Global Information Assurance Certification Paper

Implementing Visa's Cardholder Information Security Program

GIAC Security Essentials
Certification (GSEC)
Practical Assignment
Version 1.4b

Option 1 - Research on Topics
in Information Security

Submitted by: Vic Ricker

Table of Contents

Abstract

In June of 2001, Visa mandated their Cardholder Information Security Program (CISP). The CISP is comprised of twelve requirements with which members must comply to show due diligence in protecting sensitive cardholder data. The requirements cover firewalls, security patches, encryption, anti-virus, physical security, etc. Each of these requirements will be discussed and some general implementation ideas, focusing on open source solutions, will be presented.

Introduction

The problems of credit card fraud and identity theft have grown significantly over the past few years. "The Department of Justice now estimates that as many as 700,000 people are victims of identity theft each year."[1] In an effort to curb losses and protect consumers, credit card companies are mandating security programs for their merchants and service providers.

Visa's Cardholder Information Security Program[2] is comprised of twelve basic security requirements[3] (from their web site):

1. Install and maintain a working firewall to protect data
2. Keep security patches up-to-date
3. Protect stored data
4. Encrypt data sent across public networks
5. Use and regularly update anti-virus software
6. Restrict access by "need to know"
7. Assign unique ID to each person with computer access
8. Don't use vendor-supplied defaults for passwords and security parameters
9. Track all access to data by unique ID
10. Regularly test security systems and processes
11. Implement and maintain an information security policy
12. Restrict physical access to data

The "Training and Tools" section[4] of Visa's CISP web site[5] has many of the resources needed to understand the requirements and to verify compliance. Visa puts service providers and merchants into three groups based mainly on the number of credit transactions processed.[6] The lower levels are only required to perform a perimeter scan and to complete a Self Assessment Questionnaire while the higher levels are subject to stricter requirements.

The Self Assessment Questionnaires are made up of questions which assess the level of compliance with recommended "Best Practices" and "Visa Requirements". All questions labeled as "Requirements" must be answered, "yes" for the entity to be considered compliant. "No" answers to questions that are labeled, "Best Practices" do not affect compliance, but indicate potential security problems that should be addressed. Along with the Self Assessment, a system perimeter scan must be submitted to Visa (for Service Providers) or the Acquirer (for Merchants).

The higher levels, (Level 1 Merchant and Levels 1 and 2 Service Provider) are required to have compliance assessed by a Visa approved assessor. The

assessor will use Visa's “Security Audit Procedures and Reporting” document[7] to prepare a “Report on Compliance” which must be submitted securely to Visa, the Merchant or Service provider, and the Acquirer (for Merchant assessments).

Visa's Requirements

1. Install and maintain a working firewall to protect data

Using firewalls to secure the network perimeter is a good first step in protecting a network. A common misconception, however, is that firewalls alone will provide adequate security. AirMagnet's Best Practices for Securing Your WLAN explains that,

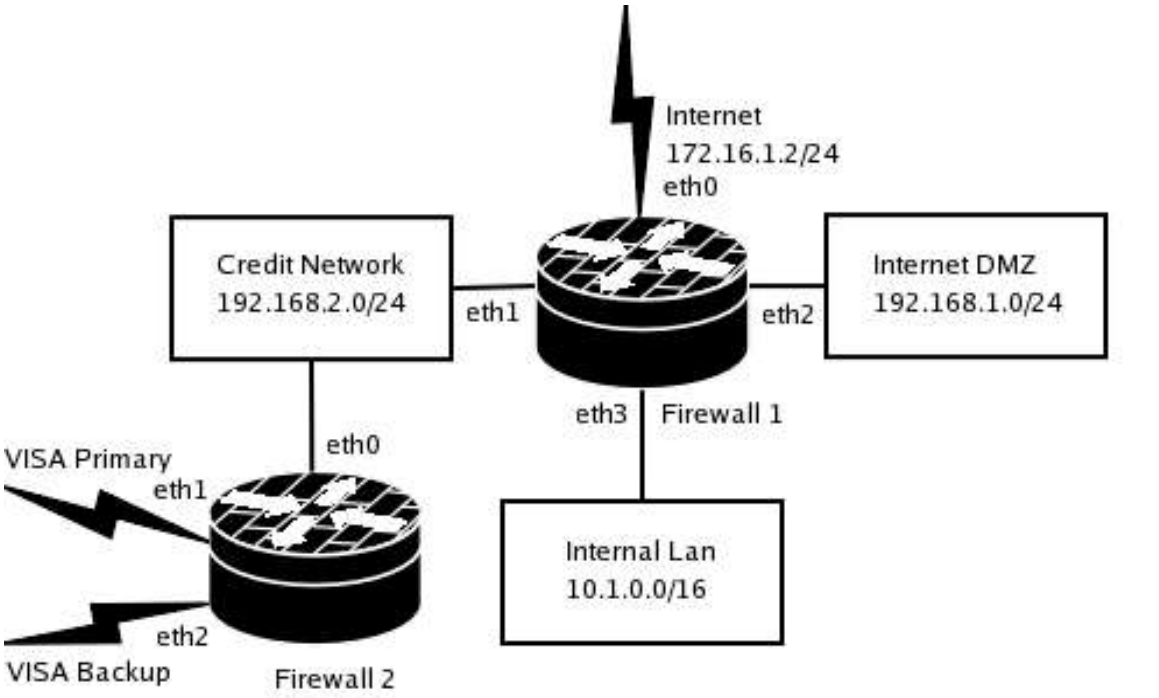
> *"A secure network is the result of an ongoing security process and not simply the installation of security technology. This means that even with a strong security strategy in place, IT must also actively monitor and enforce compliance with that policy, and be aware of the vulnerabilities inherent in the strategy they have chosen."*[8]

Visa requires that a firewall policy be in place. The firewall policy should require the following items:

- a network diagram that shows the relationships between all network trust zones
- firewalls are in place to protect the zones of trust
- documentation of the business justification for allowed services
- periodic review of firewall and router configurations
- written approval for all network connections and firewall rule changes
- a modem use policy
- configuration standards for all network devices

Network Diagram

The diagram in Figure 1 shows a basic network. Firewalls separate each of the zones of trust: the Internet DMZ containing public facing web servers, mail servers, etc., the Internal LAN, the Credit Network containing the systems that store and process credit card data, and another firewall between the credit network and external entities such as credit authorizers.



High Level Credit Network Diagram

Figure 1 - High Level Credit Network Diagram

Allowed Services

Services which are allowed through the firewall need to be documented. Port numbers, protocols, and a descriptions of the services must be recorded. Services other than HTTP, HTTPS (SSL), SSH (Secure Shell), and VPN (Virtual Private Network) must also have a justified business purpose. The reason behind this is simple. The goal is to reduce the number of services that are exposed to potential attackers.

Periodic Review

Rules and configurations should be reviewed and tested periodically to confirm that proper security controls are being maintained. Discrepancies can be detected by comparing change control documentation against running configurations. Having another person review rule sets will help discover mistakes.

Management Approval

Firewall and Network Administrators should not be responsible for approving configuration changes or external network connections. Policies should be in place that assign proper roles and responsibilities, for example, that Management

approval is required before any changes may be made. Procedures should be created for requesting network and security configuration changes, as well as the procedures for actually making those changes.

Firewall Configuration

Without a proper configuration, a firewall typically acts as a router, giving no protection at all. Worse yet, an improper configuration will give a false sense of security. The configuration below gives examples of how to implement some of Visa's requirements using Netfilter, under Linux, on Firewall #1 from the network diagram in Figure 1. Netfilter is a stateful firewall. This means that the kernel keeps track of connections made through it, and is able to handle them correctly. For example, if a user makes an FTP connection through the firewall, the firewall follows the 3-way TCP handshake for the command channel and all related packets for that connection will be allowed to pass. If an attacker sends malformed packets to port 21, they will be stopped because they are not part of an established connection in the kernel's state table. A traditional packet filtering firewall might let such an attack, with out of state packets, through. Another advantage of statefullness is that the firewall can be configured to allow “related” connections through automatically, without specific rules specifying port numbers. FTP for example, creates a data connection when transferring files or listing directories. It uses an ephemeral port number which is communicated via the control connection. Netfilter's FTP conntrack module handles inspection of the FTP communication and automatically allows the related data connections through the firewall.

The first step in the firewall configuration is to set up symbolic names for things that may change. This makes the configuration easier to maintain.

```
IPTABLES=/sbin/iptables
SYSCTL=/sbin/sysctl

external_if=eth0
credit_if=eth1
dmz_if=eth2
lan_if=eth3
lan_hide_nat=172.16.1.2

private_www=192.168.1.3
public_www=172.16.1.3

private_dns=192.168.1.4
public_dns=172.16.1.4

credit_server=192.168.2.2
```

```
credit_database=192.168.2.3

credit_service=20020

dan=10.1.3.5
ben=10.1.1.152
vic=10.1.3.138

administrator=$vic
credit_admin=$ben
credit_dba=$dan
```

Disable packet forwarding.

```
$SYSCTL -w net.ipv4.ip_forward=0
```

Set the default policy to not allow any traffic. Connections will have to be explicitly allowed by subsequent rules.

```
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP
```

Create new chains for TCP and UDP rules.

```
$IPTABLES -t filter -N tcprules
$IPTABLES -t filter -N udprules
```

If the firewall receives TCP packets in the NEW state, but the SYN flag is not set, drop them. It is wise to log all dropped packets. It is important to remember to put the logging rule before the drop rule or it will never be executed because the DROP target ends the packet's life in the chain.

```
$IPTABLES -t filter -A tcprules -p tcp ! --syn -m state
--state NEW -j LOG
$IPTABLES -t filter -A tcprules -p tcp ! --syn -m state
--state NEW -j DROP
```

The following rules are the workhorses of the configuration. They allow packets associated with established connections and any related connections. Established connections are ones that have been previously allowed and exist in the state table. Related connections are connections related to connections that are already established. An example would be the FTP data connection mentioned earlier.

```
$IPTABLES -t filter -A INPUT -m state --state
ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -t filter -A FORWARD -m state --state
ESTABLISHED,RELATED -j ACCEPT
```

So far, there are no rules that will actually allow connections to be established. A good place to start is to allow administrative access via the SSH protocol. This rule will allow access to the SSH port to anyone connecting via the LAN interface $lan_if.

```
$IPTABLES -t filter -A tcprules -p tcp -in-interface
$lan_if --destination-port 22 -m state --state NEW -j
ACCEPT
```

To be a bit more secure, it can be limited further to only allow the administrator's workstation. (--src $administrator)

```
$IPTABLES -t filter -A tcprules -p tcp --source
$administrator --destination-port 22 -m state --state
NEW -j ACCEPT
```

Since the internal LAN and the DMZ both use RFC 1918[9] private address blocks, they do not route on the Internet. Network Address Translation (NAT) needs to be used to “hide” behind the public IP addresses on the firewall.

It's wise to limit access to machines in the DMZ. By only NATting certain ports, exposure is limited to only those ports. The following rule only sends port 80 traffic in to the web server by translating the $public_www address to the $private_www address. It's also possible to have the web server running on a different port, 8080 for example and to NAT from port 80 to 8080 by adding “:8080” to $private_www in the –to-destination target.

```
$IPTABLES -t nat -A PREROUTING -p tcp --destination
$public_www –destination-port 80 -j DNAT --to-
destination $private_www
```

This rule alone, however is not enough. Web traffic still needs to be allowed.

```
$IPTABLES -t filter -A tcprules -p tcp --destination
$public_www --destination-port 80 -m state --state NEW
-j ACCEPT
```

The udprules chain has not been addressed yet. As will be shown later, TCP and

UDP traffic will be segregated into the tcprules and udprules chains. DNS is a common service that uses UDP as well as TCP. The following rules allow access to a DNS server.

```
$IPTABLES -t filter -A tcprules --destination
$public_dns --destination-port 53 -m state --state NEW
-j ACCEPT
$IPTABLES -t filter -A udprules --destination
$public_dns --destination-port 53 -m state --state NEW
-j ACCEPT
```

NATting LAN traffic is similar to what was done for the web server except in the reverse direction. Anything heading out to the Internet from the LAN should be hidden behind the firewall's public address. This causes the firewall to re-write the source address with the $lan_hide_nat address. Traffic coming into that address will automatically be translated back to the proper LAN address, assuming that it's a valid connection in the state table.

```
$IPTABLES -t nat -A POSTROUTING –out-interface
$external_if --source 10.1.0.0/16 -j SNAT --to-source
$lan_hide_nat
```

Again, this isn't enough to allow the LAN to access the Internet. Rules have to be created to allow traffic to flow. Often, organizations do not do egress filtering[1]. This makes it less of a hassle to maintain but is not very secure. There are a number of trojans that, upon infecting a computer behind a firewall, attempt to make outward connections to allow attackers past most firewalls. By only allowing a small set of services through the firewall, that risk is diminished. For example, the following rules will allow HTTP and HTTPS through from the LAN.

```
$IPTABLES -t filter -A tcprules -p tcp –in-interface
$lan_if --destination-port 80 -m state --state NEW -j
ACCEPT
$IPTABLES -t filter -A tcprules -p tcp –in-interface
$lan_if --destination-port 443 -m state --state NEW -j
ACCEPT
```

The credit network has to be the most secure. So far, nothing may pass in our out. Similar rules to those presented above can be implemented for the credit network. For example, some type of service will need to be allowed from the DMZ for passing credit transactions from the web server to the credit network. Also, access from the LAN will be needed for the system administrators and other personnel to

1 Egress filtering is filtering outbound traffic to prevent connections on unauthorized ports.

do their jobs.

```
$IPTABLES -t filter -A tcprules -p tcp --source
$www_private --destination $credit_server --
destination-port $credit_service -m state --state NEW
-j ACCEPT
$IPTABLES -t filter -A tcprules -p tcp --source
$credit_admin --destination $credit_server --
destination-port 22 -m state --state NEW -j ACCEPT
$IPTABLES -t filter -A tcprules -p tcp --source
$credit_dba --destination $credit_database --
destination-port 22 -m state --state NEW -j ACCEPT
```

All of the rules so far have been added to the tcprules chain. Before those rules will work, they have to be added to the INPUT, OUTPUT, and FORWARD chains. Packets flow through the chains based on the direction they are flowing. INPUT and OUTPUT are fairly obvious. The FORWARD chain deals with packets that are being routed.

```
$IPTABLES -t filter -A INPUT -p tcp -j tcprules
$IPTABLES -t filter -A INPUT -p udp -j udprules
$IPTABLES -t filter -A OUTPUT -p tcp -j tcprules
$IPTABLES -t filter -A OUTPUT -p udp -j udprules
$IPTABLES -t filter -A FORWARD -p tcp -j tcprules
$IPTABLES -t filter -A FORWARD -p udp -j udprules
```

Before the firewall will route, ip_forwarding has to be re-enabled.

```
$SYSCTL -w net.ipv4.ip_forward=1
```

There are other things that can be turned on with sysctl to secure the firewall further. For example, rp_filter helps stop spoofed packets (the source address is set to an internal address to try to fool the firewall). accept_source_route can be disabled to not allow source routed packets (the sender specifies the route that should be taken). The Iptables Tutorial[11] is a great resource for learning to use iptables.

2. Keep security patches up-to-date

One of the biggest challenges of security is keeping software patches up-to-date. Visa requires that they be installed within 30 days of release. At first, this may seem unreasonable but the gap is closing between the time a vulnerability is published and the time that exploit code is released. Often, security advisories are accompanied by harmless exploit code. An attacker can easily modify this

example code to add a malicious payload.

The first problem is knowing what software is vulnerable and what patches are available. Then, the patches have to be fetched and installed, along with any dependencies. Under Windows, there are many solutions. Windows Update comes standard with the Windows operating system. It works acceptably for a small number of systems. For a larger number of systems, however, enterprises typically need something more powerful. For this, Microsoft has SUS (Software Update Services) and their upcoming version, renamed WUS (Windows Update Services) which can be used to keep the Windows operating system and other Microsoft software patched.[10] Other commercial products such as Patchlink[12] claim to support a wider selection of software and platforms including Solaris and Linux.

Unfortunately, it is impossible for any product to be able to patch every piece of software so it is important to stay current on the different vulnerabilities as they arise. There are many free resources for being notified about security vulnerabilities.

Security Focus has a number of free mail lists available.[13] BugTraq for example, is a popular mailing list where new vulnerabilities are published and discussed. The other lists that are available are geared to more specific topics such as Microsoft, Linux, Sun, etc.

US-CERT[14] the United States Computer Emergency Readiness Team also publishes technical and non-technical Cyber security alerts, bulletins, and security tips.

3. Protect stored data

To limit exposure, credit card data should only be stored as long as there is a business need. Policies for retention, disposal, and audit should be created detailing how long data will be retained, how it should be removed, and how to verify that data is not being retained over the retention period.

Passwords and encryption keys should be stored using strong encryption. For example, Linux uses the MD5 one-way hashing algorithm to protect passwords. Cisco network devices can be configured to use MD5 hashes for the enable password with the 'enable secret' command.[15]

Database encryption should be enabled for stored credit card data. While it is possible to encrypt entire databases or even the entire file system that they are located on, for performance reasons, it is best to only encrypt the columns that need to be protected e.g. the credit card number.[16]

Backups should be encrypted on tape as well as when being transferred across the network to the backup server. Computer Associates' BrightStore/Arcserve uses triple-DES to encrypt data stored on tape.

4. Encrypt data sent across public networks

Encryption is an important tool in protecting sensitive data from being seen by anyone but the intended recipient. Authentication, integrity, and non-repudiation are other features commonly associated with encryption.[17]

Authentication means being able to verify the sender of a document. Integrity is the ability to detect if a document has been modified by someone other than the sender. Non-repudiation means that the sender of a message cannot deny having sent it.

When customers visit e-commerce sites, they expect that their transactions are secure. Web sites use SSL encryption not only to encrypt communication with the web browser but to also verify their identity. They do this by having their web site certificates signed by trusted third parties such as GeoTrust, Thawte, and Verisign.

Configuring a web server like Apache 2 for SSL is fairly easy. Earlier versions did not come with SSL support because of export restrictions. Apache 2, however, has it built in. In testing, it is possible to create a self-signed certificate without having to go through a third party certificate authority (CA). For a production e-commerce server, a signed certificate is necessary. The first step is to create a server key pair. The server key is used to create a certificate signing request (CSR) which is submitted when a certificate is to be purchased. The CA will return a signed certificate which must be configured into the web server. The secure site is configured as another virtual host listening on port 443. The SSLCertificateFile directive tells the web server where the web server certificate is located. The SSLCertificateKeyFile directive tells the server were the server key is located.[18]

Secure Shell (SSH) encrypts management access to a system. Support comes standard with many systems. Out of the box, ssh uses password authentication. A more secure method of authentication is to use public keys. The advantage to using public keys is that the user's password is never sent across the network, encrypted or otherwise.

Key pair generation is typically application dependent. PuTTY, an open source ssh client for Windows has a key generator called PuTTYgen. Open SSH's key generator is called ssh-keygen. Each will generate a public/private key pair. The public key can be placed on a remote server as $HOME/.ssh/authorized_keys. Anyone with the private key from that pair will be able to login to that system. It is very important that the private key be kept secure. It is common practice to encrypt the private key with a strong pass phrase.

Another benefit of SSH is that it can be used to tunnel other protocols. X11, which is insecure to use over public networks, is commonly tunneled over SSH. Other services can be tunneled over ssh such as the POP3 and SMTP mail protocols. Under Linux, this is done by using the command:

```
ssh -L 110:localhost:110 -L 25:localhost:25 mailserver
```

This command tells ssh to listen on the POP3 port (110) and the SMTP port (25) of

the localhost and forward the traffic over an encrypted tunnel to the ssh server on the remote host 'mailserver'. The remote ssh server forwards those connections to its ports 110 and 25. Once the tunnel is setup, the mail client software has to be configured to use the tunnel by telling it to connect to localhost instead of directly to mailserver.

Visa requires email encryption. Unfortunately, the previous method is not what they have in mind. It only encrypts the traffic between the client and the mail server. The encryption does not last once the traffic leaves the tunnel. A more appropriate solution is to use public key encryption. S/MIME and PGP are common methods used to encrypt email. The main difference between S/MIME and PGP is the method used to sign keys. S/MIME uses keys signed by a trusted Certificate Authority (CA). PGP is a bit more complex in that users sign each other's keys. They call it the “web of trust”, where a key is trusted if it is signed by a trusted friend.

For public key encryption to work, users need to agree on an encryption method to use and they must exchange public keys. A message encrypted with a user's public key can only be decrypted with the matching private key. The benefits to using keys do not stop there. Messages can also be signed with a private key. The public key can be used to verify the authenticity of the signature. Not only does this prove who sent the message, it can also be used to detect if the message was modified or forged by a third party.

It is extremely important that private keys be safeguarded. Encrypt them with strong pass phrases and store them securely with file permissions that restrict access by other users.

5. Use and regularly update anti-virus software

Viruses and other types of malware (trojans, worms, etc.) have grown from minor annoyances to major threats. Recent trends in malware payloads include spyware, key loggers, and backdoors.

Spyware typically collects and reports information on the victim's web browsing habits, often without their knowledge. Spyware is often accompanied by software that generates pop-ups or hijacks the user's home page with advertising.

Key loggers collect any information entered by the victim, including passwords and personal information and transmits it to the attacker, usually via another compromised computer. There was case about two years ago where an individual was caught hiding keyloggers on public computers in Kinko's stores around Manhattan. He had used the keyloggers to collect banking accounts and passwords from unsuspecting users. He would then use that information to open other accounts in the names of his victims and use the newly created accounts to siphon money from his victims' legitimate accounts.[19]

Backdoors allow a attackers to take control of the victim's computer at a later time. Armies of these zombie computers are often used to relay spam, perform

distributed denial of service attacks, and other nefarious activities. Often, these backdoors are able to circumvent firewalls that do not perform egress filtering by making outbound connections to IRC (Internet Relay Chat) servers where they sit, waiting for someone to control them. These IRC bot type backdoors can be fairly sophisticated, giving the attacker the ability to execute commands, install software, and retrieve files from the compromised system with the added benefit of giving them access to the network behind the firewall.

> *"Many virus writers are no longer satisfied with ego boosts from causing problems with malicious code-they want to make money. For example, a version of the BugBear virus surfaced last June that targeted about 1,200 financial institutions around the globe. When the virus penetrated a bank, it attempted to install a backdoor to allow intruders access. BugBear reportedly infected hundreds of thousands of systems."* [20]

To protect against such attacks, it is important to install multiple levels of "virus" protection. Typical configurations include one or more scanners at the mail gateway, desktop, and web proxy servers. Diversification of vendors provides a better probability of having signatures for brand new viruses and improved detection rates.

Signature updates

Virus scanners work best when their signatures are kept up to date. Signature updates should be scheduled as frequently as practical. Enterprise class anti-virus solutions can be used to establish and maintain a policy of signature distribution and scheduled scans across all computers in an organization.

Unfortunately, there is no silver bullet that will stop every piece of malware. Educating users about the potential dangers can be an excellent proactive measure.

6. Restrict access by "need to know"

Visa requires that access to systems which hold credit card data must be limited by job necessity. Access restrictions can be created at multiple levels. User groups based on job roles should be created at the firewall, operating system, database, and application levels.

Access to specific servers and services can be granted to groups of users at the firewall level. At the OS and application levels, rights can be granted to roles so that all users within each role get the same level of access necessary to do their jobs.

7. Assign unique ID to each person with computer access

User access to systems with credit card data must be strictly controlled. Users need to be uniquely identifiable for access control as well as for accountability. If a shared account is used for an attack, it is very difficult, if not impossible to hold anyone accountable.

Strong password controls for user accounts must be configured wherever possible. Strong password must be difficult to guess or crack in a short period of time. Visa requires that passwords must be at least seven characters long, made up of alpha and numeric characters, and changed at least every ninety days. Also, they may not be the same as any of the previous four passwords used. These types of controls can typically be set by the operating system's security policy.

Users accounts must be audited every ninety days. Inactive accounts must be removed. User access could be tied into an HR system so that accounts are removed immediately when employees are terminated.

8. Don't use vendor-supplied defaults for passwords and security parameters

Default passwords and system configurations can be easily exploited by attackers. Visa requires that things such as default passwords, community strings, and unused accounts be removed before deploying systems.

Unused services should be disabled. On Windows, it is often difficult to determine which services are necessary. Trial and error can be used on test systems to determine which services are needed. They should be documented in a standard Windows server configuration guide.

Under Linux, services are pretty straightforward and are documented, although some more than others. The 'chkconfig' command on Redhat based systems will list ('chkconfig –list') and change the services which are configured to start when the system boots. For example, to turn the Apache httpd service off:

```
chkconfig httpd off
```

The 'service' command can be used to immediately start or stop a service. To stop Apache:

```
service httpd stop
```

Unnecessary components should be removed. In Windows, most applications can be removed via the Add/Remove Programs application under the control panel. In Linux, installed packages can be listed with 'rpm -qa' and removed with 'rpm -e package_name'.

9. Track all access to data by unique ID

Visa requires that just about everything done on systems that hold credit card data, such as administrator activity, access to card data, logins, invalid logins, etc. must be logged securely and associated with a specific user by a unique ID.

Operating system, application, and firewall logs and audit trails can be sent to a dedicated syslog server or securely copied on a periodic basis. The syslog server needs to be protected so that logs cannot be tampered with. Logs should be backed up frequently to prevent tampering.

10. Regularly test security systems and processes

Systems need to be tested periodically for vulnerabilities caused by misconfiguration and software flaws. Luckily, there are quite a few free tools which can help detect security problems.

nmap[21] is a popular scanning tool which can be used to find out what services are running on a system as well as doing OS fingerprinting. It also has the ability to do scans based on specially crafted packets to get past mis-configured firewalls. A common scan is the stealth SYN scan which can be done from the LAN in the example network from Figure 1 toward the credit network with the command:

```
nmap -P0 -sS 192.168.2.0/24
```

This command scans all the hosts in 192.168.2.0/24 by half-opening TCP connections and looking at the response to the initial SYN packet. A SYN/ACK response indicates that the port is open. An RST response indicates that the port is closed. Instead of completing the three-way TCP handshake, the connection is torn down after the response is received.

Nessus[22] is a popular vulnerability scanner. It picks up where nmap leaves off, and in fact, it can use nmap as its port scanner. It can scan individual hosts or entire networks but does not stop once services are found. It has a large database of plugins for vulnerabilities that it can detect. Plugins are frequently updated and can be downloaded with the 'nessus-update-plugins' command. Custom plugins can be written in a simple scripting language called NASL.

Nessus has a nice GUI client for windows called NessusWX.[23]

Through the GUI client, Nessus can be configured to scan all the hosts in a network, for example, the credit network 192.168.2.0/24. Nessus scans each host that it can see and tests each of the services that it finds for vulnerabilities. When it finishes, the results are displayed with the option to generate a report in HTML, PDF, etc. The report lists all the vulnerabilities found along with a severity and a description which often gives suggestions for correcting the problem. Successive scan runs can be compared against previous scans to see progress or new vulnerabilities.

Audit custom application code for vulnerabilities

Vulnerabilities in application code can be abused by attackers causing applications to do things that the author did not intend. Special attention must be given to Internet facing applications due to their exposure to the entire world. Common attacks include XSS (Cross Site Scripting), SQL Injection, and parameter tampering.

Cross site scripting is an attack where malicious script or HTML code is injected into a dynamic page where parameters are not properly sanitized. When other users view the page, the additional code is sent to the user's browser, fooling it into doing things that the site creator did not intend. CERT has published a document on their website, "Understanding Malicious Content Mitigation for Web Developers"[24] which explains this type of attack and how to protect against it.

SQL Injection is an attack where parameters passed into a web application are tampered with to modify the behavior of SQL queries being called by the web application. For example, a form parameter that has not been properly sanitized being passed to an SQL select query, as the criteria for the "where" clause, could have characters which terminate the parameter early by embedding a single quote in the parameter string. Everything following will be interpreted as SQL code. An attacker could add a statement separator character, a semicolon is common, followed by a "drop table" statement. An SQL comment character may need to be added following the "drop table" to nullify the remainder of the original "select" statement. Allowing visitors to execute arbitrary SQL against a production database is not a good thing.

Naive web developers often rely solely on JavaScript code and constraints set in HTML form elements to restrict/validate form data. It is trivial to bypass all client side controls on form data. Attackers can easily craft new pages without such constraints or write scripts that will allow them to submit outrageous data. Poorly written CGI applications could be vulnerable to buffer overflows or other bugs that could be exploited by this attack. PHP has a feature called register_globals which was enabled by default in older versions. It would cause variables that were passed in as parameters from the browser to be created as variables in the program. This was handy for lazy programmers, but had a pretty nasty side effect in that, if a program did not properly initialize its variables to sane default values, an attacker could pass in new values for those variables, causing the program to do things that it should not. Needless to say, that feature is off by default in later versions of PHP.

Intrusion Detection

There are two common types of intrusion detection systems (IDS), host based and network based. Host based intrusion detection protects single hosts. Common implementations work as file integrity checkers such at Tripwire.[25]

Tripwire monitors critical files such as operating system components and application programs for unauthorized modifications and permission changes.

When a compromise is detected, alerts can be sent to the assigned personnel via SNMP, email, syslog, etc.

Snort[26] is a very popular open source network based IDS. (A number of commercial products are based on Snort.) Snort examines network traffic for attacks based on signatures, similar in concept to anti-virus software. The big difference is that Snort's main goal is to alert someone of an attack, not necessarily to stop it. Although, with FlexRESP support, Snort can stop some attacks by sending any combination of TCP reset or ICMP net, host, or port unreachable packets to either the sender, the receiver, or both.

Out of the box, Snort has no GUI. It just runs in the background, logging its output either to a log file or database. ACID [http://acidlab.sourceforge.net/] is a common web/PHP based front end for querying alerts, viewing alert details down to the packet payload, and generating statistics.

Like anti-virus software, Snort needs signatures or rules to be able to detect new vulnerabilities. The nice thing about it is that you can write your own rules. For example, if the only known traffic towards one of the credit servers (192.168.2.4) is on port 20020, a rule can be written to detect any attempts to connect to that server on any other port:

```
alert tcp any any -> 192.168.2.4 ! 20020 (msg: "Illegal
traffic detected";)
```

This tells Snort that when it sees traffic from any source host/port, toward 192.168.2.4 on any port but 20020 that it should fire an alert with the message, "Illegal traffic detected". This is a pretty simplistic example, but Snort has the ability to handle very complex rules.[27]

New Snort rules are released frequently. The process of downloading and installing them can be automated with a handy program called Oinkmaster.[28] Snort can be quite noisy with all the default rules enabled. Luckily, Oinkmaster can be configured to install new rules without breaking any local customizations.

11. Implement and maintain an information security policy

Policies and Procedures

Policies and procedures are important not only so that users know what is expected of them, but also as tool for enforcement. Employees should be required to sign a document stating that they have read and understand the security policies.

The SANS Institute has an invaluable resource called the Security Policy Project[29] which they have published for anyone to use, free of charge. They have a number of example policies covering a range of topics from encryption to wireless security.

Procedures tell how policies should be carried out. They can describe the steps

that should be performed in modifying firewall rules or creating new users. Visa requires procedures for monitoring and handling of security alerts, incident response and escalation, user management (addition and deletion of user accounts), and for monitoring and controlling data access.

Education

Education for users is very important. Users are typically only concerned about getting their jobs done with the least amount of discomfort. Strong passwords are hard to remember. Locking workstations is a pain. Encrypting email is tedious. Educating users on why it is necessary to follow these policies and procedures to protect company assets (and sometimes their jobs) makes employees more likely to comply.

Posters, fliers, and email messages with catchy graphics and slogans work well for distributing quick tips and reminders about being secure. Keep them pithy. Recognition and prizes for users who properly use security procedures are good ways to encourage compliance.

Employee screening

Visa requires that employees be screened to reduce the risks of internal attacks. While some naive organizations put their entire security budget into their perimeter defenses, according to the FBI/CSI the highest percentage of dollar losses comes from internal attacks:

> *"It's no surprise that the FBI/CSI (Computer Security Institute) report on computer-crime trends indicates that organizations this year had 70 percent of all attacks originating from the Internet. What might come as a surprise, however, is the breakdown of dollar losses. Despite the advances in exploitation trends that the Internet has provided, according to last year's FBI/CSI computer-crime report, more than 75 percent of all dollar losses came from internal intrusions."* [30]

12. Restrict physical access to data

Physical access controls are very important in protecting cardholder data. Visa requires access controls on physical access to computer systems, networking equipment, and all types of media used to store cardholder data.

Most security measures, such as password authentication, can be circumvented with local access to a computer. There are many recovery-type boot disks such as EBCD[31] with the ability to change any user's password, including the Windows administrator! UNIX systems have similarly easy access. Linux, for example can be rebooted into single user mode by passing the 'single' or 'init=/bin/sh' parameters to the kernel at boot time. To protect against this, the ability to boot from media other than the hard drive should be disabled, and access to modify the BIOS should be password protected. In some cases it is also possible to password protect the boot loader. Since these types of attack are likely to set off alarms

when the system is rebooted, Visa only requires that the consoles be locked. Additionally, password protection does nothing to protect against the theft of a server or its hard drives. Therefore, it is important to control physical access to sensitive computer systems.

Surveillance cameras should be used to monitor and record all access to the data center. Digital video recorders have revolutionized surveillance. They are typically of higher quality than VHS recorders, have programmable frame rates, have retention policies that can easily be set, are able to be monitored over the network, and have video that is tamper resistant with digital watermarks which can be used in court. Doors should be locked with access controlled by ID badge readers.

Visitor access should be strictly controlled. Visitors need to be easily identifiable. Visitor badges should not have access to restricted areas. Unauthorized individuals should be escorted at all times while in restricted areas.

Physical and electronic media containing sensitive data is an often overlooked security problem. Paper printouts with cardholder data should be destroyed by shredding them when no longer needed. Properly marked shredding bins work well.

Hard drives and tapes present a more difficult problem. Securely erasing a hard drive is very difficult, if not impossible. With the proper equipment, data can be retrieved even after it has been overwritten many times. Hard drives also have the ability to re-map bad sectors to good ones, leaving the bad sectors unavailable to be securely deleted.[32] For that reason, magnetic media that is no longer needed should be physically destroyed.

References

1. Bazely, Michael. “California Law on Data Privacy Going Into Effect.” San Jose Mercury News 1 July 2003.
2. Visa U.S.A. “Visa's Cardholder Information Security Program.” 2004 URL: http://www.visa.com/cisp/
3. Visa U.S.A. “The CISP Requirements” URL: http://usa.visa.com/business/merchants/cisp_index.html?ep=v_sym_cisp#b
4. Visa U.S.A. “Training and Tools.” URL: http://usa.visa.com/business/merchants/cisp_tools.html
5. Visa U.S.A. “Cardholder Information Security Program.” URL: http://www.visa.com/cisp/
6. Visa U.S.A. “CISP Groups Defined.” URL: http://usa.visa.com/business/merchants/cisp_index.html?ep=v_sym_cisp#d
7. Visa U.S.A. “Security Audit Procedures and Reporting.” URL: http://usa.visa.com/media/business/cisp/Security_Audit_Procedures_and_Repor ting.pdf
8. AirMagnet. “Best Practices for Securing Your WLAN.” URL: http://www.net-security.org/dl/articles/WLAN_Security_Best_Practices.pdf
9. Rekhter, Y., et al. “Address Allocation for Private Internets.” February 1996. URL: http://www.faqs.org/rfcs/rfc1918.html
10. Microsoft. “Software Update Services Home.” URL: http://www.microsoft.com/windowsserversystem/sus/default.mspx
11. Andreasson, Oskar. “Iptables Tutorial 1.1.19.” 2003 URL: http://iptables-tutorial.frozentux.net/iptables-tutorial.html
12. Patchlink. URL: http://www.patchlink.com/
13. Security Focus. “Security Focus Mailing List Subscription.” URL: http://www.securityfocus.com/subscribe
14. CERT-US. “National Cyber Alert System Mailing Lists.” URL: http://www.us-cert.gov/cas/signup.html
15. Sharma, Rajesh. Cisco Security Bible. Hungry Minds, 2002. 118
16. Application Security, Inc. “Encryption of Data at Rest.” URL: http://www.appsecinc.com/presentations/Encryption_of_Data_at_Rest.pdf
17. Schneier, Bruce. Applied Cryptography. John Wiley & Sons, Inc., 1996.
18. Warren, Blane. “Securing your Apache 2 Server with SSL.” August 2, 2004. URL: http://www.sitepoint.com/article/securing-apache-2-server-ssl
19. Poulsen, Kevin. “Guilty Plea in Kinko's Keystroke Caper.” July 18, 2003. URL: http://www.securityfocus.com/news/6447]
20. Hulme, George. “BREACH OF TRUST.” InformationWeek May 3, 2004 URL: http://www.informationweek.com/story/showArticle.jhtml?articleID=19400012
21. “Nmap.” URL: http://www.insecure.org/nmap/
22. “Nessus.” URL: http://www.nessus.org/
23. “NessusWX – Nessus Client for Win32.” URL: http://nessuswx.nessus.org/
24. CERT. “Understanding Malicious Content Mitigation for Web Developers.” URL: http://www.cert.org/tech_tips/malicious_code_mitigation.html

25.Tripwire, Inc. “Visa's CISP and Tripwire.” 2003. URL: http://www.tripwire.com/files/literature/white_papers/Visa_CISP_Flyer.pdf
26.“Snort The Open Source Network Intrusion Detection System.” URL: http://www.snort.org/
27.“Writing Snort Rules.” Snort Users Manual 2.2.0. URL: http://www.snort.org/docs/snort_manual/node14.html
28.“Oinkmaster.” URL: http://oinkmaster.sourceforge.net/
29.SANS. “The SANS Security Policy Project.” URL: http://www.sans.org/resources/policies/
30.Shipley, Greg. “How Secure is Your Network?” November 27, 2000. URL: http://www.nwc.com/1123/1123f1.html
31.“EBCD – Emergency Boot CD.” URL: http://ebcd.pcministry.com/
32.Gutmann, Peter. “Secure Deletion of Data from Magnetic and Solid-State Memory.” URL: http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html