS requirements can be addressed using compensating controls if a legitimate business or technical constraint exists preventing meeting the requirement *“as stated"*;, but that does not mean organizations should go down that route. The standard mentions that CC must satisfy 3 criteria among the following:

1. Meet the intent and rigor of the original PCI DSS requirement.
2. Provide a similar level of defense as the original PCI DSS requirement, such that the compensating control sufficiently offsets the risk that the original PCI DSS requirement was designed to defend against.

3. Be *“above and beyond"* other PCI DSS requirements. (Simply being in compliance with other PCI DSS requirements is not a compensating control.)
4. Be commensurate with the additional risk imposed by not adhering to the PCI DSS requirement.

So compensating controls must go *“above and beyond"* (#3), and *“meet the intent and rigor of the original PCI DSS requirement"* (#1) and basically address any new risk (#4), which means that the bar is set very high indeed.

To see if we are *“above and beyond"* (#3), the standard also asks us to consider that:

1. Existing PCI DSS requirements CANNOT be considered as compensating controls if they are already required for the item under review. (Password controls cannot be used for other password requirements, existing logging requirements cannot be used for lack of change detection, etc.)
2. Existing PCI DSS requirements MAY be considered as compensating controls if they are required for another area, but are not required for the item under review. (Two-factor is only mandated for external remote access, so it can be used internally to compensate for other requirements)
3. Existing PCI DSS requirements may be combined with new controls to become a compensating control.

Generally, any new compensating control must bring something new to the table (a non-existing requirement), or increase another control's frequency (from weekly to daily, from daily to real-time, etc.).

Appendix C gives us the template that must be filled (also included in the RoC template and the SAQ formats) by the organization or their assessor for each requirement that is not met (one sheet per requirement). No other format is allowed, although I can see that adding appendixes to this may be helpful in some cases. The template includes six elements that require detailed documentation:

#	Definition	Information Required	Explanation
1	Constraints	List constraints precluding compliance with the original requirement.	
2	Objective	Define the objective of the original control; identify the objective met by the compensating control.	
3	Identified Risk	Identify any additional risk posed by the lack of the original control.	

4	Definition of Compensating Controls	Define the compensating controls and explain how they address the objectives of the original control and the increased risk, if any.	
5	Validation of Compensating Controls	Define how the compensating controls were validated and tested.	
6	Maintenance	Define process and controls in place to maintain compensating controls.	

Table 6 - Compensating Controls Documentation Requirements

Item 1 is a technical or business (often costs, but could be regulatory or other) reason why the stated PCI DSS requirement cannot be met. Item 2 is the objective, or risk, that the PCI DSS requirement not met intended to address; this generally is adapted from the 'guidance' column of the relevant PCI DSS requirement. Item 3 is any new risk not identified by item 2. Item 4 is the detailed list of controls used to compensate for the unmet one. Item 5 details how we validate and test that the CC is operating as expected. Item 6 defines what processes must take place to ensure that no failure in the CC occurs.

3.11 Total Cost of Ownership (TCO) and Return-on-Investment (ROI)

One issue that most experts have is when they try to explain their domain of expertise to people with a different level of familiarity, mostly because some things become so obvious with experience that we jump over them. I've often been guilty of this, of taking shortcuts in terms of architecture decision. In my case this is due to my ample/extensive experience in IT operations (system administration), software development, information security and IT audit. This section attempts to document a simplified version of my architecture decision thought process.

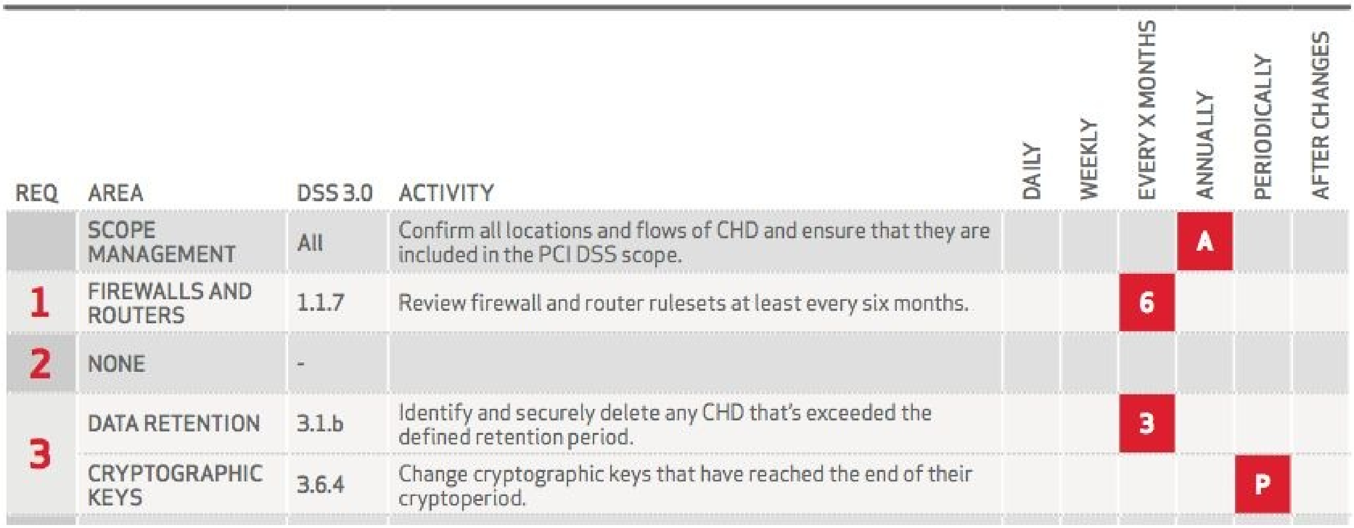
My thought process tries to reduce both information security risk and compliance costs (measured in hardware, software, but also cost in human resources which overtime can be more than the other costs). In that sense I take into account all the relevant costs to produce a recommendation. My evaluation is based on personal experience in IT.

The basic cost framework taught in business schools is:

Total Costs = Fixed Costs + Variable Costs

Our fixed costs when selecting architecture here include hardware and initial setup costs (including hardening). Note that policies and procedures are fixed costs to the organization that are not impacted by architecture decisions.

Most other PCI DSS requirements are variable costs. Appendix C of the Verizon 2015 PCI compliance report presents the period specific requirements that need to be performed (some are only identified as 'periodic' which means that an organization must define the periodicity for itself). More often than not, these variable costs are the biggest ones for an organization in the long run. And all those costs generally include infrastructure (technology: hardware, software, etc.), services (consulting, assessment, vulnerability, scanning, etc.) and staff time.



REQ	AREA	DSS 3.0	ACTIVITY	DAILY	WEEKLY	EVERY X MONTHS	ANNUALLY	PERIODICALLY	AFTER CHANGES
	SCOPE MANAGEMENT	All	Confirm all locations and flows of CHD and ensure that they are included in the PCI DSS scope.				A		
1	FIREWALLS AND ROUTERS	1.1.7	Review firewall and router rulesets at least every six months.			6			
2	NONE	-							
3	DATA RETENTION	3.1.b	Identify and securely delete any CHD that's exceeded the defined retention period.			3			
	CRYPTOGRAPHIC KEYS	3.6.4	Change cryptographic keys that have reached the end of their cryptoperiod.					P	

Figure 8 - Verizon 2015 PCI Compliance Report Appendix C

All of these requirement involve costs, but depending on the architecture decision, these can vary wildly for each organization. My recommendation is that organizations measure (what cannot be measured cannot be improved) the costs (time and equipment) used to manage all systems and environments that adhere to PCI DSS requirements and those that must adhere only to organizational requirements. It then becomes possible to say (this is an example, values are made up) that a MS Windows server costs, on average, $1000 per month (base number) and $1450 for PCI compliant ones (including human resource times). An organization can now do an apples-to-apples comparison ($ to $) of which solution makes a better sense over a number of years (3, 5, 10, etc.) depending on average duration of systems. While this calculation allows us to compare the TCO of alternative solutions, TCO is not ROI (and since it is hard to quantify the actual monetary risk that compliance addresses, a true ROI is difficult to measure), but it does allow us to make more informed decisions.

3.12 Mapping to and Missing ISO/IEC 27002 controls

3.12.1 ISO/IEC 27000 Series

ISO/IEC has created a series of standards in the 27000 series that cover information security under the title *“Information technology — Security techniques”*. The series includes many documents, with 27001 and 27002 being the most referenced ones. ISO/IEC 27001 is published under the title *“Information security management systems — Requirements”*.

ISO/IEC 27002:2013 is an information security standard published in 2013 by the International Organization for Standardization (ISO) and by the International Electrotechnical Commission (IEC). The title for this standard is *“Information technology — Security techniques — Code of practice for information security controls”*.

So ISO/IEC 27001 presents Information Security Program requirements (mostly for audit purposes) while 27002 details the controls that an Information Security Program should include. Both standards share a similar structure.

3.12.2 ISO/IEC 27002 Overview

ISO/IEC 27002 was mostly a renaming of ISO 17799, released in 2005, which was itself the international adoption of the 1995 release of British Standards Institute (BSI) BS7799 of a UK government document. The most recent version of ISO/IEC 27002 came out in 2013 and is referenced as ISO/IEC 27002:2013. This is the version used in the mapping.

The standard includes 5 introductory sections starting at 0

0. Introduction
1. Scope,
2. Normative references,
3. Terms and definitions, and
4. Structure of this standard

and then the 14 *“domains”*, described in chapters 5 to 18:

5. Information Security Policies
6. Organization of Information Security
7. Human Resource Security
8. Asset Management
9. Access Control
10. Cryptography
11. Physical and environmental security
12. Operation Security- procedures and responsibilities, Protection from malware, Backup, Logging and monitoring, Control of operational software, Technical vulnerability management and Information systems audit coordination
13. Communication security - Network security management and Information transfer
14. System acquisition, development and maintenance - Security requirements of information systems, Security in development and support processes and Test data
15. Supplier relationships - Information security in supplier relationships and Supplier service delivery management
16. Information security incident management - Management of information security incidents and improvements
17. Information security aspects of business continuity management - Information security continuity and Redundancies

18. Compliance - Compliance with legal and contractual requirements and Information security reviews

Each of the domains is divided into subdomains and eventually into detailed requirements. For example, the structure for domain 5 *“Information Security Policies”* is:

5. Information Security Policies
    - 5.1 INFORMATION SECURITY POLICY
        - 5.1.1 Information security policy document
        - 5.1.2 Review of the information security policy

3.12.3 ISO/IEC 27002:2013 and PCI DSS 3.1 high-level controls

For this document, I've mapped up to the first dotted number (or subdomain) with a similar high-level structure as PCI DSS 3.1. The mapping was performed to identify related items, and discern where lack of coverage was present.

The following ISO/IEC 27002:2013 domains and subdomains are those used within this mapping:

- 5 Information security policies
    - 5.1 Management direction for information security
- 6 Organization of information
    - 6.1 Internal organization
    - 6.2 Mobile devices and teleworking
- 7 Human resource security
    - 7.1 Prior to employment
    - 7.2 During employment
    - 7.3 Termination and change of employment
- 8 Asset management
    - 8.1 Responsibility for assets
    - 8.2 Information classification
    - 8.3 Media handling
- 9 Access control
    - 9.1 Business requirements of access control
    - 9.2 User access management
    - 9.3 User responsibilities
    - 9.4 System and application access control
- 10 Cryptography

    - 10.1 Cryptographic controls
- 11 Physical and environmental security
    - 11.1 Secure areas
    - 11.2 Equipment
- 12 Operations security
    - 12.1 Operational procedures and responsibilities
    - 12.2 Protection from malware
    - 12.3 Backup
    - 12.4 Logging and monitoring
    - 12.5 Control of operational software
    - 12.6 Technical vulnerability management
    - 12.7 Information systems audit considerations
- 13 Communications security
    - 13.1 Network security management
    - 13.2 Information transfer
- 14 System acquisition, development and maintenance
    - 14.1 Security requirements of information systems
    - 14.2 Security in development and support processes
    - 14.3 Test data
- 15 Supplier relationships
    - 15.1 Information security in supplier relationships
    - 15.2 Supplier service delivery management
- 16 Information security incident management
    - 16.1 Management of information security incidents and improvements
- 17 Information security aspects of business continuity management
    - 17.1 Information security continuity
    - 17.2 Redundancies
- 18 Compliance
    - 18.1 Compliance with legal and contractual requirements
    - 18.2 Information security reviews

The PCI DSS 3.1 summarized controls used in the mapping are:

- 1.1.* Router/Firewall Configuration Standards
- 1.1.1 Router/Firewall Change Proces

- 1.1.2 Network diagrams
- 1.1.3 Data flow diagrams
- 1.2.* Firewall between CDE/untrusted
- 1.3.* Firewall between CDE/internet
- 1.4 Personal Firewall for Mobile
- 2 Configuration Management
- 2.1 Change Default Settings
- 2.2.* System Configuration Standards
- 2.3 Encrypt Administrative Access
- 2.4 / 11.1.1 Inventory
- 3.1 Data Retention and Disposal
- 3.2.* No storage of SAD
- 3.3 Mask PAN (display)
- 3.4.* PAN storage
- 3.5.* / 3.6.* Cryptographic Key Management
- 4.1 Encryption in transit (open networks)
- 4.1.1 Secure Wireless Configuration
- 4.2 No PAN in Email, Chat, etc.
- 5.* Antimalware
- 6.1 Vulnerability Management (id & rank)
- 6.2 Patching
- 6.3.* SDLC
- 6.4.* Change Management
- 6.5.* Secure Application Coding
- 6.6 Protect web-facing application
- 7.1.* Define user roles
- 7.2.* RBAC
- 8.1.* User Identification
- 8.2.* User Authentication
- 8.3 Two-factor for remote access
- 8.4 User training on selecting secure passwords
- 8.5 / 8.6 No shared account
- 8.7 Data access Segregation of Duties

- 9.1.* Physical Access Control and Monitoring
- 9.2 / 9.4.* Visitor management
- 9.3 Physical Badge Access
- 9.5.* Physically secure media
- 9.6.* Classify media
- 9.7.* Control media
- 9.8.* Media destruction
- 9.9.* Device Tampering
- 10.1 / 10.2.* / 10.3.* Logging (data access, admin actions)
- 10.4.* Synchronize time clocks (NTP)
- 10.5.* Securing logs
- 10.6.* Log Monitoring
- 10.7 Log Retention
- 11.1 Identify unauthorized wireless networks
- 11.2.* Vulnerability Testing
- 11.3.* Penetration Testing
- 11.4 IDS / IPS at critical points
- 11.5.* Change Detection Management (FIM)
- 12 Information Security Policy (Program)
- 12.1 Information Security Policy
- 12.2 Risk Assessment
- 12.3.* Acceptable Use Policy
- 12.4 / 12.5.* Information Security Responsibilities
- 12.6 Security Awareness Training
- 12.7 HR Background Checks
- 12.8.* / 12.9 Third-party management
- 12.10.* Incident Response

3.12.4 ISO/IEC 27002:2013 mapping to PCI DSS 3.1

This mapping is not a detailed comparison of every single control (which could not be mapped completely one-to-one anyway), but is performed at a higher level. And since PCI DSS only covers confidentiality and not integrity or availability, there is no overlap over those two elements. The goal of this mapping is to help you identify best practice controls that are not covered by one standard and that you should likely implement as well, or in case you use ISO/IEC 27002 as a basis for compliance with many different regulatory requirements (PCI DSS, HIPAA, Sarbanes Oxley, etc.).

3.12.4.1 ISO/IEC 27002:2013 Domain 5 - Information security policies

This domain contains the policies which map well with PCI DSS requirement 12.1 (and the distributed policies, not the procedures which are covered in domain 12) moved to PCI DSS requirements 1.5, 2.5, 3.7, 4.3, 5.4, 6.7, 7.3, 8.8, 9.10, 10.8, 11.6).

ISO/IEC 27002:2013	PCI DSS 3.1
5 Information security policies	12.1, 1.5, 2.5, 3.7, 4.3, 5.4, 6.7, 7.3, 8.8, 9.10, 10.8, 11.6 Information Security Policies
5.1 Management direction for information security	

3.12.4.2 ISO/IEC 27002:2013 Domain 6 - Organization of information

Domain 6 includes the internal assignment of *"information security responsibilities"* (ISO 6.1) which maps to PCI DSS requirements 12.4 and 12.5.*. These responsibilities must ensure separation of duties (6.1.2). This domain also includes *"mobile devices and teleworking"* (ISO 6.2) which covers PCI DSS requirements 1.4 (Personal Firewall for Mobile Devices), 8.3 (Two-factor for remote access), 4.1.1 (Secure Wireless Configuration).

ISO/IEC 27002:2013	PCI DSS 3.1
6 Organization of information	
6.1 Internal organization	12.4 / 12.5.* Information Security Responsibilities
6.2 Mobile devices and teleworking	1.4 Personal Firewall for Mobile
	8.3 Two-factor for remote access
	4.1.1 Secure Wireless Configuration

3.12.4.3 ISO/IEC 27002:2013 Domain 7 - Human resource security

Domain 7 is divided into three items: before, during and after employment. Prior to employment (ISO 7.1) maps to PCI DSS requirement 12.7 on having HR perform background checks for new hires. During employment (ISO 7.2) maps to PCI DSS 12.6 (Security Awareness Training), but should also likely include a sanctions policy (not present in PCI DSS and which I call out for in volume 1). Termination and change of employment (ISO 7.3) is not mapped to groups of requirements but to individual PCI DSS requirements 9.3 (Physical access is revoked immediately upon termination), and 8.1.3 (Immediately revoke logical access for any terminated users).

ISO/IEC 27002:2013	PCI DSS 3.1

7 Human resource security	
7.1 Prior to employment	12.7 HR Background Checks
7.2 During employment	12.6 Security Awareness Training
7.3 Termination and change of employment	8.1.3 Immediately revoke logical access for any terminated users
	9.3 Physical access is revoked immediately upon termination

3.12.4.4 ISO/IEC 27002:2013 Domain 8 - Asset management

Asset management allows us to know what we're trying to protect (physical assets as well as data). Subdomain 8.1 (Responsibility for assets) is mapped to our PCI DSS inventory of all in-scope systems (2.4) and wireless access points (11.1.1) but also covers our Acceptable Use Policy (12.3.*). Subdomain 8.2 (Information classification) should include a data classification policy that is missing (or merely implied) from PCI DSS, but does include classifying media (9.6.*). Subdomain 8.3 (Media handling) maps to PCI DSS requirements 9.5.* (Physically secure media), 9.7.* (Control media distribution) and 9.8.* (media destruction).

ISO/IEC 27002:2013	PCI DSS 3.1
8 Asset management	
8.1 Responsibility for assets	2.4 / 11.1.1 Inventory
	12.3.* Acceptable Use Policy
8.2 Information classification	9.6.* Classify media
8.3 Media handling	9.5.* Physically secure media
	9.7.* Control media distribution
	9.8.* Media destruction

3.12.4.5 ISO/IEC 27002:2013 Domain 9 - Access control

Logical access controls (physical access controls are in domain 11) are fairly well mapped between both standards. Subdomain 9.1 matches with PCI DSS requirement 7.1.* defining user roles. Subdomain 9.2 is mapped to using Role-Based-Access-Control (RBAC) of 7.2.*, ensuring user identification (8.1.*), including unique user accounts (8.5 / 8.6) to ensure traceability, and Changing Default Settings (2.1). Subdomain 9.3 is mapped to PCI DSS requirement 8.2.* (user authentication) and 8.4 (user training on selecting secure passwords). Subdomain 9.4 covers PCI DSS requirement 2.3 to ensure encrypted administrative access to systems.

ISO/IEC 27002:2013	PCI DSS 3.1
9 Access control	
9.1 Business requirements of access control	7.1.* Define user roles
9.2 User access management	7.2.* RBAC
	8.1.* User Identification
	8.5 / 8.6 No shared account
	2.1 Change Default Settings
9.3 User responsibilities	8.4 User training on selecting secure passwords
	8.2.* User Authentication
9.4 System and application access control	2.3 Encrypt Administrative Access

3.12.4.6 ISO/IEC 27002:2013 Domain 10 - Cryptography

The cryptographic controls are fairly well mapped between both standards. This domain maps partly to PCI DSS requirements for PAN storage (3.4.*, also called *“render PAN unreadable”*) when encryption is used, and also covers Cryptographic Key Management (3.5.* / 3.6.*).

ISO/IEC 27002:2013	PCI DSS 3.1
10 Cryptography	
10.1 Cryptographic controls	3.4.* PAN storage
	3.5.* / 3.6.* Cryptographic Key Management

3.12.4.7 ISO/IEC 27002:2013 Domain 11 - Physical and environmental security

Physical security is the oldest form of information security and is also very well mapped between both standards. Subdomain 11.1 (secure areas) is mapped to PCI DSS requirements for Physical Access Control and Monitoring (9.1.*), Visitor management (9.2 / 9.4.*), andPhysical Badge Access Controls (9.3). Subdomain 11.2 (equipment) is mapped to protecting payment devices from tampering, as well as 9.1.3 (restrict physical access to network devices) (and potentially 9.1.2).

ISO/IEC 27002:2013	PCI DSS 3.1
11 Physical and environmental security	
11.1 Secure areas	9.1 Physical Access Control and Monitoring
	9.2 / 9.4.* Visitor management
	9.3 Physical Badge Access
11.2 Equipment	9.9.* Device Tampering
	9.1.3 Restrict physical access to network devices

3.12.4.8 ISO/IEC 27002:2013 Domain 12 - Operations security

Domain 12 (operations security) is the biggest of the ISO/IEC 27002:2013 domains, but its mapping is partial. Subdomain 12.1 (Operational procedures and responsibilities) is mapped to the procedures portion of PCI DSS requirements 1.5, 2.5, 3.7, 4.3, 5.4, 6.7, 7.3, 8.8, 9.10, 10.8, and 11.6. Subdomain 12.2 (Protection from malware) is well mapped to the complete requirement 5 of PCI DSS (Protect all systems against malware and regularly update anti-virus software or programs).

Subdomain 12.3 (Backup) is not covered by PCI DSS requirements, nor is 12.7 (Information systems audit considerations). Subdomain 12.4 (Logging and monitoring) is mapped to the complete requirement 10 of PCI DSS as well as 11.5 (Change Detection Management).

Subdomain 12.6 (Technical vulnerability management) is mapped to PCI DSS requirements 6.1 (Vulnerability identification & ranking), 6.2 (Patching), 11.1 (Identify unauthorized wireless networks), 11.2.* (Vulnerability Testing) and 11.3.* (Penetration Testing).

PCI DSS requirements 6.4 (Change control, covered more in detail in section 3.12.4.15) is partly mapped to subdomain 12.5 (control of operational software) and 12.1, as well as 14.2 .

ISO/IEC 27002:2013	PCI DSS 3.1
12 Operations security	
12.1 Operational procedures and responsibilities	1.5, 2.5, 3.7, 4.3, 5.4, 6.7, 7.3, 8.8, 9.10, 10.8, 11.6 Procedures

12.2 Protection from malware	5 Antimalware
12.3 Backup	
12.4 Logging and monitoring	10.1 / 10.2.* / 10.3.* Logging (data access, admin actions)
	10.4.* Synchronize time clocks (NTP)
	10.5.* Securing logs
	10.6.* Log Monitoring
	10.7 Log Retention
	11.5.* Change Detection Management (FIM)
12.5 Control of operational software	6.4.* Change Control
12.6 Technical vulnerability management	6.1 Vulnerability Management (id & rank)
	6.2 Patching
	11.1.* Identify unauthorized wireless networks
	11.2.* Vulnerability Testing
	11.3.* Penetration Testing
12.7 Information systems audit considerations	

3.12.4.9 ISO/IEC 27002:2013 Domain 13 - Communications security

Domain 13 (communication security) includes everything regarding network security. Subdomain 13.1 (network security management) maps to multiple PCI DSS requirements including maintaining Router/Firewall Configuration Standards (1.1.*) and Change Process (1.1.1), placing a firewall between the CDE and untrusted internal networks (1.2.*) as well as between the CDE and the internet (1.3.*). It is also mapped to placing IDS / IPS at critical networks points (11.4). Subdomain 13.2 (Information transfer) is partly mapped to PCI DSS requirement 4.2 requiring that no PAN be sent in end-user messaging such as email and chat.

ISO/IEC 27002:2013	PCI DSS 3.1

13 Communications security	
13.1 Network security management	1.1.* Router/Firewall Configuration Standards
	1.1.1 Router/Firewall Change Process
	1.2.* Firewall between CDE/untrusted
	1.3.* Firewall between CDE/internet
	11.4 IDS / IPS at critical points
13.2 Information transfer	4.2 No PAN in Email, Chat, etc.

3.12.4.10 ISO/IEC 27002:2013 Domain 14 - System acquisition, development and maintenance

Domain 14 covers the software development controls of PCI DSS. Subdomain 14.1 (security requirements of information systems) maps to PCI DSS requirements 6.5.* (Secure Application Coding) and 6.6 (Protect web-facing application), and 4.1 (Encryption of data in transit on open networks). Subdomain 14.2 (security in development and support processes) is mapped to the PCI DSS SDLC controls (6.3.*). Subdomain 14.3 (test data) is mapped to some of PCI DSS requirements of change control management (6.4.3, 6.4.4).

ISO/IEC 27002:2013	PCI DSS 3.1
14 System acquisition, development and maintenance	
14.1 Security requirements of information systems	6.5.* Secure Application Coding
	6.6 Protect web-facing application
	4.1 Encryption in transit (open networks)
14.2 Security in development and support processes	6.3.* SDLC
14.3 Test data	6.4.3 / 6.4.4

3.12.4.11 ISO/IEC 27002:2013 Domain 15 - Supplier relationships

Domain 15 that deals with suppliers is mapped to PCI DSS requirements covering third-party service provider management (12.8.* / 12.9).

ISO/IEC 27002:2013	PCI DSS 3.1
15 Supplier relationships	12.8.* / 12.9 Third-party service provider management
15.1 Information security in supplier relationships	
15.2 Supplier service delivery management	

3.12.4.12 ISO/IEC 27002:2013 Domain 16 - Information security incident management

Domain 16 (information security incident management) is mapped to PCI DSS requirements 12.10.* (requesting an incident response plan), and requirement 11.1.2 (invoke incident response if unauthorized wireless is found).

ISO/IEC 27002:2013	PCI DSS 3.1
16 Information security incident management	
16.1 Management of information security incidents and improvements	12.10.* / 11.1.2 Incident Response Plan

3.12.4.13 ISO/IEC 27002:2013 Domain 17 - Information security aspects of business continuity management

Business continuity and Disaster Recovery (BC/DR), like integrity and availability, are not covered by PCI DSS. The only concern slightly related to these is in requirement 12.10.1 to ensure no degradation of confidentiality of CHD in the event of the invocation of BC/DR plans.

Thus an Information Security Program should be reviewed to cover organizational BC/DR needs not mandated by PCI DSS.

ISO/IEC 27002:2013	PCI DSS 3.1
17 Information security aspects of business continuity management	
17.1 Information security continuity	
17.2 Redundancies	

3.12.4.14 ISO/IEC 27002:2013 Domain 18 - Compliance

The compliance domain aligns more with the recently released document Designated Entities Special Validation (DESV). Subdomain 18.1 *“Compliance with legal and contractual requirements”* aligns well with DESV DE.1.* *“Implement a PCI DSS compliance program”*, while subdomain 18.2 *” Information security reviews”* aligns more closely with DE.3.*, *“Validate PCI DSS is incorporated into business-as-usual (BAU)* *activities”*. Subdomain 18.1 also maps partly to requirement 3.1 of PCI DSS (Data Retention and Disposal Policies)

ISO/IEC 27002:2013	**PCI DSS 3.1**
18 Compliance	
18.1 Compliance with legal and contractual requirements	DESV DE.1.*
	3.1 Data Retention and Disposal Policies
18.2 Information security reviews	DESV DE.3.*

3.12.4.15 PCI DSS 3.1 requirements partially or not covered by ISO/IEC 27002:2013

While ISO/IEC 27002:2013 is a very comprehensive information security framework, there are nonetheless specificities of PCI DSS that are not, or at least not fully, covered by the ISO/IEC standard.

Scope definition is an area that is more specific in PCI DSS. In this area, we identify the following unmapped PCI DSS requirements:

- 1.1.2 Create and Maintain Network Diagrams
- 1.1.3 Create and Maintain Data Flow Diagrams

Management of certain information is also more detailed within PCI DSS 3.1 (and not covered by ISO/IEC):

- 3.2.* No storage of SAD after authentication
- 3.3 Masking of PAN during presentation on screens and receipts
- 8.7 Data Access Controls and Segregation of Duties (including usage of programmatic methods by everyone else but DBA's, partially mapped to ISO/IEC control 6.1.2 Segregation of Duties)

PCI DSS requirements 6.4.* covering change management is mostly addressed by parts of two ISO domains (12 and 14) and the following subdomains:

- 12.1 Operational procedures and responsibilities
- 12.5 Control of operational software
- 14.2 Security in development and support processes
- 14.3 Test data

The PCI DSS requirement covering Risk Assessment (12.2, at least annually and during significant changes) is covered partly in 6.1.5 (information security in project management) but this only covers doing so in projects and does not require annual review of the complete environment as required by PCI DSS. As stated earlier in section 3.5.2 of this volume, the ISO/IEC 27005:2011 standard called *“Information security risk management”* may be employed for these risk assessments.

Finally, the concept of 'hardening' also referred to in PCI DSS as 'System Configuration Standards' (2.2.*) is not covered well by the ISO standard. While we may get some requirements that map partly to some sub-elements, this more procedural and technical portion of the PCI DSS standard is mostly not covered.

3.13 A primer on encryption

3.13.1 What is encryption?

So what is encryption and why should you care?

Encryption is all around all of us, and most of us do not even realise it. We use it whenever we do online shopping (hopefully), or access the biggest free email providers (gmail, hotmail/outlook, yahoo, others). It's securing information on our phones and our computers, both personal and business. Encryption keeps information safe. It's gotten to the point that Arthur C. Clarke so famously described as *“any sufficiently advanced technology is indistinguishable from magic”*.

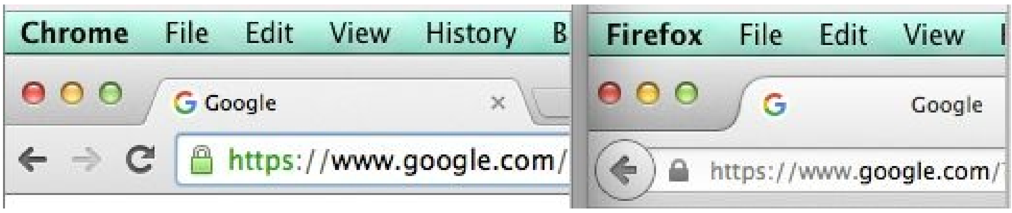


Figure 9 - The lock icon on https://www.google.com on Chrome and Firefox under a Mac

A few years back in another job, I initially considered creating a small course called *'encryption for auditors'*. This is the evolution of this initial idea. It presents an accurate, but simplified (meaning not all details are presented) explanations.

This document is just an overview of cryptography. It will not make you a cryptographer, or a cryptographic engineer. A cryptographer designs (and evaluates) cryptographic algorithms, while a cryptographic engineer designs solutions using cryptography. One can do both, but these are very complex functions and, as with many things, the devil often lies in the details.

Cryptography, the science of encryption is a very complex science that requires strong mathematical bases. This primer will not go into the mathematics, but will give you the basics of how encryption is used in IT environments, for meeting security requirements such as those in PCI DSS.

I encourage users that wish to understand encryption more in depth (including the math) to review books and online courses. Users of cryptography, like with most fields of science, get to enjoy the fruits of the scientists (in this case, the cryptographers) without all of the hard work.

The Merriam-Webster dictionary defines encryption as: *“to change (information) from one form to another especially to hide its meaning”*. The science of encryption is called cryptography. Webster's definition for cryptography is a bit clearer: *“the process of writing or reading secret messages or codes”*. Wikipedia gives us a clearer description still: *“In cryptography, encryption is the process of encoding messages or information in such a way that only authorized parties can read it. Encryption does not of itself prevent interception, but denies the message content to the interceptor”*.

So encryption is one way we use to protect information from unauthorized people, even if they can get a hold of the message. To encode such a message, some sort of key is required. This is akin to being around a group speaking another language. Without the *'key'*, in this case understanding of the language, the speakers can say anything they want and we cannot gain any information of what is being said. This was exactly what the US military leveraged during World War II. The Navajo people and their language were used to exchange secret messages that the enemy could not understand (neither could regular US military personnel).

The easiest definition to agree on, is that encryption is the process of securing information so that it can only be viewed by someone with the right key. Someone without the key can do nothing with the information.

3.13.2 Encryption basics

Reading most security standards, you will hear of two cases of encryption: for data being transmitted, referred to as encryption of data-in-motion, and for data stored, referred to as encryption of data-at-rest.

There are generally two processes in encryption.

- Encryption takes an input, the message, and through a key and a specific algorithm, generates an output, which we call cipher-text.
- Decryption takes the cipher-text as input, alongside the decryption key and algorithm, generates the original message as output.

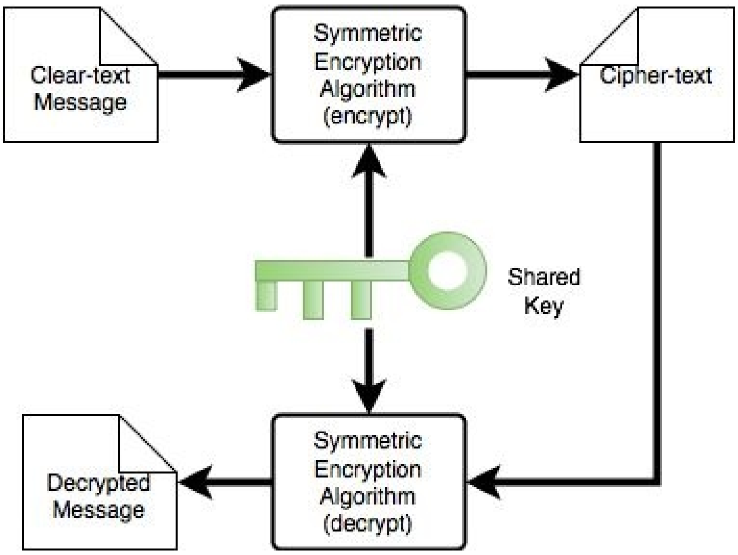

Clear-text Message
Symmetric Encryption Algorithm (encrypt)
Cipher-text
Shared Key
Decrypted Message
Symmetric Encryption Algorithm (decrypt)

Figure 10 - Basic Symmetric Encryption Process

In cryptography, the algorithm used is called a 'cipher', and the resulting encrypted text is called 'cipher-text'.

In both cases, the algorithm, or cipher, used is generally publicly known, has been vetted by the cryptographic community over years, and its strength lie in the key(s) used. The example of the AES standard describes this well. In 1997, NIST created a contest to replace an aging cipher called Triple-DES (more in section 3.13.3.1), created in the 1970s. Cryptographers worldwide were to present for selection their cryptographic algorithms, or cipher, over the following months. Then, during the next couple of years cryptographers evaluated and tested the proposed algorithms. Finally, based on all this evaluation, one cipher was selected out of the five remaining ones. The winning cipher, Rijndael, was announced on October 2, 2000, and then approved in 2001. AES is now one of the most used ciphers out there.

Keys, just like passwords, need to be *'strong'*, meaning hard to guess. For keys, this means the sufficiently large key length (generally measured in bits, 0/1 pairs) and a lot of *'randomness'*. Randomness is one of those details generally understood but harder to define and that we will not delve into here. Suffice to say, randomness is required.

3.13.3 Ciphers, cryptographic algorithms

Ciphers can be categorized in several ways.

One type of category is whether we are working on files or network communication. If dealing with fixed size blocks of data, usually for files, we use what we call block ciphers. If working with a continuous stream of information, as in the case of any network communications, then we use a stream cipher.

A second more useful categorization for our purposes is as cryptographic primitives. There are basic cryptographic functions (called primitives) that we need to understand before going forward. I'll go over the three most common ones: symmetric cryptography, asymmetric cryptography and hashing functions.

3.13.3.1 Symmetric cryptography

Symmetric, as in *'the same viewed both ways'*, refers to the fact there is only one single key used for both encryption and decryption. This is why we often refer to it as private-key or shared-key encryption. This means that the key must be shared between the various people (or processes) that have access to the information (or message). And we all can understand that the more people know of something, the less secret it is likely to remain. See figure 10 for a representation of symmetric cryptography.

The most known and commonly used symmetric ciphers are: DES, 3DES (also called Triple-DES, TDES, which uses 3 DES iterations with 3 different keys), AES, Blowfish and Twofish.

3.13.3.2 Asymmetric cryptography

Asymmetric encryption is the contrary to the symmetric encryption, meaning the use of different keys, and is also referred to as public-key cryptography. Two keys are generally used. They are usually referred to as the public key and the private key. The private key is to be kept very secret by its owner, and the public key is shared with everyone else. Those two keys are mathematically related to each other and this allows the algorithm to work; the mathematical details vary per algorithm. But the private key must be extremely well guarded. Now, you may say to yourself, it seems that asymmetric is more secure since less people will have the keys so it should be easier to protect. You would be right. The downside to asymmetric cryptography is that it is vastly slower than its symmetric counterpart. You'll see in the examples below that we actually often use all three forms of primitives, leveraging the strengths that each provides.

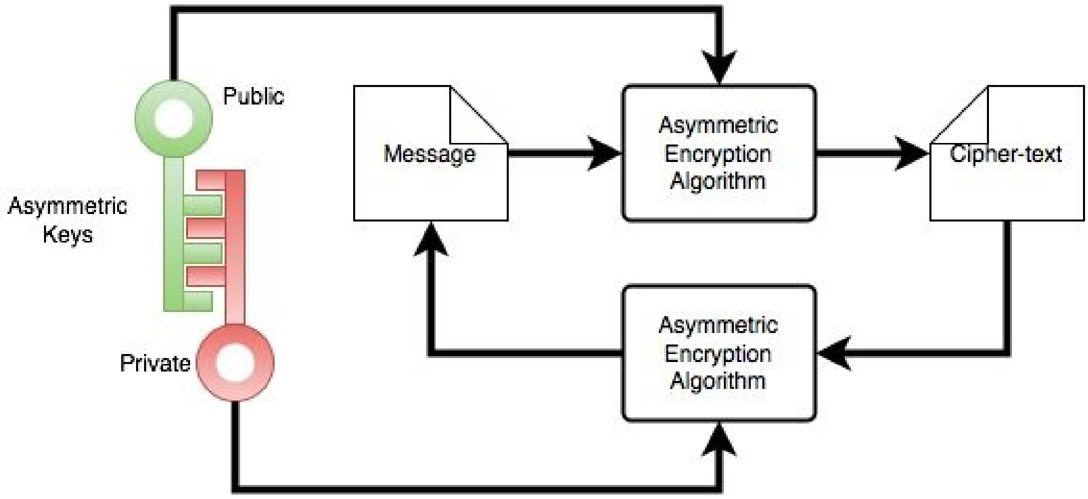

Public
Asymmetric Keys
Private
Message
Asymmetric Encryption Algorithm
Cipher-text
Asymmetric Encryption Algorithm

Figure 11 - Asymmetric cryptography

The most famous asymmetric ciphers are RSA and Diffie-Hellman. Elliptic-curve ciphers are also currently gaining traction.

3.13.3.3 Hashing functions

Hashing functions, unlike symmetric or asymmetric functions, are irreversible. They are also often called *'one-way functions'*. That is, there is no decryption process involved with them. It is not possible to recalculate the input from the output. A hashing function takes an input message of any length (very small or very big), and through an algorithm, generates an output of a specific length to an algorithm. The output is always the same for a defined input. But if even only one character changes in the input message, the output will be completely different. We call this process *'hashing'*, or say that the value is *'hashed'* or that we get the *'hash'* of the value.

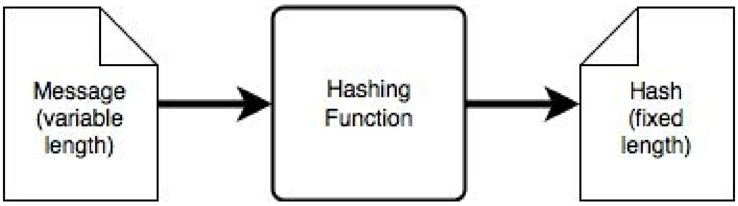


Figure 12 - Hashing functions

Some of the most common hashing algorithms include MD5 and the SHA series such as SHA-1, SHA-256 . Here are a few examples for the message *'password'* (without the quotes):

Algorithm	**Lengths (bin)**	**Hash of *'password'***
MD5	128	5f4dcc3b5aa765d61d8327deb882cf99

SHA-1	160	5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8
SHA256	256	5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721

Table 7 - Example of different hash values for 'password'

3.13.4 Usage of cryptographic primitives

The usage section is here to help you understand real world example of cryptography.

3.13.4.1 Usage: secure storage of passwords

Storing passwords *“in clear-text”* or unencrypted should never be done. If any attacker were to get access to the systems, then they would have those passwords, and they could do anything as if they were you (impersonation). Hashing functions are the cryptographic primitives generally used for storing passwords. When you set your new password to a system, that password is then hashed and that hash is saved. When you try to log in again, the password you entered is hashed and compared to the hashed version stored. That way, your password is never stored in clear-text (that an attacker could see and then use to impersonate you). This is why most audit controls for password will use the language “passwords are stored using non-reversible encryption”. 'Non-reversible encryption' is another word for hashing functions, a one-way function.

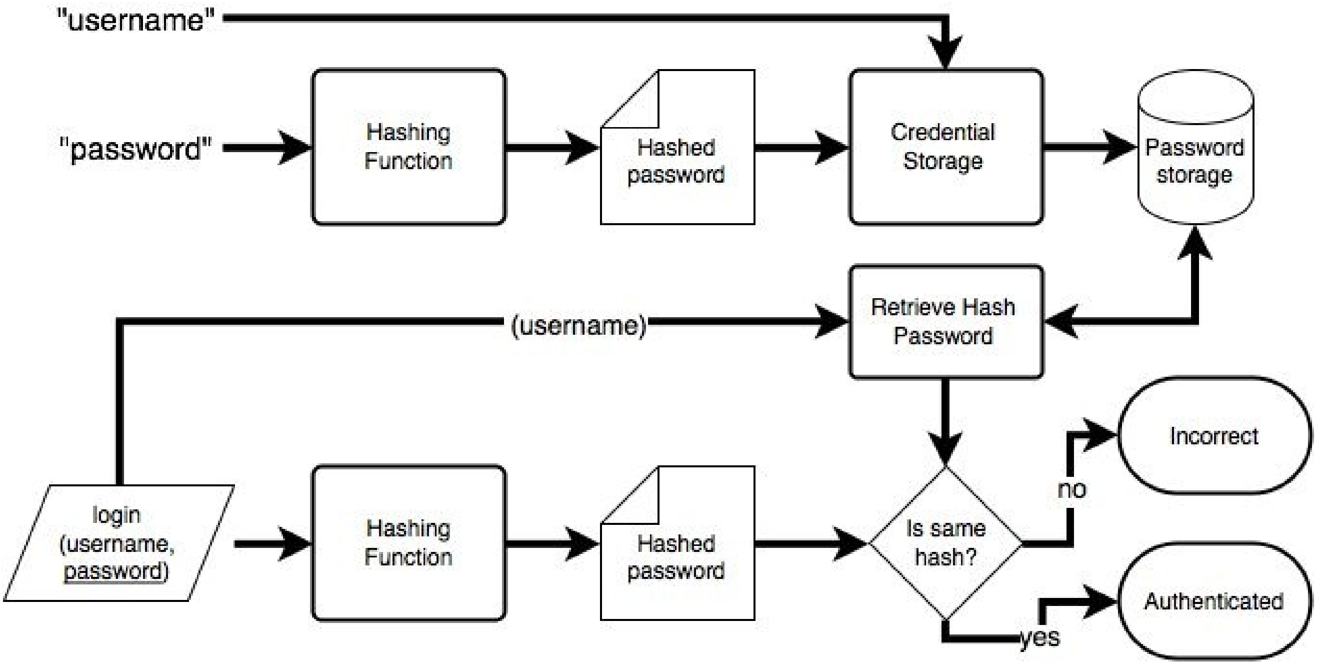


Figure 13 - Password storage and validation

Hashing functions, like any cryptographic functions, are of varying strength depending on the algorithm in use. And depending of the use case (passwords, file integrity) different algorithms can be used.

Now remember how I mentioned that passwords should be hashed? Or that it is not possible to recalculate the input from the output? Well both are true, but also have their limits. And that can be a problem if we simply use the algorithms (although some algorithms are better than others) for typical password values. Say we decide to store passwords using MD5. I create the password *'Password23'*. This gives us a hash of *'37f2b0b7eff2cd34bb5cbd77c14d4850'*. While *'Password12'* gives us a hash of *'08f5b04545cbf7eaa238621b9ab84734'*. Anybody can calculate these since everyone knows the algorithm. And therein lies the issue. If a system used in multiple (think millions) of systems worldwide were to use the same worldwide function, then the hash for a specific password would always be given the same hash.

Thus, an attacker targeting systems like these could calculate the hash of all possible permutations (or variations of all possible characters) of passwords and store these in a big database. If he gets access to a system with these hashes, all he now has to do is look in that very big database for the hash and he could retrieve the password. We call such a process brute-forcing, meaning the attacker actually had to calculate all the values to get back at the password. Precomputed tables like the ones I described are called *'rainbow tables'* in information security. And many of these are already made by attackers and available for download (some at a fee). So what are we to do about this? Well that's where we can make an attacker's life harder by adding what we call a *'salt'*. A *'salt'* is a fixed value that will be prefixed to the input message so that the hash output is changed.

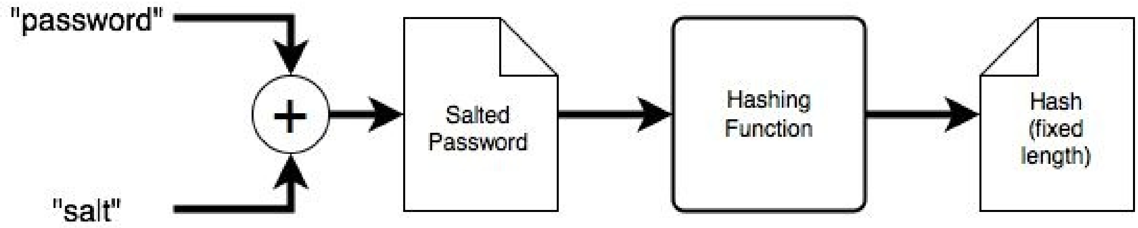


Figure 14 - Diagram of salted hash process

The value of a *'salted'* hash is that precomputed data for that hash may not exist and may make the attack much harder for an attacker who will have to perform the brute-forcing himself.

One can also use multiple iterations of the hashing algorithm to make an attacker's job more difficult. Some algorithms are more demanding and make this an almost impossibility for almost all attackers, unless insecure passwords are used (but that is a topic for another day).

3.13.4.2 Usage: Transmitted (or Stored) Data - example OpenPGP

Pretty Good Privacy, or PGP, was an application created by Phil Zimmermann in 1991 for encrypting data that was to be transmitted. The OpenPGP standard was born out of this application and is now used not only in PGP (owned by Symantec) but also by the GnuPG application, as well as others.

The OpenPGP standard uses the basic primitives we've just learned about, plus a few others. It starts by compressing the message. A session key, a one-time-only secret key, is then generated. The session key is a symmetric (shared) key which, using a symmetric cipher, to encrypt the compressed message will result in secure cipher-text. The session key is then encrypted with the asymmetric public-key of the recipient, meaning that only the recipient (who is the only one with the private-key) can decrypt the session key and decrypt the cipher-text.

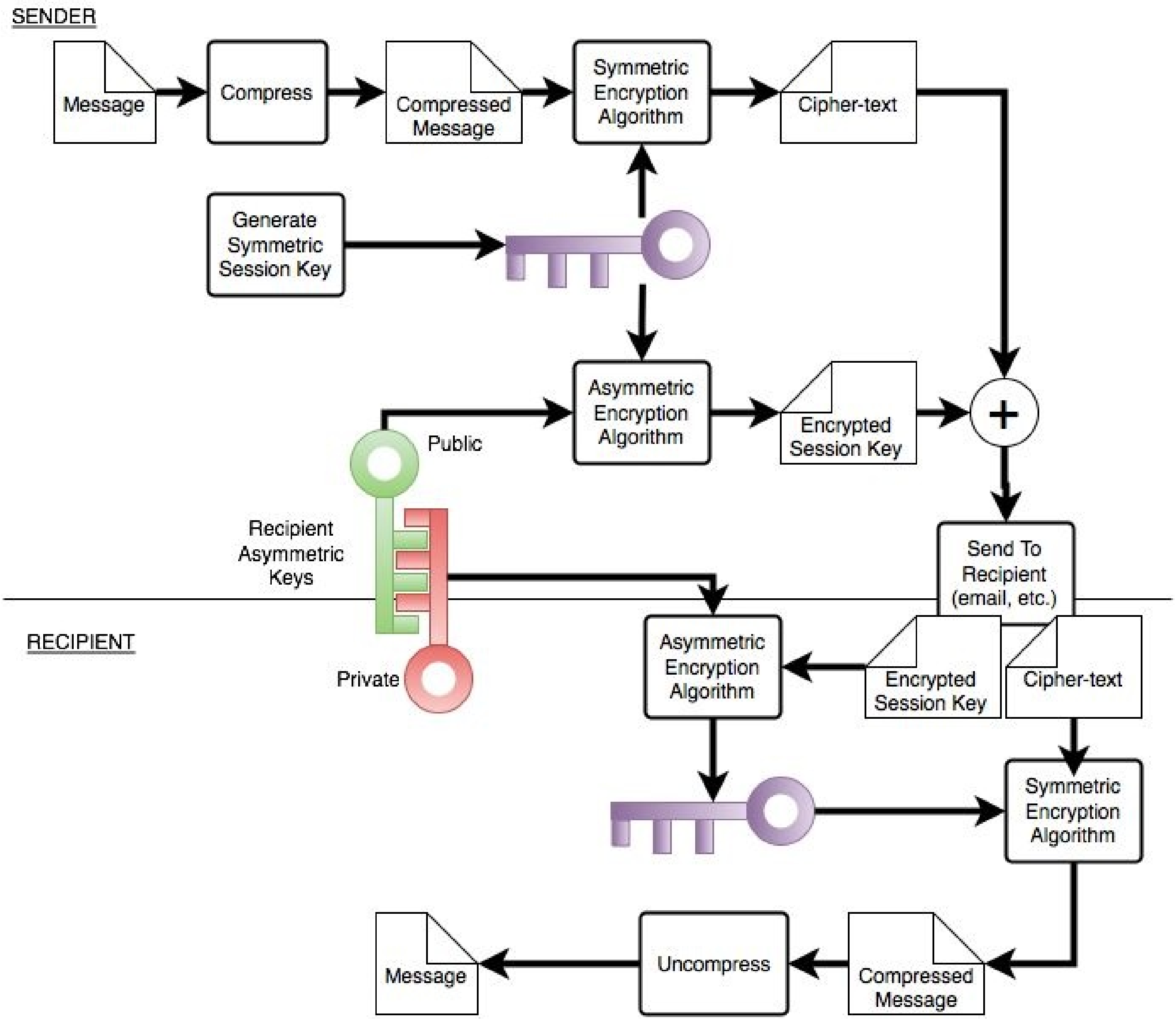

SENDER
Message
Compress
Compressed Message
Symmetric Encryption Algorithm
Cipher-text
Generate Symmetric Session Key
Asymmetric Encryption Algorithm
Encrypted Session Key
+
Public
Recipient Asymmetric Keys
Send To Recipient (email, etc.)
RECIPIENT
Private
Asymmetric Encryption Algorithm
Encrypted Session Key
Cipher-text
Symmetric Encryption Algorithm
Message
Uncompress
Compressed Message

Figure 15 - PGP encryption and decryption processes

3.13.4.3 Usage: Digital Signatures - OpenPGP

OpenPGP can also be used to create digital signatures. Digital signatures allow one to prove that they actually sent a message. For this purpose, we need two primitives: asymmetric encryption and hashing functions.

So how do we do it? Well, we first take the message to be sent and calculate a hash using a secure and well-defined hashing function. We thus get a small piece of text, the hash. We then encrypt this hash using our private-key, and send both the message and the digital signature together to the recipient. Now, I know this is not the standard process, but bare with me a minute. The two keys are reversible in nature. Thus, what we encrypt with the private-key can be decrypted with the public-key. Since the public-key is available to anyone, such an encryption is not secure, but that is exactly the point. Anyone who gets the message can calculate the hash themselves, and compare it by decrypting (with the public key) the value received. If they are the same, then we know it was sent by the legitimate sender (as long as the sender managed to keep his private-key secure).

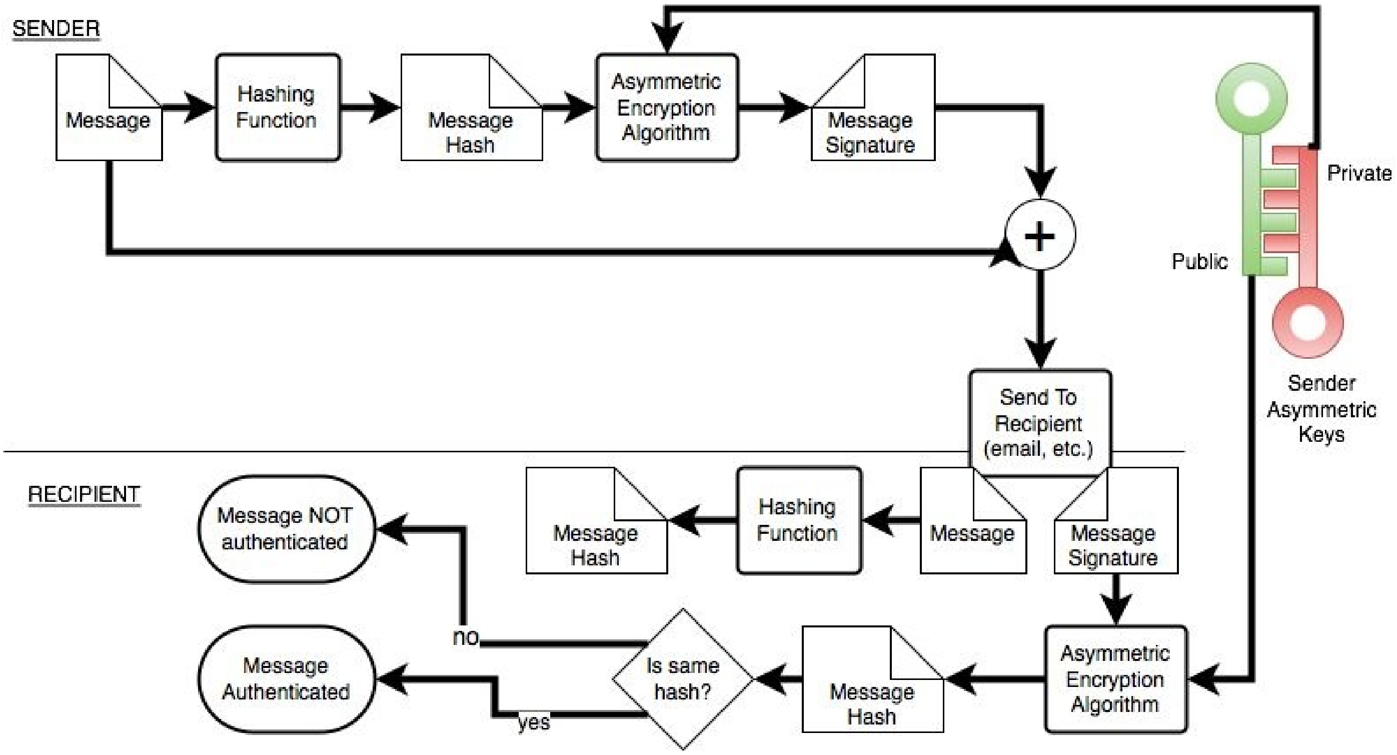


Figure 16 - Digital Signatures using PGP

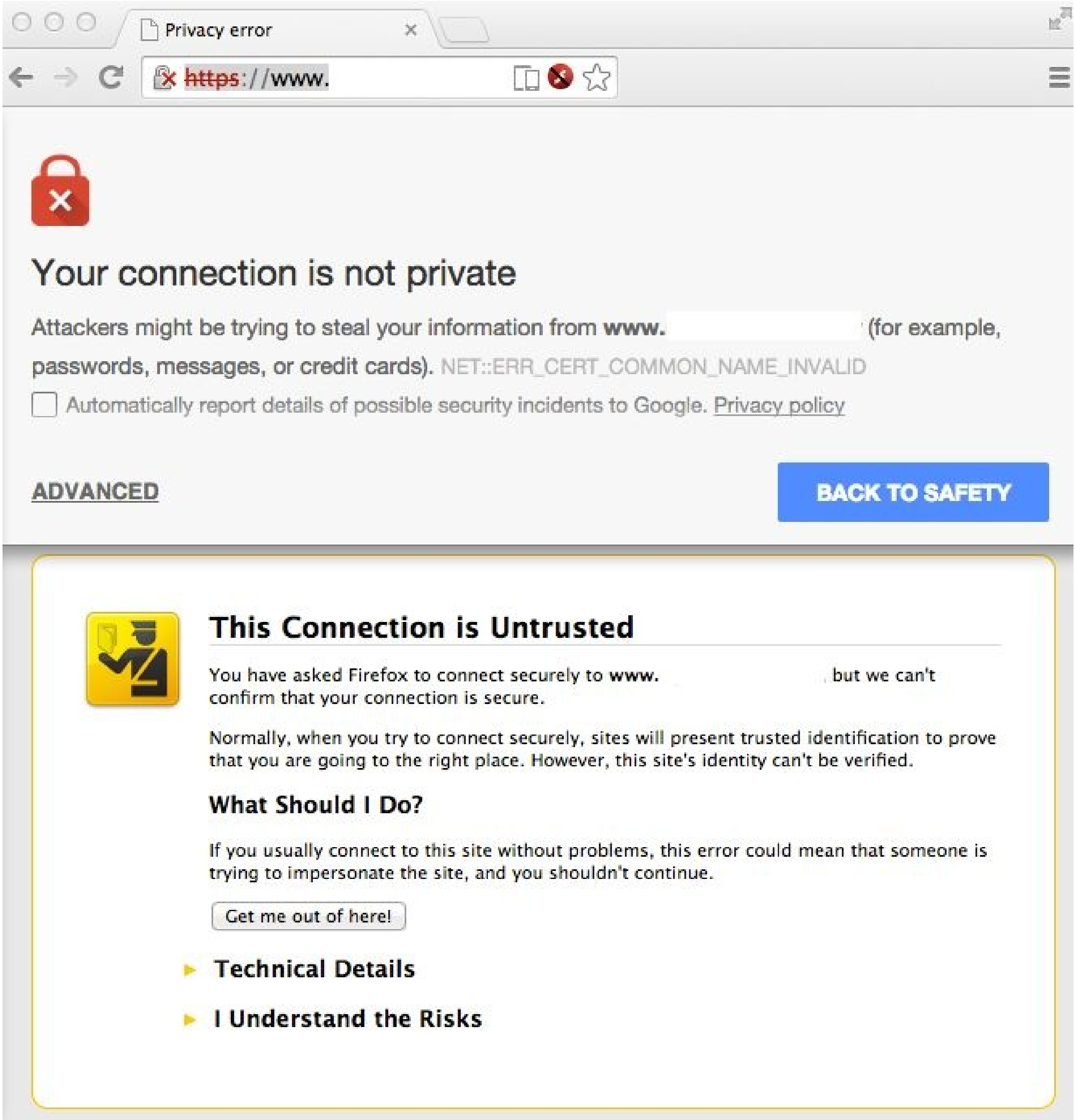
3.13.4.4 Usage: Sent Data - HTTPS (SSL/TLS)

Hypertext Transfer Protocol Secure (HTTPS) is a protocol that allows encryption when transmitting data on the web. It is the most used encryption protocol in use today. You've all seen it when the lock icon appears in the site address you are visiting (see figure 9 for an example).

HTTPS is actually the use of the web protocol, HTTP, over the SSL/TLS protocol. SSL was created in the mid 1990's by Netscape (which later became the Mozilla and maker of Firefox web browsers) to secure connections and prevent eavesdropping and tampering by man-in-the-middle (MITM) attacks. In a MITM attack, someone in the middle acts as a middle-man and can eavesdrop and even modify the information sent. TLS is the evolution of SSL as an independent standard. TLS 1.0 would have been SSL 3.1. SSL has not been updated since 1996. SSL (all versions) and TLS 1.0 are considered insecure and more recent versions of TLS should be used.

Now we know how to use encryption to keep data safe, but how do we know that somebody is not intercepting our information? This is why we use SSL/TLS certificates. Certificates are an electronic document to prove, through digital signatures, ownership of a public key.

The process of negotiation is called the SSL handshake (an agreement on how to communicate securely, presented graphically in figure 18). A user starts a connection using a web browser (some applications also do this) transparently to the server, and the browser and server negotiate some details of which version of SSL/TLS and which ciphers to use. The server then returns its SSL/TLS certificate. This certificate includes a public-key for asymmetric cryptography. The certificate is validated by the browser. This can include matching the name of the server (i.e. a certificate for www.yahoo.com should not be used for a www.google.com site), whether the certificate has expired (valid dates), and its ownership (more on the validation of ownership a little later on). If the certificate is valid, everything continues as expected. If there's something wrong, the browser generally presents a warning message or page (see example in figure 17).



Your connection is not private

Attackers might be trying to steal your information from www. (for example, passwords, messages, or credit cards). NET::ERR_CERT_COMMON_NAME_INVALID

□ Automatically report details of possible security incidents to Google. Privacy policy

This Connection is Untrusted

You have asked Firefox to connect securely to www. but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

▸ Technical Details

▸ I Understand the Risks

Figure 17 - SSL/TLS certificate errors in Chrome and Firefox

In some cases, the server can also request a certificate from the client and validate it in much the same way. The browser generates a pre-master secret that it encrypts with the server certificate's public key and then sends it to the server. The server decrypts the pre-master secret using its certificate private-key. That pre-master secret, through a series of mathematical steps, is converted to a symmetric session key by both the browser and the server. These two finalize the handshake and secure communication can then begin using the session keys for encryption. You may notice that when you start a connection to an encrypted website (https) the connection often seems sluggish, but later feels much faster. This is simply because the handshake requires slower asymmetric encryption and other calculations, while the later part uses the faster symmetric encryption and the session key.

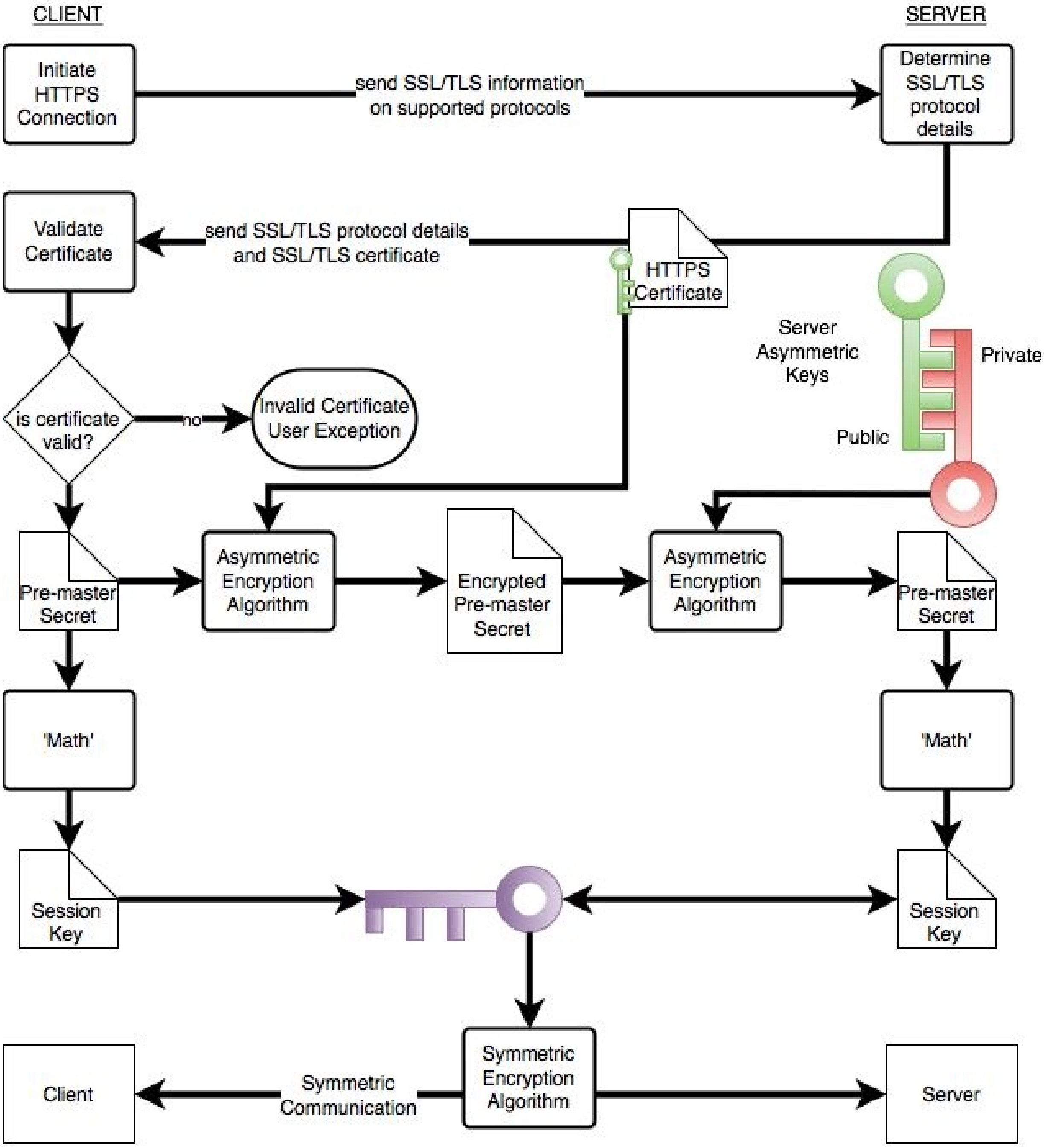

CLIENT
SERVER
Initiate HTTPS Connection
send SSL/TLS information on supported protocols
Determine SSL/TLS protocol details
Validate Certificate
send SSL/TLS protocol details and SSL/TLS certificate
HTTPS Certificate
Server Asymmetric Keys
Private
Public
is certificate valid?
no
Invalid Certificate User Exception
Pre-master Secret
Asymmetric Encryption Algorithm
Encrypted Pre-master Secret
Asymmetric Encryption Algorithm
Pre-master Secret
'Math'
'Math'
Session Key
Session Key
Symmetric Encryption Algorithm
Client
Symmetric Communication
Server

Figure 18 - HTTPS handshake

Now back to certificate validation. To confirm the ownership of the certificate, we need to get someone to vouch for it. But how can someone know all of the websites and validate their certificates? No one can. What happens is that we have organizations called certificate authority (CA) to aid us in that. CAs are organizations that are already recognized by browsers as trusted authorities. They form the *'root'* of the certificate process. Each web browser stores *'root'* certificates for authorities it trusts.

A CA signs the certificate (basically the public-key) of a site (using a process similar to the PGP digital signature process described in section 3.13.4.3) using the CA's private-key (see figure 19 for a graphical representation of this process). There can even be a chain of those certificates with many intermediate CAs, in which case there will be a series of signed public-key certificates with each level signing the next level. In order to validate a site's certificate (this process is presented on the right-hand side of figure 19), we first retrieve the 'chain' of certificates and start validating the signatures in the sequence. This is generally performed starting at the root certificate and working towards the site's certificate, but the important task is to confirm that the chain has not been broken and that all signatures are as expected.

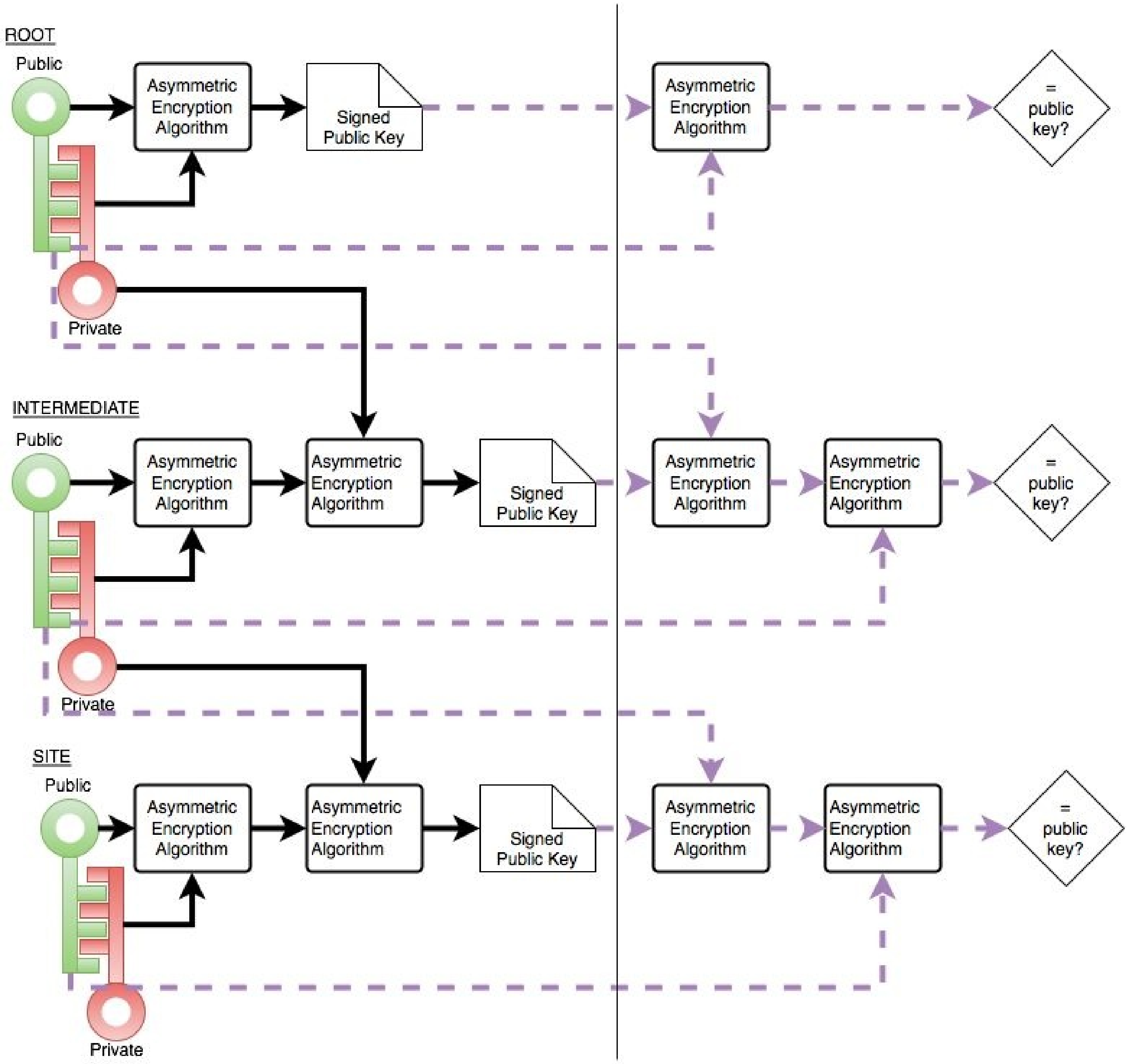

ROOT
Public
Asymmetric Encryption Algorithm
Signed Public Key
Asymmetric Encryption Algorithm
= public key?
Private
INTERMEDIATE
Public
Asymmetric Encryption Algorithm
Asymmetric Encryption Algorithm
Signed Public Key
Asymmetric Encryption Algorithm
Asymmetric Encryption Algorithm
= public key?
Private
SITE
Public
Asymmetric Encryption Algorithm
Asymmetric Encryption Algorithm
Signed Public Key
Asymmetric Encryption Algorithm
Asymmetric Encryption Algorithm
= public key?
Private

Figure 19 - Certificate chaining

3.13.5 Secure Ciphers (Algorithms)

So which algorithms should I use? That is something that will vary over time. Worldwide, the US National Institute of Standards in Technology, or NIST, is recognized as the source of what is acceptable cryptography. Most security standards (including PCI DSS) defer to NIST, as will I.

3.13.6 Summary Table

Primitive	Other Names	Keys	Reversible?	Speed
Symmetric	Private-key	One: shared	Yes	Fast

Asymmetric	Public-key	2: Private and Public	Using other key	Slow
Hash function	One-way function	None	No, only using brute-force	Fast

Table 8 - Summary of cryptographic primitives

Footnotes

[1] PCI Security Standards Council (2015). Payment Card Industry Data Security Standard - Requirements and Security Assessment Procedures - Version 3.1. Retrieved July 13, 2015, from https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-1.pdf.

[2] PCI Security Standards Council (2015, p.5). Payment Card Industry Data Security Standard - Requirements and Security Assessment Procedures - Version 3.1.

[3] Tom on Leadership (2012). How and Why to Benchmark. Retrieved September 22, 2015, from http://tomonleadership.com/2012/12/10/how-and-why-to-benchmark/.

[4] ISACA (2010). The Basics of Internal Controls. Retrieved September 22, 2015, from http://www.theiia.org/chapters/pubdocs/242/Internal_Controls_Basics_IIA_040709.pdf.

[5] International Organization for Standardization / International Electrotechnical Commission (2014). ISO/IEC 27000 - Information technology — Security techniques — Information security management systems. Retrieved September 22, 2015, from http://www.iso.org/iso/catalogue_detail?csnumber=63411

[6] Wikipedia (2011). ITIL. Retrieved September 22, 2015, from https://en.wikipedia.org/wiki/ITIL.

[7] ISACA (2010). COBIT 5: A Business Framework for the Governance and Management of Enterprise IT Retrieved September 22, 2015, from http://www.isaca.org/cobit/

[8] National Institute of Standards and Technology (2007). Special Publications. Retrieved September 23, 2015 from http://csrc.nist.gov/publications/PubsSPs.html

[9] U.S. Department of Health & Human Services (2009). Health Information Privacy. Retrieved September 23, 2015 from http://www.hhs.gov/ocr/privacy/

[10] Wikipedia (2011). Sarbanes–Oxley Act. Retrieved September 23, 2015 from https://en.wikipedia.org/wiki/Sarbanes%E2%80%93Oxley_Act

[11] PCI DSS only applies to cards that bear the logo of one of the 5 founding members of the SSC: Visa, MasterCard, American Express, Discover, JCB.

[12] PCI Security Standards Council (2015, p.8). Payment Card Industry Data Security Standard - Requirements and Security Assessment Procedures - Version 3.1.

[13] U.S. Department of Health & Human Services (2009). Health Information Privacy.

[14] Office of the Privacy Commissioner of Canada (2012). The Personal Information Protection and Electronic Documents Act (PIPEDA). Retrieved September 22, 2015, from https://www.priv.gc.ca/leg_c/leg_c_p_e.asp.

[15] European Commission (2011). Protection of personal data. Retrieved September 22, 2015, from http://ec.europa.eu/justice/data-protection/.

[16] National Institute of Standards and Technology (2010, p.2-1). Special Publication 800-122 Revision 1, Guide to Protecting the Confidentiality of Personally Identifiable Information (PII) . Retrieved September 22,2015, from http://csrc.nist.gov/publications/nistpubs/800-122/sp800-122.pdf.

[17] National Institute of Standards and Technology (2010, p.2-2). Special Publication 800-122 Revision 1, Guide to Protecting the Confidentiality of Personally Identifiable Information (PII) .

[18] Quote Investigator (2011). Everything Should Be Made as Simple as Possible, But Not Simpler. Retrieved September 22, 2015, from http://quoteinvestigator.com/2011/05/13/einstein-simple/.

[19] Wikipedia (2011). Classified information. Retrieved September 22, 2015, from https://en.wikipedia.org/wiki/Classified_information.

[20] Merriam-Webster (2005). Definition of governance. Retrieved September 22, 2015, from http://www.merriam-webster.com/dictionary/governance.

[21] PCI Security Standards Council (2015, p.13). Payment Card Industry Data Security Standard - Requirements and Security Assessment Procedures - Version 3.1.

[22] PCI Security Standards Council (2015, p.5). Payment Card Industry Data Security Standard - Requirements and Security Assessment Procedures - Version 3.1.

[23] Computer Emergency Response Team (CERT) (2014). OCTAVE - Cyber Risk and Resilience Management. Retrieved September 22, 2015, from http://www.cert.org/resilience/products-services/octave/.

[24] International Organization for Standardization / International Electrotechnical Commission (2014). ISO/IEC 27005 - Information security risk management . Retrieved September 22, 2015, from http://www.iso27001security.com/html/27005.html.

[25] National Institute of Standards and Technology (2012). Special Publication 800-30 Revision 1, Guide for Conducting Risk Assessments. Retrieved September 22,2015, from http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf.

[26] PCI Security Standards Council (2015). ROC Reporting Template for v3.1 Retrieved July 8, 2015, from https://www.pcisecuritystandards.org/documents/PCI_DSS_v3_1_ROC_Reporting_Template.pdf.

[27] Open Web Application Security Project (2008). Application Threat Modeling. Retrieved September 22, 2015, from https://www.owasp.org/index.php/Application_Threat_Modeling.

[28] Microsoft Development Network (2005). Security Development Lifecycle. Retrieved September 22, 2015, from https://msdn.microsoft.com/security/default.aspx?pull=/library/en-us/dnsecure/html/sdl.asp.

[29] Microsoft Development Network (2005). Security Development Lifecycle.

[30] PCI Guru (2015). Policies, Standards And Procedures. Retrieved September 22, 2015 from https://pciguru.wordpress.com/2015/07/02/policies-standards-and-procedures/.

[31] PCI Security Standards Council (2015). PCI DSS Designated Entities Supplemental Validation For use with PCI DSS v3.1. Retrieved July 2, 2015, from https://www.pcisecuritystandards.org/documents/PCI_DSS_v3_DESV.pdf.

[32] Designated entities will be formally identified by their acquirers or the card brands, and is likely to include already breached organizations and those that hold substantial amounts of CHD.

[33] PCI Security Standards Council (2015). ROC Reporting Template for v3.1.

[34] PCI Security Standards Council (2015). PCI DSS Designated Entities Supplemental Validation For use with PCI DSS v3.1.

[35] PCI Security Standards Council (2015, p.19). Payment Card Industry Data Security Standard - Requirements and Security Assessment Procedures - Version 3.1.

[36] PCI Security Standards Council (2015, p.25). Payment Card Industry Data Security Standard - Requirements and Security Assessment Procedures - Version 3.1.

[37] Verizon Enterprise Solutions (2015, p.29). 2015 PCI Compliance Report. Retrieved July 1, 2015, from http://www.verizonenterprise.com/pcireport/2015/.

[38]Wikipedia (2011). Long-range Wi-Fi. Retrieved September 22, 2015, from https://en.wikipedia.org/wiki/Long-range_Wi-Fi.

[39] PCI Security Standards Council (2015, p.69). Payment Card Industry Data Security Standard - Requirements and Security Assessment Procedures - Version 3.1.

[40] Center for Internet Security (2003). Retrieved September 22, 2015, from https://www.cisecurity.org/.

[41] ISO - International Organization for Standardization (2015). Retrieved September 22, 2015, from http://www.iso.org/.

[42] SANS Information Security Training (2004). Retrieved September 22, 2015, from https://www.sans.org/.

[43] National Institute of Standards and Technology (2015). ” Retrieved September 22, 2015, from http://www.nist.gov/.

[44] PCI Security Standards Council (2015). FAQ 1224. Public Knowledge Base - What does one function per server mean? Retrieved September 22, 2015, from https://pcissc.secure.force.com/faq/articles/Frequently_Asked_Question/What-does-one-function-per-server-mean/.

[45] PCI Security Standards Council (2015, p.8). PCI DSS Designated Entities Supplemental Validation For use with PCI DSS v3.1.

[46] PCI Security Standards Council (2015, p.8). PCI DSS Designated Entities Supplemental Validation For use with PCI DSS v3.1.

[47] PCI Security Standards Council (2015). FAQ 1154. Public Knowledge Base - Is pre-authorization account data in scope for PCI DSS? Retrieved September 22, 2015 from https://pcissc.secure.force.com/faq/articles/Frequently_Asked_Question/Is-pre-authorization-account-data-in-scope-for-PCI-DSS/.

[48] PCI Security Standards Council (2015, p.42). Payment Card Industry Data Security Standard - Requirements and Security Assessment Procedures - Version 3.1.

[49] PCI Security Standards Council (2015). FAQ 1045. Public Knowledge Base - Is MPLS considered a private or public network when transmitting cardholder data? Retrieved September 22, 2015, from https://pcissc.secure.force.com/faq/articles/Frequently_Asked_Question/Is-MPLS-considered-a-private-or-public-network-when-transmitting-cardholder-data/.

[50] PCI Security Standards Council (2015, p.10). PCI DSS Designated Entities Supplemental Validation For use with PCI DSS v3.1. Retrieved July 2, 2015, from https://www.pcisecuritystandards.org/documents/PCI_DSS_v3_DESV.pdf.

[51] Verizon Enterprise Solutions (2015, p.22). 2015 Data Breach Investigations Report (DBIR). Retrieved July 1, 2015, from http://www.verizonenterprise.com/DBIR/2015/.

[52] Open Web Application Security Project (2008). Category:OWASP Top Ten Project. Retrieved September 22, 2015, from https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology.

[53] Visa Europe (2011).Using the Visa Private BIN Range. Retrieved September 22, 2015, from http://www.visaeurope.com/media/images/12_using_the_visa_private_bin_range_-_best_practice_guide%20110615-73-24720.pdf.

[54] Open Web Application Security Project (2013). Top 10 2013-A1-Injection. Retrieved September 22, 2015, from https://www.owasp.org/index.php/Top_10_2013-A1-Injection.

[55] PCI Security Standards Council (2013).PCI DSS E-commerce Guidelines. Retrieved September 22, 2015, from https://www.pcisecuritystandards.org/pdfs/PCI_DSS_v2_eCommerce_Guidelines.pdf.

[56] Open Web Application Security Project (2013). Top 10 2013-A2-Broken Authentication and Session Management. Retrieved September 22, 2015, from https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Session_Management.

[57] The Daily WTF (2014). The Robot Guys. Retrieved September 22, 2015, from http://thedailywtf.com/articles/the-robot-guys.

[58] Bryan Krebs (2014). All About Skimmers - Krebs on Security. Retrieved July 1, 2015, from http://krebsonsecurity.com/category/all-about-skimmers/.

[59] PCI Guru (2014). Requirement 10.6.2 Clarification Retrieved September 22, 2015, from https://pciguru.wordpress.com/2014/08/08/requirement-10-6-2-clarification/

[60] PCI Security Standards Council (2015). FAQ 1304. Public Knowledge Base - What devices does PCI DSS Requirement 10.6.2 apply to? Retrieved September 22, 2015, from https://pcissc.secure.force.com/faq/articles/Frequently_Asked_Question/What-devices-does-PCI-DSS-Requirement-10-

6-2-apply-to/.

[61] PCI Security Standards Council (2015). FAQ 1317. Public Knowledge Base - What is a *“significant change”* for PCI DSS Requirements 11.2 and 11.3? Retrieved September 22, 2015, from https://pcissc.secure.force.com/faq/articles/Frequently_Asked_Question/What-is-a-significant-change-for-PCI-DSS-Requirements-11-2-and-11-3/.

[62] PCI Guru (2014). Significant Change And Periodic. 22 Sep. 2015 Retrieved September 22, 2015, from https://pciguru.wordpress.com/2014/12/09/significant-change-and-periodic/

[63] PCI Security Standards Council (2015). Approved Scanning Vendors. Retrieved September 22, 2015, from https://www.pcisecuritystandards.org/approved_companies_providers/approved_scanning_vendors.php

[64] PCI Security Standards Council (2015). Approved Scanning Vendors.

[65] PCI Security Standards Council (2014, p.1). Information Supplement: Third-Party Security Assurance. Retrieved July 13, 2015, from https://www.pcisecuritystandards.org/documents/PCI_DSS_V3.0_Third_Party_Security_Assurance.pdf.

[66] PCI Security Standards Council (2015, p.106). Payment Card Industry Data Security Standard - Requirements and Security Assessment Procedures - Version 3.1.

[67] PCI Security Standards Council (2015, p.107). Payment Card Industry Data Security Standard - Requirements and Security Assessment Procedures - Version 3.1.

[68] Verizon Enterprise Solutions (2015, p.59). 2015 PCI Compliance Report.

[69] PCI Security Standards Council (2015). Prioritized Approach for PCI DSS Version 3.1. Retrieved September 22, 2015, from https://www.pcisecuritystandards.org/documents/Prioritized_Approach_for_PCI_DSS_v3-1.pdf.

[70] PCI Security Standards Council (2015). FAQ 1046. Public Knowledge Base - Will the PCI Security Standards Council “approve” my organization's implementation of compensating controls in my effort to comply with the PCI DSS? Retrieved September 22, 2015, from https://pcissc.secure.force.com/faq/articles/Frequently_Asked_Question/Will-the-PCI-Security-Standards-Council-approve-my-organization-s-implementation-of-compensating-controls-in-my-effort-to-comply-with-the-PCI-DSS/.

Table of Contents

Term	Description	Source
AAA	Acronym for “authentication, authorization, and accounting”. Protocol for authenticating a user based on their verifiable identity, authorizing a user based on their user rights, and accounting for a user's consumption of network resources.	PCI
Access Control List	An access control list is a list of permissions attached to an object. In networking, an access control list is a set of permissions allowing or denying network traffic between a source and destination connected to the network.	Author
ACL	Acronym for “Access Control Lists”.	Author
Acquirer	The entity that takes on the financial risk of the merchant transaction (sometimes the acquirer is also a payment processor and the roles are mingled - the volumes distinguish between these functions).	Author
AoC	Acronym for “Attestation of Compliance”. The AOC is a form for merchants and service providers to attest to the results of a PCI DSS assessment, as documented in the Self-Assessment Questionnaire or Report on Compliance.	PCI
APT	Acronym for “Advanced Persistent Threat”. An 'advanced persistent threat'(APT) is a set of stealthy and continuous computer hacking processes, often orchestrated by human(s) targeting a specific entity.'APT'usually targets organizations and/or nations for business or political motives.	Wikipedia
ASV	Acronym for “Approved Scanning Vendor.” Company approved by the PCI SSC to conduct external vulnerability scanning services.	PCI
ATM	Acronym for “Automatic Teller Machine”.	Author
Authorization	In the context of access control, authorization is the granting of access or other rights to a user, program, or process. Authorization defines what an individual or program can do after successful authentication. In the context of a payment card transaction, authorization occurs when a merchant receives transaction approval after the acquirer validates the transaction with the issuer/processor.	PCI
Automatic Teller Machine	An ATM, also known as an Automated Banking Machine (ABM), is an electronic machine that allows a bank cardholder to withdraw cash without the assistance of a cashier.	Author
Bank Identification Number	The first four to six digits of a credit card. The Bank Identification Number (BIN) is often called Institution Identification Number (IIN).	Author
BAU	An Acronym for “business as usual.” BAU is an organization”s normal daily business operations.	PCI
BIN	Acronym for “Bank Identification Number”.	Author
Card brands	The 5 founding members of the PCI SSC that enforced the PCI DSS within the PCI industry, and facilitate the payment and settlement.	Author

Card Production	Card Production is a standard developed and maintained by the PCI SSC that covers the requirements that payment card producers (which can be issuers) must implement.	Author
Card Verification Code or Value	Also known as Card Validation Code or Value, or Card Security Code. Refers to either: (1) magnetic-stripe data, or (2) printed security features. (1) Data element on a card’s magnetic stripe that uses secure cryptographic processes to protect data integrity on the stripe, and reveals any alteration or counterfeiting. Referred to as CAV, CVC, CVV, or CSC depending on payment card brand. The following list provides the terms for each card brand: • CAV - Card Authentication Value (JCB payment cards) • CVC - Card Validation Code (MasterCard payment cards) • CVV - Card Verification Value (Visa and Discover payment cards) • CSC - Card Security Code (American Express) (2) For Discover, JCB, MasterCard, and Visa payment cards, the second type of card verification value or code is the rightmost three-digit value printed in the signature panel area on the back of the card. For American Express payment cards, the code is a four-digit unembossed number printed above the PAN on the face of the payment cards. The code is uniquely associated with each individual piece of plastic and ties the PAN to the plastic. The following list provides the terms for each card brand: • CID - Card Identification Number (American Express and Discover payment cards) • CAV2 - Card Authentication Value 2 (JCB payment cards) • CVC2 - Card Validation Code 2 (MasterCard payment cards) • CVV2 - Card Verification Value 2 (Visa payment cards)	PCI
Card-not-present payment	Card-present refer to transactions where the cardholder (the payer) is not physically in the presence of the merchant (in the store), and 'includes (postal) mail (or even fax) order catalog, a phone-based transaction such as airline ticket reservation or very often an online store.	Author
Card-present payment	Card-present refer to transactions where the cardholder (the payer) is physically in the presence of the merchant (in the store) and uses his payment card to pay.	Author
Cardholder Data	The main data covered by PCI DSS. Consists of the PAN, cardholder name, card expiration date, and sometimes service code. See Sensitive Authentication Data for additional data elements that may be transmitted or processed (but not stored) as part of a payment transaction.	PCI
Cardholder Data Environment	Basically the area (people, process and technologies) we are trying to protect, which starts with the systems that SPT CHD or SAD but is not limited to these.	Author
cardholders	The individual person to whom a payment card is issued and who pays for products or services using that card	Author
CDE	Acronym for “Cardholder Data Environment”.	Author
CHD	Acronym for “Cardholder Data”.	PCI
CISP	Aconym for “Cardholder Information Security Program”. A program	Author

	created by Visa’s in 1999 and that served as the foundation for the PCI DSS.	
Clearing	Clearing is the process of matching (called reconciliation in accounting terms) merchant bank (which is generally the acquirer) and issuer transactions.	Author
Controlled Access	In the context of network segmentation for PCI DSS, the configuration that allows only limited (restricted) communications possible between systems.	Author
Critical Security Controls	SANS top 20 recommended security controls	Author
CSC	Acronym for “Critical Security Controls”.	Author
CVE	The Common Vulnerabilities and Exposures (CVE) system provides a reference-method for publicly known information-security vulnerabilities and exposures. (See zero-day vulnerabilities for the contrary).	Wikipedia
CVV, CVV2	See “Card Verification Code or Value” for more detail.	PCI
DDOS	Acronym for “Distributed Denial of Service” attack. A DDOS attack is a DOS attack where the attack source is more than one-and often thousands-of unique IP addresses.	Wikipedia
DESV	PCI DSS Designated Entities Supplemental Validation for PCI DSS 3.1 (DESV) - A new set of requirements to increase assurance that an organization maintains compliance with PCI DSS over time, and that non-compliance is detected by a continuous (if not automated) audit process; this set of requirements applies to entities designated by the card brands or acquirers that are at a high risk level for the industry.	Author
DLP	Acronym for “Data Loss Prevention”. Data loss prevention (DLP) solution is a system that is designed to detect potential data breach / data ex-filtration transmissions and prevent them by monitoring, detecting and blocking sensitive data while in-use (endpoint actions), in-motion (network traffic), and at-rest (data storage).	Wikipedia
DMZ	Abbreviation for “demilitarized zone.” Physical or logical sub-network that provides an additional layer of security to an organization”s internal private network. The DMZ adds an additional layer of network security between the Internet and an organization”s internal network so that external parties only have direct connections to devices in the DMZ rather than the entire internal network.	PCI
DOS	Acronym for “Denial of Service” attack. A DOS attack is an attempt to make a machine or network resource unavailable to its intended users, such as to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet.	Wikipedia
DR/BC	Acronym for “Disaster Recovery/Business Continuity”. Disaster recovery (DR) involves a set of policies and procedures to enable the recovery or continuation of vital technology infrastructure and systems following a natural or human-induced disaster. Disaster recovery focuses on the IT or technology systems supporting critical business functions,as opposed to business continuity, which involves keeping all essential aspects of a	Wikipedia

	business functioning despite significant disruptive events. Disaster recovery is therefore a subset of business continuity.	
DSS	Acronym for “Data Security Standard”. See PCI DSS.	Author
EMV	Acronym for “Europay MasterCard Visa”. EMV equipped payment cards use a small chip to store cardholder data more securely than a magnetic track. EMV is a technical standard for smart payment cards and for payment terminals and automated teller machines which can accept them.	Wikipedia
Exfiltration	Used by some'computer security'practitioners in place of 'data theft', to mean an unauthorized release of data from within a computer system or network (data or files extracted from borders of a computer operations center [Source: OPM Director Katherine Archuleta Testimony])	Wikipedia
FTP	Acronym for “File Transfer Protocol.” Network protocol used to transfer data from one computer to another through a public network such as the Internet. FTP is widely viewed as an insecure protocol because passwords and file contents are sent unprotected and in clear text. FTP can be implemented securely via SSH or other technology. See S-FTP.	PCI
HIPAA	The Health Insurance Portability and Accountability Act of 1996 (HIPAA) includes requirement for the establishment of national standards for electronic health care transactions and national identifiers for providers, health insurance plans, and employers.	Wikipedia
Host	For virtualization, the system (hardware of software) where the hypervisor runs.	Author
HTTP	Acronym for “hypertext transfer protocol.” Open internet protocol to transfer or convey information on the World Wide Web.	PCI
HTTPS	Acronym for “hypertext transfer protocol over secure socket layer.” Secure HTTP that provides authentication and encrypted communication on the World Wide Web designed for security-sensitive communication such as web-based logins.	PCI
Hypervisor	For virtualization, the application that allows for virtualization of systems	Author
IDS	Acronym for “intrusion-detection system.” Software or hardware used to identify and alert on network or system anomalies or intrusion attempts. Composed of: sensors that generate security events; a console to monitor events and alerts and control the sensors; and a central engine that records events logged by the sensors in a database. Uses system of rules to generate alerts in response to detected security events. See IPS	PCI
IIN	Acronym for “Institution Identification Number”.	Author
Institution Identification Number	The six digits of a payment card as defined in the ISO/IEC 7812 standard.	Author
IPS	Acronym for “intrusion prevention system.” Beyond an IDS, an IPS takes the additional step of blocking the attempted intrusion.	PCI
ISA	Acronym for “Internal Security Assessor.” ISAs are qualified by PCI SSC. ISAs are employees of organizations that help their organizations build	PCI

	their internal PCI Security Standards expertise and strengthen their approach to payment data security, as well as increasing their efficiency in compliance with data security standards.	
ISO	In the context of industry standards and best practices, ISO, better known as “International Organization for Standardization” is a non-governmental organization consisting of a network of the national standards institutes.	PCI
Isolation	In the context of network segmentation for PCI DSS, the configuration that allows no possible access between systems.	Author
Issuer	The entity that issues the card to the cardholder, often (but not limited to) your bank.	Author
IT	Acronym for “Information Technology”. Information technology (IT) is the application of computers and telecommunications equipment to store, retrieve, transmit and manipulate data,[1] often in the context of a business or other enterprise.	Wikipedia
Malware / Malicious Software	Software or firmware designed to infiltrate or damage a computer system without the owner’s knowledge or consent, with the intent of compromising the confidentiality, integrity, or availability of the owner”s data, applications, or operating system. Such software typically enters a network during many business-approved activities, which results in the exploitation of system vulnerabilities. Examples include viruses, worms, Trojans (or Trojan horses), spyware, adware, and rootkits.	PCI
Merchant	The entity who receive payments from cardholders for products or services.	Author
MOTO or MO/TO	Acronym for “Mail-Order/Telephone-Order.”	PCI
NAT	Acronym for “network address translation.” Also known as network masquerading or IP masquerading. Change of an IP address used within one network to a different IP address known within another network, allowing an organization to have internal addresses that are visible internally, and external addresses that are only visible externally.	PCI
NERC	Acronym for “North American Electric Reliability Corporation”. The organization which manages information security standards for electrical energy companies, and the name of the main standard produced.	Author
NFC	Acronym for “Near field communication”. In the payment context, NFC allow payments to be performed simply by placing the payment card with the NFC chip close to the payment reader (no need to swipe the magnetic track or insert the chip).	Author
NIST	Acronym for “National Institute of Standards and Technology.” Non-regulatory federal agency within U.S. Commerce Department’s Technology Administration.	PCI
Organization	In the context of the PCI Resources book volumes, any entity subject to the PCI DSS and that may include, business, non-for-profits.	Author
OSI Network Model	The Open Standards Interconnect (OSI) network model is a conceptual model which consists of 7 layers built on top of each other.	Author

P2PE	Point-to-Point Encryption (P2PE) is a standard developed and maintained by the PCI SSC that allows scope reduction through the use of encrypted transmission on payment terminals where the merchant cannot decrypt the information.	Author
PA-DSS	Acronym for “Payment Application Data Security Standard.” A standard maintained by the PCI SSC that provides controls over an application used in the environment of a organization that stores, processes or transmits cardholder data or sensitive authentication data.	Author
PAN	Acronym for “primary account number” and also referred to as “account number.” Unique payment card number (typically for credit or debit cards) that identifies the issuer and the particular cardholder account.	PCI
Payment processor	The entity that receives payment information from the merchant, authorizes, settles and clears the transaction (can be a bank, but can also be a service provider).	Author
PCI	Acronym for “Payment Card Industry.”	PCI
PCI DSS	Acronym for “Payment Card Industry Data Security Standard.” A standard maintained by the PCI SSC that provides controls over the environment of a organization that stores, processes or transmits cardholder data or sensitive authentication data.	Author
PCI SSC	Acronym for “Payment Card Industry Security Standard Council.” The PCI SSC was formed by the card brands, and manages information security standards to help protect cardholder data.	Author
PFI	Acronym for “PCI Forensics Investigator”. PFIs are qualified by PCI SSC to perform PCI DSS forensic investigations in case of cardholder data breaches.	Author
PIN	Acronym for “personal identification number.” Secret numeric password known only to the user and a system to authenticate the user to the system. The user is only granted access if the PIN the user provided matches the PIN in the system. Typical PINs are used for automated teller machines for cash advance transactions. Another type of PIN is one used in EMV chip cards where the PIN replaces the cardholder”s signature.	PCI
Acronym for “PIN Transaction Security,” PTS is a set of modular evaluation requirements managed by PCI Security Standards Council, for PIN acceptance POI terminals. Please refer to www.pcisecuritystandards.org.	Acronym for “PIN Transaction Security,” PTS is a set of modular evaluation requirements managed by PCI Security Standards Council, for PIN acceptance POI terminals. Please refer to www.pcisecuritystandards.org.	PCI
Ping sweeps	In computing, a ping sweep is a method that can establish a range of IP addresses which map to live hosts.	Wikipedia
POS	Acronym for “point of sale.” Hardware and/or software used to process payment card transactions at merchant locations.	PCI
Primary Account Number	The card number printed on the front of the card.	Author

PWN	Pwn is a slang term derived from the verb own, as meaning to appropriate or to conquer to gain ownership.	Wikipedia
QSA	Acronym for “Qualified Security Assessor.” QSAs are qualified by PCI SSC to perform PCI DSS on-site assessments. Refer to the QSA Qualification Requirements for details about requirements for QSA Companies and Employees.	PCI
QSAC	Acronym for “Qualified Security Assessor Company.” A QSA company is a firm qualified by PCI SSC to perform PCI DSS on-site assessments. See QSA for more information.	Author
RAM-scraper	A type of malware program that grab informations that flows through an electronic device’s memory.	Author
Regular Expressions	A regular expression (abbreviated regex or regexp and sometimes called a rational expression) is a sequence of characters that define a search pattern, mainly for use in pattern matching with strings, or string matching, i.e. “find and replace”-like operations.	Wikipedia
Report on Compliance	Report documenting detailed results from an entity”s PCI DSS assessment.	PCI
RoC	Acronym for “Report on Compliance”.	PCI
S-FTP	Acronym for Secure-FTP. S-FTP has the ability to encrypt authentication information and data files in transit. See FTP.	PCI
SAD	Acronym for “Sensitive Authentication Data”.	PCI
SANS	Acronym for “SysAdmin, Audit, Networking and Security,” an institute that provides computer security training and professional certification. (See www.sans.org.)	PCI
SAQ	Acronym for “Self-Assessment Questionnaire.” Reporting tool used to document self-assessment results from an entity”s PCI DSS assessment.	PCI
Sarbanes Oxley	The Sarbanes-Oxley Act of 2002, is a United States federal law that set new or expanded requirements for all U.S. public company boards, management and public accounting firms.	Wikipedia
Sensitive Authentication Data	Includes the magnetric track information, the PIN or PIN block, as well as the Card-not-present authorization value which we will refer to as CVV2 but can take any of the following acronyms: CAV2/CVC2/CVV2/CID.	Author
Service provider	An entity that performs some functions regarding to the payment process and/or provides services that may affect the security of the cardholder data.	Author
Settlement	Payment of the outstanding balance owed by the issuer to the acquirer, and later the merchant.	Author
SIEM	Security information and event management (SIEM) is a term for software products and services combining security information management (SIM) and security event management (SEM).	Wikipedia

SIN	A social insurance number (SIN) is a number issued in Canada to administer various government programs, including in the administration of the Canada Pension Plan and Canada’s varied employment insurance programs, and for tax reporting purposes.	Wikipedia
SOX	Acronym for “Sarbanes Oxley”.	Author
SPT	An Acronym for “Store, Process, or Transmit”, meaning that a system or process comes into contact with CHD and/or SAD and is therefore automatically in scope for PCI DSS.	Author
SQL Injection	Form of attack on database-driven web site. A malicious individual executes unauthorized SQL commands by taking advantage of insecure code on a system connected to the Internet. SQL injection attacks are used to steal information from a database from which the data would normally not be available and/or to gain access to an organization's host computers through the computer that is hosting the database.	PCI
SSL	Acronym for “Secure Sockets Layer.” Industry standard that encrypts the channel between a web browser and web server. Now superseded by TLS. See TLS.	PCI
SSN	In the United States, a Social Security number (SSN) is a nine-digit number issued to U.S. citizens, permanent residents, and temporary (working) residents.	Wikipedia
Third-Party Service Providers	In the context of the PCI Resources book volumes, any entity subject to the PCI DSS and that may include, business, non-for-profits.	Author
TLS	Acronym for “Transport Layer Security.” Designed with goal of providing data secrecy and data integrity between two communicating applications. TLS is successor of SSL.	PCI
TPSP	Acronym for “Third-Party Service Providers”.	Author
Virtual machine	The individual “abstract” system that runs on an hypervisor	Author
VM	Acronym for “virtual machine”. a VM is an emulation of a particular computer system.	Wikipedia
Zero-day vulnerabilities	A zero-day (also known as zero-hour or 0-day) vulnerability is an undisclosed and uncorrected computer application vulnerability that could be exploited to adversely affect the computer programs, data, additional computers or a network. It is known as a “zero-day” because once a flaw becomes known, the programmer or developer has zero days to fix it.	Wikipedia

The column “source” identified the origin of the terms in this glossary.

- “Author” refers to terms defined by the author.
- “PCI” refers to definitions adapted from the PCI SSC documents, mainly the PCI DSS Glossary (https://www.pcisecuritystandards.org/documents/PCI_DSS_Glossary_v3-1.pdf).
- “Wikipedia” refers to definitions adapted from the wikipedia website.