Prometheus Global

OSSEC and PCI DSS Compliance

Casey Priester CISSP CISA SSCP CEH

Vice President

Prometheus Global Corporation

April 5, 2018

ATOMICORP ®

Casey Priester CISSP CISA SSCP CEH

- 20 years InfoSec experience
- Penetration Testing / Vulnerability Assessment
- Forensics
- Incident / Breach Response
- Auditing and Compliance
    - DITSCAP/DIACAP
    - FISMA
    - HIPAA
    - PCI DSS
    - NYDFS
- Currently supporting NRC cybersecurity regulatory program for power reactors, fuel fabrication and uranium enrichment facilities

What is PCI DSS?

- Security standards framework for protecting credit card information
- Developed by the Payment Card Industry (PCI)
- Applies to any organization that accepts, stores, or transmits cardholder data
- Contains over 250 requirements
    - ~100 administrative controls
    - ~95 technical controls
    - ~55 hybrid controls

PCI DSS Requirements

1. Install and maintain a firewall configuration to protect cardholder data.
2. Do not use vendor-supplied defaults for system passwords and other security parameters.
3. Protect stored cardholder data.
4. Encrypt transmission of cardholder data across open, public networks.
5. Use and regularly update antivirus software.
6. Develop and maintain secure systems and applications.
7. Restrict access to cardholder data by business need-to-know.
8. Assign a unique ID to each person with computer access.
9. Restrict physical access to cardholder data.
10. Track and monitor all access to network resources and cardholder data.
11. Regularly test security systems and processes.
12. Maintain a policy that addresses information security.

OSSEC Components that support compliance

- **Syscheck**: performs file integrity management
  - Can watch for changes to configuration files, log files, data files
  - Can detect changes to attributes as well as content

- **Rootcheck**: performs rootkit detection, enforces system policy
  - Can be used to enforce system hardening requirements
  - Can inspect system files, configuration filesoss

- **Logcollector / analysisd**: applies rule logic to collected logs
  - Can be used to detect user activity
  - Can run custom commands using **execd** to
    - extract system information
    - make changes to files and configurations
    - perform actions

PCI DSS Compliance using syscheck

8.1.2 Control addition, deletion, and modification of user IDs, credentials, and other identifier objects.

Solution: Use syscheck to determine if user accounts are added, deleted or changed

```
<syscheck>
    <directories realtime="yes" check_all="yes">/etc</directories>
</syscheck>

<rule id="100345" level="12">
    <if_matched_group>syscheck</if_matched_group>
    <match>/etc/shadow</match>
    <description>Changes to /etc/shadow - Critical
file!</description>
</rule>
```

PCI DSS Compliance using rootcheck

2.2 Develop configuration standards for all system components. Assure that these standards address all known security vulnerabilities and are consistent with industry-accepted system hardening standards.

- **2.2.1** Implement only one primary function per server to prevent functions that require different security levels from co-existing on the same server.
- **2.2.2** Enable only necessary services, protocols, daemons, etc., as required for the function of the system.
- **2.2.3** Implement additional security features for any required services, protocols, or daemons that are considered to be insecure.
- **2.2.4** Configure system security parameters to prevent misuse.
- **2.2.5** Remove all unnecessary functionality, such as scripts, drivers, features, subsystems, file systems, and unnecessary web servers.

Solution: Use rootcheck to enforce hardening policy

Sample RHEL7 Hardening Policy (rootcheck)

```
# 3.6 Configure Network Time Protocol (NTP) [CIS - RHEL7 - 3.6 -
NTPD not Configured {CIS: 3.6 RHEL7} {PCI_DSS: 2.2.2}]
f:/etc/ntp.conf -> r:restrict default kod nomodify notrap nopeer
noquery && r:^server;
f:/etc/sysconfig/ntpd -> r:OPTIONS="-u ntp:ntp -p
/var/run/ntpd.pid";


# 2.1.1 Remove telnet-server [CIS - RHEL7 - 2.1.1 - Telnet enabled
on xinetd {CIS: 2.1.1 RHEL7} {PCI_DSS: 2.2.3}]
f:/etc/xinetd.d/telnet -> !r:^# && r:disable && r:no;
f:/usr/lib/systemd/system/telnet@.service -> r:ExecStart=-
/usr/sbin/in.telnetd;


# 6.2.5 Set SSH MaxAuthTries to 4 or Less [ CIS - RHEL7 - 6.2.5 -
SSH Configuration - Set SSH MaxAuthTries to 4 or Less  {CIS - RHEL7
- 6.2.5} {PCI_DSS: 2.2.4}]
f:$sshd_file -> !r:^\s*MaxAuthTries\s+4\s*$;
```

PCI DSS Compliance using analysisd

8.1.4 Remove/disable inactive user accounts within 90 days
Solution: Use analysisd to periodically check users

```
<localfile>
     <log_format>full_command</log_format>
     <command>lastlog -b 90 | grep -v "Never logged in"</command>
     <frequency>86400</frequency>
</localfile>

<rule id="500001" level="5">
  <if_sid>530</if_sid>
  <match>ossec: output: ‘lastlog -b 90</match>
  <check_diff />
  <description>User(s) not logged in for 90 days.</description>
</rule>
```

PCI DSS Compliance using logcollector/analysisd

10.2 Implement automated audit trails for all system components to reconstruct the following events:

- **10.2.1** All individual user accesses to cardholder data
- **10.2.2** All actions taken by any individual with root or administrative privileges
- **10.2.3** Access to all audit trails
- **10.2.4** Invalid logical access attempts
- **10.2.5** Use of and changes to identification and authentication mechanisms and all changes, additions, or deletions to accounts with root or administrative privileges
- **10.2.6** Initialization, stopping, or pausing of the audit logs
- **10.2.7** Creation and deletion of system-level objects

Solution: Logcollector / analysisd

PCI DSS Compliance using logwatch with auditd

Advanced Solution: leverage auditd

```
## 10.2.1 Implement audit trails to detect user accesses to cardholder data
## This would require a watch on the database that excludes the daemon's access.
-a always,exit -F path=<path-to-db> -F auid>=1000 -F auid!=unset -F uid!=daemon-acct -F perm=r -F key=10.2.1-
cardholder-access

## 10.2.2 Log administrative action. To meet this, you need to enable tty
## logging. The pam config below should be placed into su and sudo pam stacks.
session   required pam_tty_audit.so disable=* enable=root

## 10.2.5.b All elevation of privileges is logged
-a always,exit -F arch=b64 -S setuid -Fa0=0 -F exe=/usr/bin/su -F key=10.2.5.b-elevated-privs-session
-a always,exit -F arch=b64 -S setresuid -F a0=0 -F exe=/usr/bin/sudo -F key=10.2.5.b-elevated-privs-session
-a always,exit -F arch=b64 -S execve -C uid!=euid -F euid=0 -F key=10.2.5.b-elevated-privs-setuid

## 10.4.2b Time data is protected.
## We will place rules to check time synchronization
-a always,exit -F arch=b64 -S adjtimex,settimeofday -F key=10.4.2b-time-change
-a always,exit -F arch=b64 -S clock_settime -F a0=0x0 -F key=10.4.2b-time-change
-w /etc/localtime -p wa -k 10.4.2b-time-change
```

OSSEC for PCI DSS Compliance Example

5.1 Deploy anti-virus software on all systems commonly affected by malicious software (particularly personal computers and servers).

Solution: rootcheck policy can ensure A/V is installed and running.

5.2 Ensure that all anti-virus mechanisms are maintained as follows: are kept current; perform periodic scans; generate audit logs.

Solution: logwatch/analysisd can check if anti-virus is updated and scans occurring.

5.3 Ensure that anti-virus mechanisms are actively running and cannot be disabled or altered by users, unless specifically authorized by management on a case-by-case basis for a limited time period.

Solution: syscheck can watch associated file and process permissions.

The Atomic OSSEC difference

Supports over 100 PCI-DSS technical and hybrid controls out of the box.

1. Install and maintain a firewall configuration to protect cardholder data.
   - Built-in firewall provides full functionality
2. Do not use vendor-supplied defaults for system passwords and other security parameters.
   - Scans all passwords, looks for blank/weak passwords
   - Will detect and alert on bad salts and ciphers
3. Protect stored cardholder data.
   - Built-in DLP capabilities that can detect PAN data and alert/redact/block
4. Encrypt transmission of cardholder data across open, public networks.
   - Automatically configures all services to run using strongest supportable encryption
   - Forces TLS where possible, checks for weak keys/ciphers
5. Use and regularly update antivirus software.
   - VirusTotal API integration

The Atomic OSSEC difference

6. Develop and maintain secure systems and applications.
    - Integrated Web Application Firewall (WAF) functionality
    - Virtual Patching will provide real-time protection against zero-days
    - Threat Intelligence module provides real-time black listing and other protections
7. Restrict access to cardholder data by business need-to-know.
    - Atomic OSSEC uses a least privilege Role-Based Access Control (RBAC) system
8. Assign a unique ID to each person with computer access.
    - Brute force detection
    - Supports two-factor authentication, can automatically enforce if the service supports it
9. Restrict physical access to cardholder data.
    - Automated sentry guns are currently in alpha testing (not really, though)
10. Track and monitor all access to network resources and cardholder data.
    - Employs a secure audit trail system
    - Understands and employs auditd natively
11. Regularly test security systems and processes
    - Built-in vulnerability scanner will check installed applications for flaws/known vulnerabilities
12. Maintain a policy that addresses information security.
    - Advanced hardening using OpenSCAP can assist in enforcement of defined policy

ATOMICORP ®

OSSEC as a Flexible Compliance Tool

- Provides artifacts for audit support
- Can use ELK stack for compliance dashboard
- Can support other compliance regimes:
  - FISMA
  - HIPAA
  - NYDFS
  - SSAE-16
  - FedRAMP
- Compliance reporting tools and modules currently in testing, release planned for Q3 2018.

Prometheus Global

THANK YOU!

Casey Priester CISSP CISA SSCP CEH

OSSEC and PCI DSS Compliance

ATOMICORP ®