Analyzing Websites for User-Visible Security Design Flaws

http://cups.cs.cmu.edu/soups/2008/proceedings/p117Falk.pdf

Banking trends

- An increasing number of people rely on secure websites to carry out their daily business
- Pew Internet: 42% of all internet users bank online
- Forbes.com conducted a survey on +900 people and divided users in:
  - Used online banking systems and paid bills online
  - Used online banking systems but not online bill payments
  - Did not use online banking systems
- Those who used online banking were satisfied with the services.
- Those who chose not to use online banking cited security concerns as a reason why they did not use the services.

How banks deal with online security

- Due to the sensitive nature of these sites, security is a top priority
- Hire security experts to conduct vulnerability assessments
- Deploy encryption protocols such as SSL
- Monitoring accounts for suspicious activities
- Online security has improved compare with a few years ago

Study Conducted in the Paper

- Conducted during Nov - Dec 2006
- Analyses 214 U.S. financial institutions for user-visible security design flaws
- Design flaws are a result of decisions made during the website design phase and they promote insecure user behaviour
- These design features made it very difficult for someone to use the site securely

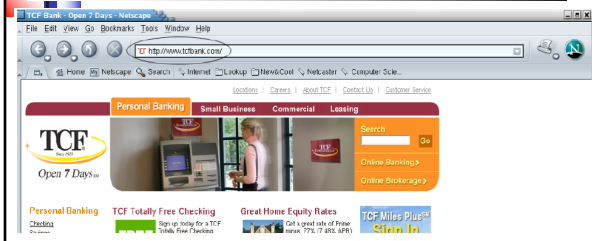
Design Flaws

- Break in the chain of trust: websites forward users to new pages that have different domains without notifying the users

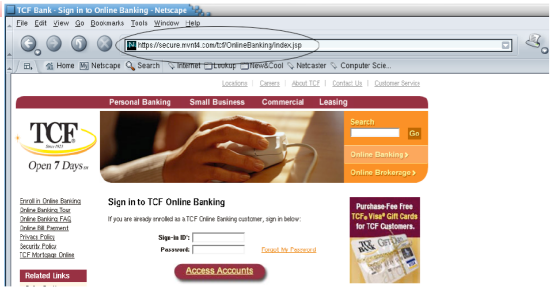
Break in the chain of trust

- Customer is redirected to a site that has a different domain name than the financial institution's site that was originally visited
- The switch is usually done without warning customers about such redirection
- It is up to the user to determine if the new site is really affiliated with the financial institution

Break in the chain of trust (Cont.)



Break in the chain of trust (Cont.)



- The signed certificate of https://secure.mvnt4.com, but TCF Bank was not the owner of this certificate. The owner is Metavante Corporation.

Example: Break in the chain of trust

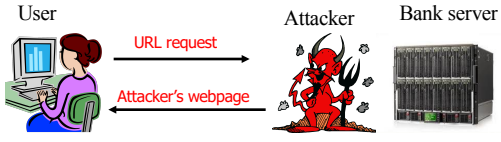
- University of Michigan credit union's website, users authenticate properly and are taken to a secure page.
- If an account holder decides to sign-up for Bill Pay, a new window pops up that belongs to a third-party.
- This window asks the user to enter info., such as mother's maiden name, SSN, account #, and birth date.
- No message is given, indicating that this pop-up from third-party website will occur.
- The credit union could have handled this design better, by either providing better disclosures or by not requiring the user to enter that information.

Design Flaws

- Break in the chain of trust: websites forward users to new pages that have different domains without notifying the users
- Presenting secure login options on insecure pages: Some sites present login forms that forward to a secure page but do not come from a secure page.

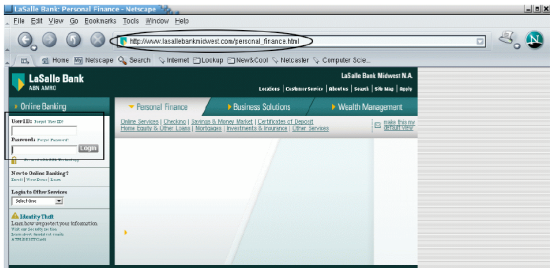
Presenting secure login options on insecure pages

- Login pages and options displayed on insecure pages leave users vulnerable to man-in-the-middle attacks.
  - They have no way of knowing if their usernames and passwords are being sent to a hacker site.



Presenting secure login options on insecure pages

- E.g. LaSalle Bank's -- http://www.lasallebank.com



Presenting secure login options on insecure pages

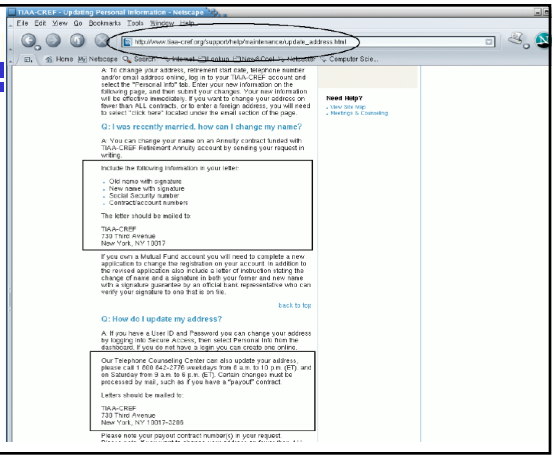
- Vanguard, a brokerage company, used to provide the login window on their home page (http page)
- Response: if a customer was concerned, the customer could hit the Submit button without entering a valid user id and password, and that would take the customer to an SSL protected login page.
- However, Vanguard modified their login process, moving the login window to an SSL-protected page.

Design Flaws

- Break in the chain of trust: websites forward users to new pages that have different domains without notifying the users
- Presenting secure login options on insecure pages: Some sites present login forms that forward to a secure page but do not come from a secure page.
- Contact information/security advice on insecure pages: some sites host their contact information etc. on insecure pages.

Contact information/security advice on insecure pages

- Contact information is considered security-relevant context because users rely on that information being correct for security-sensitive operations.
- Allows modification of the page by replacing the customer service phone numbers with bogus numbers
- Then crooks answer the phone and ask for SSN, birth date, or other confidential information



Design Flaws

- Break in the chain of trust: websites forward users to new pages that have different domains without notifying the users
- Presenting secure login options on insecure pages: Some sites present login forms that forward to a secure page but do not come from a secure page.
- Contact information/security advice on insecure pages: some sites host their contact information etc. on insecure pages.
- Inadequate policies for user ids and passwords: It is important to maintain consistent and strong policies on passwords and user ids.

Inadequate policies for user ids and passwords

- Design flaws
  - The use of email addresses for user IDs
    - E.g. LaSalle Bank website, www.lasallebank.com
    - TIAA CREF, www.tiaa-cref.com
  - No policy on allowed passwords creates weak passwords making them vulnerable to dictionary attacks.
- 31% of the banks affected allow e-mail addresses as user names
- They concluded that a strong username could be more important then a strong password.

Design Flaws

- Break in the chain of trust: websites forward users to new pages that have different domains without notifying the users
- Presenting secure login options on insecure pages: Some sites present login forms that forward to a secure page but do not come from a secure page.
- Contact information/security advice on insecure pages: some sites host their contact information etc. on insecure pages.
- Inadequate policies for user ids and passwords: It is important to maintain consistent and strong policies on passwords and user ids.
- Emailing security sensitive information insecurely

E-mailing security sensitive information insecurely

- Design flaw: the financial sites offer to send security-sensitive statements or passwords via emails.
- If passwords are e-mailed through an insecure mail server, an attacker could intercept unencrypted traffic on the network.

Detecting Design Flaws

- Use a tool for automatically detecting flaws
- They used wget to recursively download the financial institution websites and use scripts to recursively traverse and analyze the web pages

Detecting Breaking in the Chain of Trust

- For each web site, record the domain and search each page for URLs that did not match the domain.
- Looked for two cases:
  - Insecure pages making a transition to a secure page
  - A secure page making a transition to a secure page.

Presenting Secure Login Options on Insecure Pages

- Search each web page for the string login.
- If so, search the same page for the strings username or user id or password.
- If such strings were found on the same page, we then verified whether the page was displayed using the http protocol.
- http ➔ contained the design flaw.

Contact Information/Security Advice on Insecure Pages

- Search each web page for the string contact, information, or FAQ.
- If those strings where found, check whether the page was protected with SSL.
- If not, then we considered it to contain the design flaw.

Inadequate Policies for User IDs and Passwords

- The use of email addresses for user IDs
  - Search for the string e-mail
  - If such a page also contained the strings login and user id, it was assumed to violate the property.
  - They manually confirmed the results, filtering out any false matches.

Inadequate Policies for User IDs and Passwords (Cont.)

- Inadequate password strength policies
  - Search for the string password (excluding the Login pages).
  - If the string is found, searched for the presence of one of the following strings: recommendation, strong, or setting.
  - If so, they made a conservative assumption that the website had a policy on setting strong passwords.

E-Mailing Security-Sensitive Information Insecurely

- Search for the presence of either of the two strings statements or password as well as the presence of the two strings sending and e-mail.
- In order to reduce the number of false positives, we assigned values based on proximity.
  - The closer the two sets of words, the higher the value or probability.

Results

- With automated tools (such as this one) false positives are possible
- They tried to manually eliminate them wherever was possible
- Especially the "break-in-chain-of-trust" test has a significant number false positives (30% reported but in fact there were only 17%)
- Most sites made an effort to provide good policies for user ids and passwords

What did they find?

- 17% of the sites broke the chain of trust
- 47% presented a login page on an insecure page
- 55% presented contact and other sensitive information on insecure pages
- 31% allowed e-mail addresses as user names
- 76% of sites have at least one design flaw
- 68% had 2 or more design flaws
- 10% of the sites had all five design flaws
- 24% of sites were completely free of design flaws

Chapter 17.3
Secure Electronic Transaction (SET)

Secure Electronic Transactions (SET)

- Open encryption and security specification
- To protect Internet credit card transactions
- A wide range of companies were involved in developing the initial specification.
  - IBM, Microsoft, Netscape, RSA, etc.

Secure Electronic Transaction (SET)

- Not a payment system, rather a set of security protocols and formats that enable users to employ the existing credit card payment infrastructure on an open network in a secure fashion.
  - Provides a secure communications channel among all parties
  - Provides trust by the use of X.509v3 certificates
  - Ensures privacy because the information is only available to parties in a transaction.

SET Overview

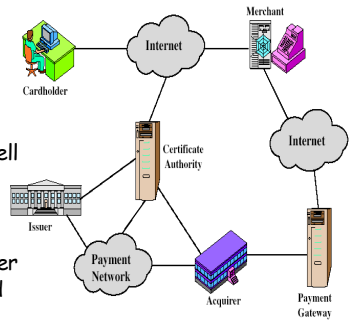
- Business requirements for secure payment processing with credit cards over the internet
  - Provide confidentiality of payment and ordering information - encryption
    - Assure the cardholders that this information is safe and accessible only to the intended recipient
  - Ensure the integrity of all transmitted data – digital signature.
    - No changes in content occur during transmission of SET message.
  - Provide authentication of a cardholder – digital signature and certificate.

SET Overview

- Business requirements for secure payment processing with credit cards over the internet
  - Cardholders need to be able to identify merchants with whom they can conduct secure transaction - digital signature and certificate.
  - Ensure the use of the best security practices and system design techniques to protect all legitimate parties in an electronic commerce transaction.
  - Create a protocol that neither depends on transport security mechanisms and prevents their use.
    - SET does not interfere with the use of other security mechanisms, e.g. IPSec and SSL/TLS.

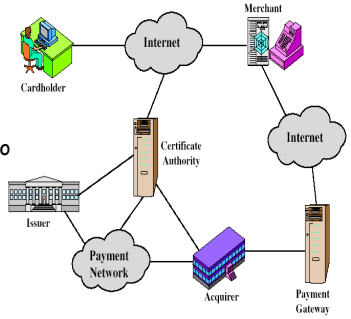
SET Components

- Cardholder: interact with merchants from personal computers over the Internet
- Merchant: a person or organization that has goods or services to sell to the cardholder.
- Issuer: a financial institution that provides the cardholder with the payment card



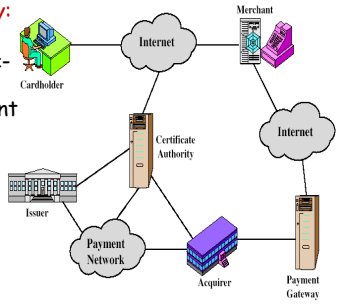
SET Components

- Acquirer: a financial institution that processes payment card authorizations and payments. Also provides electronic transfer of payment to merchant's account.
- Payment gateway: A function operated by the acquirer that processes merchant payment messages.



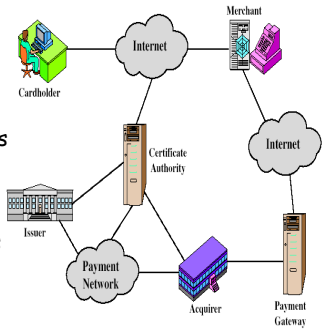
SET Components

- Certification authority: an entity that is trusted to issue public-key certificates for merchants, and payment gateways.



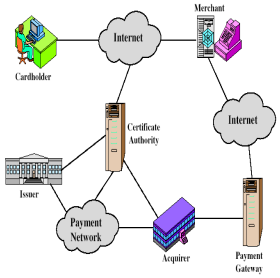
SET Transaction

1. The customer opens a credit card account with a bank that supports electronic payment and SET
2. The customer receives a certificate, which is signed by the bank.
3. The merchants have their own certificates
4. The customer selects items



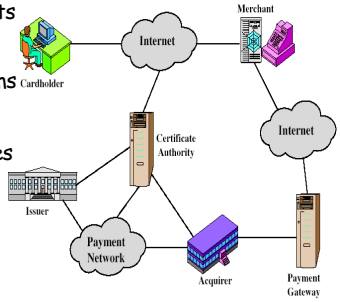
SET Transaction

5. The merchant sends a copy of its certificate so that the customer can verify that he/she is dealing with a valid store
6. The customer sends both order and payment to the merchant, along with the customer's certificate



SET Transaction

7. The merchant requests payment authorization
8. The merchant confirms order
9. The merchant provides goods or services
10. The merchant requests payment

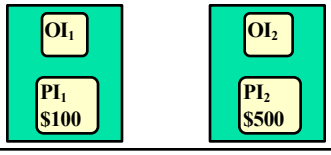


Dual Signature

- The goal of dual signature is to link two messages that are intended for two different recipient.
  - Order information (OI) for merchant
  - Payment information (PI) for bank
- Neither merchant nor bank needs details of other – the customer is offered protection in terms of privacy by keeping these two separate.
- But the two items must be linked in a way that can be used to resolve disputes if necessary.
  - So the customer can prove that this payment is intended for this order and not for some other order.

Dual Signature

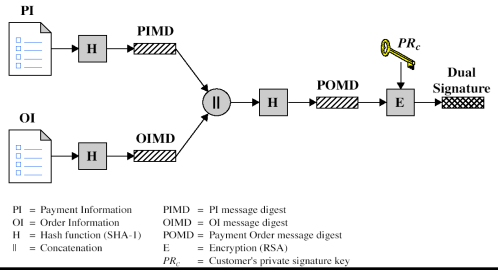
- Example: why link is needed:
  - Suppose that the customers send the merchant two messages: a signed OI and a signed PI.
  - The merchant passes the PI on to the bank
  - If the merchant can capture another OI from this customer, the merchant can claim that this OI goes with PI rather than the original OI



Dual Signature

- Solution: use a dual signature for this
  - Signed concatenated hashes of OI & PI
  - Encrypt the final hash with the customer's private key.

$$DS=E(PR_c, [H(H(PI) \| H(OI))])$$



PI = Payment Information
OI = Order Information
H = Hash function (SHA-1)
|| = Concatenation
PIMD = PI message digest
OIMD = OI message digest
POMD = Payment Order message digest
E = Encryption (RSA)
PR_c = Customer's private signature key

Dual Signature

- Dual signature: $DS=E(PR_c, [H(H(PI) \| H(OI))])$
- Merchant: the dual signature DS, OI, the message digest for PI (PIMD), and the pub-key of the customer

Dual Signature

- Dual signature: $DS=E(PR_c, [H(H(PI) \| H(OI))])$
- Merchant: the dual signature DS, OI, the message digest for PI (PIMD), and the pub-key of the customer
  - The merchant computes $D(PU_c, DS)$. if the same as $H(PIMD \| H(OI))]$, then the signature is verified.
- Bank: the dual signature DS, PI, the message digest for OI (OIMD), and the public-key of the customer

Dual Signature

- Dual signature: $DS=E(PR_c, [H(H(PI) \| H(OI))])$
- Merchant: the dual signature DS, OI, the message digest for PI (PIMD), and the pub-key of the customer
  - The merchant computes $D(PU_c, DS)$. if the same as $H(PIMD \| H(OI))]$, then the signature is verified.
- Bank: the dual signature DS, PI, the message digest for OI (OIMD), and the public-key of the customer
  - The bank can compute $D(PU_c, DS)$, if the result is the same as $H(H(PI) \| OIMD)]$, then the bank has verified the signature.

Dual Signature

- If the merchant wishes to substitute another OI in this transaction, it would then have to find another OI whose hash matches the existing OIMD.

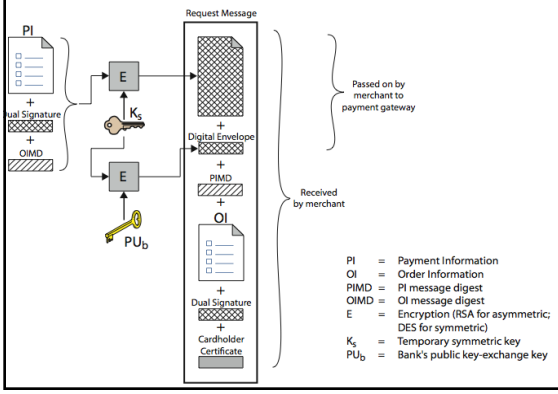
Payment Processing

- Purchase request: message from customer to merchant containing OI for merchant and PI for bank.
- Payment authorization: exchange between merchant and payment gateway to authorize a given amount for a purchase on a given credit card account.
- Payment capture: funds to be transferred to the merchant.

SET Purchase Request

- SET purchase request exchange consists of four messages
  1. Initiate Request – the customer requests the certificates of merchant.
  2. Initiate Response – the merchant generates a response.
  3. Purchase Request – the customer prepares the purchase request message including PI, OI, cardholder certificate, etc.
  4. Purchase Response – acknowledges the order and references the corresponding transaction #.

Purchase Request – Customer



Purchase Request – Merchant

1. Verifies cardholder certificates using CA public key.
2. Verifies dual signature using customer's public key to ensure order has not been tampered with in transit and that it was signed using cardholder's private signature key
3. Processes order and forwards the payment information to the payment gateway for authorization
4. Sends a purchase response to cardholder

Payment Gateway Authorization

- The merchant sends an Authorization Request message to the payment gateway
  - Purchase-related information obtained from the customer
  - Authorization-related information – generated by the merchant
    - An authorization block that includes the transaction ID, signed with the merchant's private key and encrypted with a one-time symmetric key
    - A digital envelop: encrypting the one-time symmetric key with payment gateway's public key.
  - Certificates – cardholder's and merchant's

Payment Gateway Authorization

1. Verifies all certificates
2. Decrypts digital envelope of authorization block to obtain symmetric key and then decrypts authorization block
3. Verifies merchant's signature on authorization block
4. Decrypts digital envelope of payment block to obtain symmetric key and then decrypts payment block.
5. Verifies dual signature on payment block
6. Requests & receives an authorization from issuer
7. Sends authorization response back to merchant

Payment Capture

- Merchant sends payment gateway a payment capture request
- Gateway checks request
- Then causes funds to be transferred to merchants account
- Notifies merchant using capture response