A PCI Threat Model (Alpha Version)

Adam Shostack

Drafted March 2015

Released September 2020

Summary

This document is a draft of a threat model, reverse engineered from the PCI Requirements and Security Assessment Procedures, version 3.0 of November 2013. It also presents an analysis of PCI based on details made visible by the modeling, and discusses some of the ways in which the modeling makes clear how PCI covers technical and organizational factors in a way that might be charitably described as "an opportunity for improvement."

MOTIVATION

I. Why reverse engineer PCI to develop a threat model? Because the PCI requirements are hard to parse and reason about, exacerbating tensions between assessors and those assessed, amplifying the frustration and miscommunication between parties, and making it harder to secure systems and payment card data.

II. This document was created in 2015, and lay fallow until Anton Chuvakin published "Data Security and Threat Models," where he said that we should publish threat models, and reminded me that this wasn't helping anyone hidden on my local hard drive. Anton provided useful comments on that draft before publication.

APPROACH

Starting with a naïve model of a card data processing system, I used requirements from PCI to refine the model. As I progressed through the model, it became clear that PCI freely intermixes what might be termed "technical" and "process" requirements.[1] (Technical requirements are those, such as 1.2.1, 'restrict traffic', where process requirements are more like 1.2.1, 'Establish a formal approval process for network connections.' As this model is being reverse engineered, the distinctions are sometimes unclear.) Thus, there are two main parts of the model presented, a technical model and a process model. They are different for our purposes because the problems and mitigations are different. Problems with a failure to restrict traffic are noticeable by, and possibly exploitable by an attacker. Problems with a process come from inside: an attacker isn't going to try to bypass your network connection process.

Having created a main model, I then create a table of element, threat, PCI requirement. The threats used are appropriate for this exercise. In particular, the threats to a process are not the same as the threats to technical systems, and the mitigations are also different. If an operating system will accept a 4-character password,

[1] This may be a fine approach, but it made the extraction of a threat model much more complex because the first step was to deduce what sort of a problem the requirement addressed.

that's a technical issue, and if a VP signed off on it, that's a separate process issue. In this document, I do not create a high-quality taxonomy of threats, as doing so is an expensive and time-consuming endeavor.

A Model of the Threatened Systems

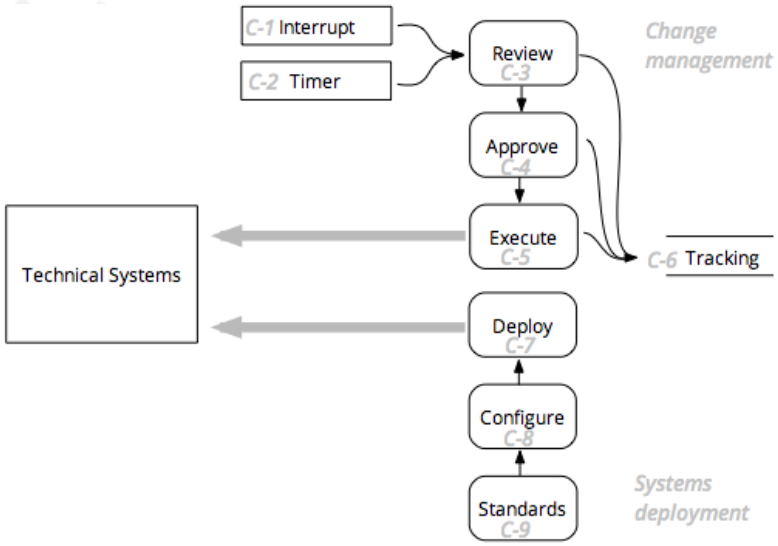

FIGURE 1: THE PROCESS MODEL

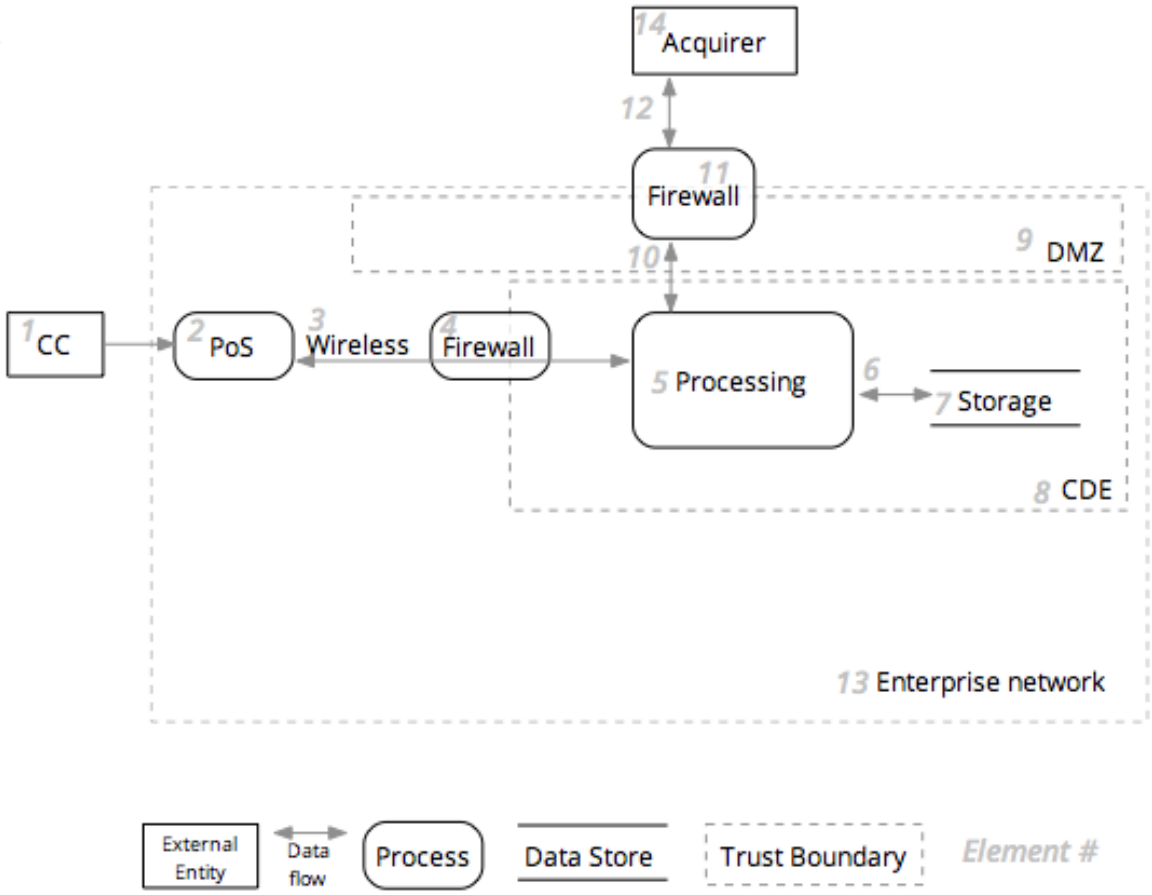

FIGURE 2: THE TECHNICAL SYSTEMS MODEL

Threats and PCI Mitigations

These threats are arranged by diagram element, not by PCI requirement. This choice is made to help break away from a PCI-centric mindset, and to expose possible gaps or overlaps. To assure completeness, whenever a PCI requirement does not map cleanly to the system or process model, those models are updated before moving on.

THREATS ADDRESSED BY CHANGE MANAGEMENT OR SYSTEMS DEPLOYMENT

Note that this threat model is based on evidence (a lack of controls) that PCI treats change management and systems deployment as if all actors within the process are well intentioned, and as such threats to the data flows are not considered.

A few words on notation in the table:

- The initial in the brackets is a categorization, either [P]revent/[D]etect/[Respond], [I]nformation disclosure, or [C]omply, or [M]anage. (The initial is a category of hypothesis that the PCI control should address.)

  - Some issues are marked '2nd order' to indicate that there is at least successful one prior action, such as the theft of encrypted data.
- PCI uses a letter at the end of a testing procedure to distinguish it from requirements when there's not a one-to-one correlation. Requirements are addressed at the leaf' level, thus there is no entry for 3 or 3.5, but there is for 3.5.1
- Entries with a * have requirements/test processes on multiple elements of the diagrams.

Element	Threat/Problem/Vulnerability/Notes[2]	PCI Requirement/ Test Process
C-1 Interrupt	Firewall rulesets go out of date [M]	1.1.7
C-2 Timer	Old data not deleted/not deleted securely	3.1* bullet 2
	Old data not deleted/not deleted securely	3.1* bullet 4
	Malware might be published for new systems	5.1.2
	Anti-malware sw goes out of date	5.2
C-3 Review	[3]	1.1
C-4 Approval		
	Firewall config fails to prevent [C]	1.1.1.a
	Anti-virus fails to protect [P, D, R]	5.1.1
C-5 Execution		
C-6 Tracking & documentation		
	Approval process failures (network connections) [M]	1.1.1.b
	Approval process failures (firewall) [M]	1.1.1.c
	No network diagram [M]	1.1.2, 1.1.3
	Network diagram is not current [M]	1.1.2.a

[2] If this list were created by the PCI Council, then we might reasonably ask for more crisp categorization, but imposing such categories onto a reverse-engineered list does not obviously add value.
[3] Some rows have no threat because there is no single control which clearly maps to it.

	Change not well managed [M]	1.1.5
	Changes not tracked [M]	1.1.6
	Firewall arbitrary changes [M]	1.5
	Systems dropped from management scope [m]	2.4
	Policy/activity drift [m]	2.5
	People with crypto keys might not understand their roles [I,T]	3.6.8
	Policies and procedures ignored [M]	3.7
C-7 Deployment	Unmanaged change [M]	1.2.2
	Default passwords [S]	2.1
	Default wireless configs [S, EoP]	2.1.1
	Malware might be on systems [P, D, R]	5.1
C-8 Configuration		1.2.2
	Configuration errors lead to insecurity [P]	2.2.4
	Unneeded services lead to insecurity [P]	2.2.5
	Keys overly exposed [P] (Not on 7/8 because keys can be in many places)	3.5.1, 3.5.2, 3.5.3
C-9 Standards & Policies	Products ship insecure by default [P, T, EoP]	1.3.6, 2.2
	Insecurity leads to horizontal movement [P, EoP]	2.2.1
	Unneeded services lead to insecurity [P]	2.2.2
	Unencrypted comms lead to network problems [P, T,I]	2.2.3
	Insecurity leads to horizontal movement at hosting provider [P]	2.6
	Data is kept too long	3.1*
	Crypto keys are bad [I, 2nd order]	3.6.1

	Crypto keys are disclosed after generation [I, 2nd order]	3.62*, 3.6.3*
	Crypto keys are used after disclosure [I "2nd order"]	3.6.4
	Compromised keys can't be replaced [I, "2nd order"]	3.6.5
	People might see the crypto keys, remember them, and abuse them [I, "2nd order"]	3.6.6
	Crypto keys might be substituted by attackers [T]	3.6.7
	PANs might be sent via unencrypted channels [I]	4.2
	People might not realize that PANs are sensitive, and send them without encryption [I]	4.3
	People might not realize malware is a threat [P, D, R]	5.4

THREATS ADDRESSED BY TECHNICAL SYSTEMS

Element	Threat/Vulnerability/Problem/Notes	PCI Requirement
1 Credit Card	[4]	
2 Point of Sale		
3 Wireless network & data flows	Eavesdropping [P, I]	4.1*
4 Wireless firewall	[P]	1.2, 1.2.1, 1.2.3,
5 Processing	Full PAN displayed [I]	3.3
6 Processing to storage (Data flows)		
7 Storage	Authentication data stored [I] [5]	3.2
	Full track data stored [I]	3.2.1
	Card verification code/values stored [i]	3.2.2

[4] Again, some rows have no threat because there is no single control which clearly maps to it.
[5] Properly, a threat might be that the data is *disclosed*, but the *storage* of the data is in violation of the requirement.

	PIN block/encrypted PIN block stored [i]	3.2.3
	PAN disclosed [i]	3.4
	Account access allows file access [i]	3.4.1
	Crypto keys stored securely [I, 2nd order]	3.6.3
8 Card Data Environment		
9 DMZ		
10 Processing to DMZ (Data flows)	Eavesdropping [P, I]	4.1*
11 Internet firewall	[P]	1.1.4, 1.2, 1.3, 1.3.2, 1.3.3*, 1.3.4
12 DMZ to Acquirer (data flows)	Eavesdropping [P, I]	4.1*
13 Enterprise network	Eavesdropping [P, I]	4.1*
14 Acquirer		
15 DMZ/Internal FW	[P]	1.1.4, 1.2.1, 1.3, 1.3.1, 1.3.3*, 1.3.5
16 Employee/Mobile systems	[P:eop] (firewall)	1.4*
	[P:tamper] Users change their devices	1.4*, 5.3
	[P:DoS] software is not running	1.4*
17 Mgmt Console	*Included to support 2.3, but see element 18*	
18 Mgmt console to CDE (data flows)	Info disclosure	2.3
	Keys disclosed after generation [I, 2nd order]	3.6.2
19 Logs	PAN stored in logs [i]	3.4.d