Tokenisation: Reducing Data Security Risk

OWASP Meeting – September 3, 2009
Agenda

- Business Drivers for Data Protection
- Approaches to Data Security
- Tokenisation to reduce audit scope and lower risk
- Examples and Case Studies
- Questions
International Data Security Mandates

- Countries
  - United Kingdom – Companies Bill
  - Data Protection Act
  - European Union – European Union Privacy Act (EUPA)
  - Japan - Japanese Personal Information Act 2003 (JPIPA)
  - Canada – Personal Information Protection and Electronic Documents Act (PIPEDA)

- Industries
  - Payment Card Industry Data Security Standard (PCI DSS)
  - Code of Practice on Data Protection for the Insurance Sector (UK)
Many more if you do business in the U.S.

- **Government**
  - Sarbanes Oxley Act
  - Gramm Leach Bliley Bill
  - Healthcare Insurance Portability & Accountability Act (HIPAA)
  - Part 11 of the Title 21 Code of Federal Regulations
  - California State Bill 1386

- **Industry**
  - Payment Card Industry Data Security Standard (PCI DSS)
  - Healthcare Insurance Portability & Accountability Act (HIPAA)

- **Company**
  - Secure FTP - Bank of America, BankOne
  - AS2 - Walmart, Food Lion, McKesson
Data Security impacts a wide range of sensitive data

- Payment Card Industry Data Security Standard (PCI DSS)
- Credit / Debit Card Numbers
- Other Personally Identifiable Information:
  - Passport Number
  - Date/Place of Birth
  - Postal or Email Address
  - Telephone Numbers (home/mobile)
  - Mother's Maiden Name
  - Biometric Data
  - Unique Electronic Number, Address, or Routing Code
  - Telecommunication Id Information or Access Device
- Laws:
  - National Insurance Number
  - Social Security Number
  - Driver’s License Number
  - Bank Account Numbers etc.
- Healthcare:
  - Medical related information (Patient / Doctor, etc.)
Approaches to Data Security
Waves of Data Protection Investment

First Wave: Secure the perimeter – keep the bad guys out

Second Wave: Encrypt laptops, tape drives and mobile devices

Third Wave: Encrypt or tokenise specific data in databases and applications to neutralize breaches; pay more attention to internal threats
Trend in securing sensitive data

Boundary moving inward to the data itself

- PHI Protection
- PII Protection
- PCI DSS Compliance

Funded the development of solutions and best practices
PCI DSS Driving Best Practices
PCI DSS 3.1 – Minimise cardholder data storage

Protect Cardholder Data

Requirement 3: Protect stored cardholder data

Encryption is a critical component of cardholder data protection. If an intruder circumvents other network security controls and gains access to encrypted data, without the proper cryptographic keys, the data is unreadable and unusable to that person. Other effective methods of protecting stored data should be considered as potential risk mitigation opportunities. For example, methods for minimizing risk include not storing cardholder data unless absolutely necessary, truncating cardholder data if full PAN is not needed, and not sending PAN in unencrypted e-mails.

3.1 Keep cardholder data storage to a minimum. Develop a data retention and disposal policy. Limit storage amount and retention time to that which is required for business, legal, and/or regulatory purposes, as documented in the data retention policy.
**PCI DSS 3.4 – Render PAN unreadable**

**Options**
- Hashing
- Truncation
- Tokens
- Strong cryptography

### PCI DSS Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Testing Procedures</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4 Render PAN, at minimum, unreadable anywhere it is stored (including on portable digital media, backup media, in logs) by using any of the following approaches:</td>
<td>3.4.a Obtain and examine documentation about the system used to protect the PAN, including the vendor, type of system/process, and the encryption algorithms (if applicable). Verify that the PAN is rendered unreadable using one of the following methods:</td>
</tr>
<tr>
<td>- One-way hashes based on strong cryptography</td>
<td>- One-way hashes based on strong cryptography</td>
</tr>
<tr>
<td>- Truncation</td>
<td>- Truncation</td>
</tr>
<tr>
<td>- Index tokens and pads (pads must be securely stored)</td>
<td>- Index tokens and pads, with the pads being securely stored</td>
</tr>
<tr>
<td>- Strong cryptography with associated key-management processes and procedures</td>
<td>- Strong cryptography, with associated key-management processes and procedures</td>
</tr>
</tbody>
</table>

**Notes:**

- If for some reason, a company is unable render the PAN unreadable, refer to Appendix B: Compensating Controls.

- “Strong cryptography” is defined in the PCI DSS Glossary of Terms, Abbreviations, and Acronyms.

**3.4.1** If disk encryption is used (rather than file- or column-level database encryption), logical access must be managed independently of native operating system access control mechanisms (for example, by not

| 3.4.1.a If disk encryption is used, verify that logical access to encrypted file systems is implemented via a mechanism that is separate from the native operating systems mechanism (for example, not using local user accounts and databases). | 3.4.1.b Verify that cryptographic keys are stored securely (for example, stored on removable media that is ├──
mechanisms (for example, by not using local system or Active Directory accounts). Decryption keys must not be tied to user accounts.

3.5 Protect encryption keys used for encryption of cardholder data against both disclosure and misuse.

3.5.2 Store keys securely in the fewest possible locations and forms.
PCI DSS 3.6 – Rotate Keys Annually

and ...
- secure the keys,
- know which keys are used for which data,
- run your business, ....
Challenges of PCI DSS Compliance

- Store Card Holder Data (CHD) in fewest number of places
- Protect CHD wherever it is stored
- Store cryptographic keys in fewest number of places
- Rotate cryptographic keys at least annually
Tokenisation to reduce audit scope and lower risk
What kind of token are we talking about?

- It’s not the same as the ‘token’ used for two-factor authentication
- It’s not the ‘token’ used for lexical analysis in a programming language

- In data security, it’s a **surrogate value** which is substituted for the actual data (e.g. credit card) while the **actual data is encrypted** and stored elsewhere.
Tokens act as data surrogates

- Tokens maintain the length and format of the original data

- After tokenisation - tokens now reside where sensitive data previously resided in the application infrastructure
  - Input: sensitive data  Output: token
  - Input: token  Output: sensitive data

- Limits or eliminates modifications to applications.
Format Preserving Tokenisation

Tokens can be formatted to:

- Preserve the format (length and data type), and leading/trailing:
  
  
<table>
<thead>
<tr>
<th>Original data</th>
<th>head</th>
<th>body</th>
<th>tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>3752 5712 2501 3125</td>
<td>3752 0000 0010 3125</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Preserve length but not data type, and leading/trailing:
  
<table>
<thead>
<tr>
<th>Original data</th>
<th>head</th>
<th>body</th>
<th>tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>3752 5712 2501 3125</td>
<td>3752 X4mb AdLQ 3125</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Mask a portion of the token when a full value is not needed or desirable (can’t be subsequently translated back):
  
<table>
<thead>
<tr>
<th>Original data</th>
<th>head</th>
<th>body</th>
<th>tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>3752 5712 2501 3125</td>
<td>3752 **** **** 3125</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Tokens generally maintain the length and format of the original data so that applications require little or no modification.
Centralised Data Vault

- Protected Data Vault where sensitive data is encrypted and stored
  - Reduces the footprint where sensitive data is located
  - Eliminates points of risk
  - Simplifies security management
Tokenisation Model

Token Manager

Data Vault

Point of Sale

Human Resources

Backup

Loss Prevention

Marketing

Customer Relationship Management

Ciphertext in data vault

Tokens  Keys  Ciphertext
Tokens are surrogates for masked data

- Formatted tokens can be used wherever masked credit card information is required

<table>
<thead>
<tr>
<th>USING CREDIT CARD NUMBER</th>
<th>USING TOKEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>3752 5712 2501 3125</td>
<td>3752 0000 0010 3125</td>
</tr>
</tbody>
</table>

- Therefore wherever tokenised data suffices, risk is reduced
1:1 Token / Data Relationship

- Same token value is consistent for same data across entire enterprise; maintains **referential integrity** across applications
- Data analysis can be performed using token – e.g. data warehouse
Tokens Not Derived from Data

- Original data values cannot be mathematically derived from tokens
  - Tokens can be safely passed to databases, applications, mobile devices, etc.
- Token has no intrinsic value
- Solves the age-old problem of data for development and testing – it can be the same as production!
Test systems use ‘production tokens’

Production HR System
Germany

Outsourced Development
India

HR System

Production Data Vault

HR System

Masked Data Vault

Tokens Ciphertext
Centralised Key Management

- Control over who accesses sensitive data
- Rotate keys without having to decrypt and re-encrypt old data, and no system downtime
- Keys are distributed to token server, not throughout enterprise
Tokenisation Model

Token Manager

Data Vault

Point of Sale
Human Resources
Backup

Loss Prevention
Marketing
Customer Relationship Management

Tokens
Keys
Ciphertext

Ciphertext in data vault
Localised Encryption Model

Key Manager

Point of Sale

Human Resources

Backup

Loss Prevention

Marketing

Customer Relationship Management

Ciphertext is everywhere

Key and Ciphertext
Hybrid architecture includes both Central and Local protection mechanisms working with the same Enterprise Key Management.

- **Central Tokenisation**
- **Database Level Encryption & Tokenisation**
- **Application Level Encryption**

**Tokens**

**Tokens**

**Keys**

**Cipher text**

**Data Vault**

**Corporate Applications**

**Web Services**
Before: Order Flow without Tokenisation

80+ systems in PCI DSS scope
After: Order Flow with Tokenisation

Key Manager
Token Manager
Data Vault
Token Manager

Load Balancer
Web Services
Order Processing

Web Order Entry
Phone Order Entry
Mail Order Entry

Credit Card Entry Hub

3752 5712 2501 3125

Corporate Applications
Corporate Applications

Out of Scope
Case Study 2: Order Flow with Tokenisation

Web Customer → Internet → Web Servers → Web Services

Customer → Register at Store → Virtual Private Network → Web Services

Credit Card Authorization

Transaction Processing

Customer Database

Application Servers → Call Center

Token Manager

Key Manager

Data Vault
Thank you!

Questions?

For more information, visit:
http://nubridges.com/resource-center/