

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
- Created an application to interact directly with the bank's server (using SSL for communications)

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

Online Banking Security

CS177 2008 7

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

Online Banking Security

CS177 2008 8

Different Approaches for Components

- Branch Number: from bank web site
- 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

Online Banking Security

CS177 2008 9

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

Online Banking Security

CS177 2008 10

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

Online Banking Security

CS177 2008 11

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

Online Banking Security

CS177 2008 12

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

```
Received: from bankx.com (provider.bankx.com [192.168.1.13])
  by mail.yahoo.com (8.8.6/8.8.6) with SMTP id MAA04E
  for <vigna@yahoo.com>; Thu, 21 Jan 1999 12:55:48 000 (PST)
From: security@bankx.com
Received: from SECURITYBANKXSECURITYBANKXSECURITYBANKX[...]
Date: Thu, 21 Jan 1999 18:53:52 -0200
Message-Id: <199901212053.SAA0008@bankx.com>
Apparently-To: vigna@yahoo.com
Content-Length: 1092

Dear Client:

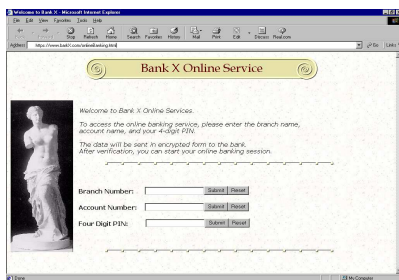
Bank X continuously works to be at the leading edge using new technologies and
providing user-friendly interactions so its clients have a pleasant experience.
It is with this in mind that we provide Internet mess to our clients. This new
way of interacting with your bank provides you with 24 hours a day service from the
convenience of your home.

Bank X has always made a big effort to assure the security of each client that uses
the internet. Unfortunately, there are many pin code that are not suitable to use,
because they are too easy to guess. Because of this we are asking our clients to
access the secure page on http://128.111.48.69. You can test your pin code on this
page to certify that it is suitable for the Internet. We ask your understanding and
hope that this inconvenience is a small one to assure your security. Thank you for
using our services.

Director of Security

Alice Smith
```

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