Real-life Example: Security Testing of an Online Banking Service

We will refer to the bank as Bank X

At the time of the experiments Bank X had
~ 30 million accounts
> 400,000 online accounts

Starting Point

- Blackbox testing
- No privileged information
- Had access to one online account
- Had Internet access
- Had letter from the bank verifying that we were working for them

Authentication

- User ID and Pin Code
  - User ID: Branch Number + Account Number + Control digit
  - Pin code: 4 digits
- Randomly generated personal information request (e.g., SSN, mother’s maiden name)
  - 2 out of 4 for personal
  - always EIN for business
- Used SSL for communication and a Java program with undisclosed encryption protocol

Experiments Attempted to Find Out

- What accounts existed
- What the pin number for each account was
- Who owned the account
  - personal
  - business
- Personal data on the owner/business

Client Applet

- 3 Java classes
- Classes were obfuscated
- Broke the obfuscation
  - constant strings were declared to be larger than they really were
  - parameters containing line feed were inflated to line feed plus carriage return without increasing the string size of the parameter

Client-side Java Classes

- Reverse engineered the java classes
  - built pre-decompiler to clean up bytecode
  - Used the Jasmine decompiler
- Studied the applet classes to better understand the protocols used
  - user interface
  - crypto algorithm
  - interface to the crypto algorithm
Custom Client-side Application

- Created an application to interact directly with the bank’s server
- Disguised itself as the bank’s applet
- Used two of the three original Java classes
  - crypto algorithm
  - interface to the crypto algorithm
- Used SSL for communication with the bank server

Account Information

- Branch number (4 digits)
- Account number (6 digits)
- Control number (1 digit)
- PIN - Personal Identification Number (4 digits)

Using our custom application we could get information on one account every six seconds

Different Approaches for Components

- Branch Number: from bank website
- Account Number: sequential
- Control Digit: from answer to log-in
  - were able to break the control digit algorithm after analyzing 240 account numbers
- User-friendly answers allow one to identify non-existent account and those that are not registered for online services

User Friendly Messages

Allow one to identify different accounts:
- Wrong branch or account or control digit
- Non existent
- Not Registered for online services
- Registered for online services
  - Wrong Pin
  - Right Pin

For Complete Access
Also Need Personal Information

- Social security number
- Date of birth
- Mother’s maiden name
- Father’s name
- Business identification number for business accounts

Need Account Owner’s Name to Get Personal Information

- Accomplished because
  - user friendly aspect of online transfer function
  - we had a legal online account
- Last step of the transfer function gives the name of the account owner to assure that the transfer is going to the correct account
First Complete Compromise

- Small town branch
- Discovered the pin for 25 accounts in a few hours
- Name of the owner using transfer method
- Complete set of personal data by talking to a person in the town

Used social engineering

Compromised Several Personal and Business Accounts

- Obtained personal/business information through
  - Social engineering
  - Purchase for cash (~$300.00)
  - System errors
    - able to substitute our choice of personal questions asked when returning answers
    - took advantage of inconsistencies between personal and business accounts
- Compromised large multi-national corporation that regularly made transfers greater than $100,000.00

Used Mail Relay to Send Bogus Messages to Bank Clients

- Allows one to send email that appears to have originated at the bank
- Can be used in social engineering attacks to induce users to disclose sensitive information
- An example: security management asking for PIN verification

Fake Site

Conclusions

- Current online systems have flaws
  - The flaws can be used to completely compromise accounts
  - The flaws can be explored remotely with no risk to an attacker
  - No privileged information about the systems is required in order to explore the flaws
- Computer security research is essential