Hacking Web Apps

Def Con 11 version
Warning – Hazards to your Freedom

• Unauthorized access to systems & data is illegal in most places.
  – Get permission in writing before performing scans, audits, assessments, etc!
  – For details see http://www.lightlink.com/spacenka/fors/
This is not a Drill

• **True Stories**
  – The vulnerabilities you are about to see are real, only the names have been changed to protect the vulnerable.
  – Discovered over the past several years by the author during AUTHORIZED security assessments of customers
    • consumer banking, credit cards, travel reservations, B2B banking, 401K, stock broker, project collaboration & document sharing

Copyright 2002-2003 - David Rhoades
Course Purpose

• **We will cover...**
  – various web application weaknesses
  – tools & methods to find and exploit them

• **We will not cover...**
  – comprehensive audit/assessment methodologies
  – all tools/techniques
  – solutions for holes seen
About the Instructor/Author

- **David Rhoades**
  - PSU - B.S. Computer Engineering
  - Info Sec since 1996
  - david.rhoades@mavensecurity.com

- **Maven Security Consulting, Inc.**
  - www.MavenSecurity.com

(I’m the one on the right.)
Course Agenda

- **The Problem**
- **Tools of the Trade (i.e. warez)**
- **Points of Attack**
  - live demos
- **Further Resources**
The Problem
(Can’t we all just get along? ...No!)

- **Web sites are hacked for various reasons:**
  - political, revenge, fame, fortune, fun (genetic?, vitamin deficiency?)

- **Not just web “sites” - applications too**
  - Hotmail, CD Universe, shopping carts
  - See for the latest casualties

- **SANS/FBI – The Twenty Most Critical Internet Security Vulnerabilities**
  - Web servers are at the top of the list, see
    [http://www.sans.org/top20/](http://www.sans.org/top20/)
  - Vulnerability stats

- **The results:**
  [www.zone-h.org/en/defacements](http://www.zone-h.org/en/defacements)
  - bad press => lost customer confidence => lost revenue & legal consequences

Copyright 2002-2003 - David Rhoades
Tools of the Trade Overview

• The Problem
  ➢ Tools
• Points of Attack
• Resources

HTTP – Hyper Text Transfer Protocol
HTML – Hyper Text Markup Language

• Some essential techniques
  – Intercept & manipulate raw HTTP
  – Mirror web sites
  – Automate fake browser requests (a.k.a. brute force)
  – Decompile Java Applets
Technique – Traffic Interception & Manipulation

• **Purpose: Manipulate Input**
  – *Bypass client-side size restrictions*
    - HTML’s MAXLENGTH
    - Client-side JavaScript filters
  – *Violate the protocol (i.e. HTTP)*
  – *Insert alternate choices into lists and pull down menus*
  – *Change cookies, hidden elements, everything & anything*

• **Other purpose**
  – *Record HTTP/HTML for analysis (e.g. code comments, custom headers)*
Interception Tool – Achilles Intro

• *Old news* World’s first publicly released general purpose web application security assessment tool

  – Concept: David Rhoades  
    (with apologies to web app developers everywhere)

  – Code: Robert Cardona

  – Released Oct 2000
Achilles – Matrix-style Web Proxy

• **Simple web proxy**
  – Win32 GUI or UNIX via WINE
  – Notepad with an attitude

• **Freeze traffic mid-stream and modify**
  – outbound and inbound browser traffic
  – SSL and non-SSL
  – Change any HTTP header, cookie, form element
    • Body length automatically recalculated for POST statements
  – Log all traffic to a text file
Achilles – HTTP Exposed

- **SSL does not protect your web app, it protects traffic in transit**
  - Provides server/client auth too

```
HTTP/1.1 200 Ok
Server: Xitami
Date: Tue, 19 Mar 2002 04:13:18 GMT
Content-length: 4114
Content-type: text/html; charset=ISO-8859-1
Date: Tue, 19 Mar 2002 04:13:18 GMT
Expires: Tue, 19 Mar 2002 04:13:18 GMT
Set-cookie: oldSessionID=1016511198; path=/cgi-bin/bb2.cgi

<?xml version="1.0" encoding="utf-8"?><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN" "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US"><head><title>Welcome to Buggy Bank - Login</title><head><body><BODY BGCOLOR="#FFFFFF" LINK="#000099" VLINK="#336699" TEXT="#000000" TOPMARGIN=0 LEFTMARGIN=0 MARGINWIDTH=0 MARGINHEIGHT=0></body></html>
```
Achilles – Architecture for SSL Sites

Web Browser

Achilles looks like a web server to the browser

Achilles

SSS 1

Web Server

Achilles looks like a web browser to the remote site

Copyright 2002-2003 - David Rhoades
DEMO – Achilles

- Capture outbound web request
- Capture inbound reply
Achilles – Stupid Party Tricks: Modify Inbound Traffic Too
Several ‘intercept and modify’ proxies are now available…much better than Achilles

- **WebProxy v1 (freeware)**
  - Java (Windows/UNIX)
  - Auto hack feature (i.e. fuzz)

- **WebProxy v2+ (Commercial)**
  - [http://www.atstake.com/webproxy](http://www.atstake.com/webproxy)

- **Spike Proxy**
  - Python script (Window/UNIX)
  - Auto hack feature (i.e. fuzz)
  - [www.immunitysec.com/spikeproxy.html](http://www.immunitysec.com/spikeproxy.html)
Tools – More Intercept & Modify Proxies

- **Tool: Odysseus**
  - Win32 EXE
  - GUI/SSL/Proxy based

- **Tool: Paros v2.2 Free Edition**
  - [http://www.proofsecure.com](http://www.proofsecure.com)
  - Win32 EXE
  - GUI/SSL/Proxy based
  - HTTP 1.1
  - spider function
  - XSS testing

- **Tool: PenProxy**
  - Java (Windows/UNIX)
  - No SSL/TLS support

- **Tool: HTTPush**
  - [http://sourceforge.net/projects/httpush](http://sourceforge.net/projects/httpush)
  - Client interface thru browser
  - Open Source Project
  - XML plugins (e.g. whois)
  - SSL and non-SSL
  - This tool is not actively being developed.

Copyright 2002-2003 - David Rhoades
Tools – Browsers/Browser Extensions

- **These are browser-like, or browser extensions useful for manipulating web traffic**
  - All IE-based

- **Form Scalpel**
  - [http://www.ugc-labs.co.uk/tools/formscalpel/](http://www.ugc-labs.co.uk/tools/formscalpel/)

- **IE Booster**
Tool – General Purpose Tool Kits for Web App Testing

- **Web Sleuth**
  - http://www.geocities.com/dzzie/sleuth/
  - **Platform:** Win32 GUI
  - **Purpose:** All-in-one web app security audit tool set.
    - Parses web pages to catalog forms, cookies, HTML comments, etc...
    - Modify form elements manually
    - Modify form elements automatically (via plugin)
  - Supports SSL
  - Free, open-source version
  - Commercial version

- **Web Scarab**
  - www.owasp.org/webscarab/
  - Java based
  - “…a true ‘Open Source’ web application security assessment tool. The tool will be able to examine a complete web site or individual applications running within a web site for security issues.”
  - Status: Beta now available. More coming...
A closer look at WebProxy – Features

- **Works with HTTPS (SSL/TLS).**
- **Fuzzing – permutations of user selected traffic components**
  - text file defines input (fuzzstrings)
  - text file defines signature to look for in server’s output (errorstrings)
- **Automatic, on-the-fly, find-and-replace of HTTP traffic**
WebProxy – Administration Interface

- **Interface via browser**
  - change browser’s proxy settings

- **Surf to** http://webproxy
WebProxy – Terminal Window Monitor

- A command prompt window will display client requests and server responses

- Beware of "Select" mode
WebProxy – Intercepting Browser Requests
WebProxy – “Un”documented Features

• **Official FAQ states**…
  – “Are there any undocumented features in WebProxy? **Yes.**”

• **Transparent proxy**

• **Add to .webproxyrc file**
  – addproxy transhttp 5113 <REMOTE PROXY IP> 8080 127.0.0.1
  – **Transparent proxy now running on 127.0.0.1 port 5113**
  – Remote proxy on port 8080 will think it is the only proxy

• **Now you can daisy chain with a normal proxy.**

• **Normal proxy will not see WebProxy (i.e. transparent)**
Tool – IE Booster Intro

- **Web Browser Extensions for IE 5/6**
  - Extended context menu (left click)
  - Show all forms and applets of a web page
  - See and edit hidden form elements 😊

- **Version 1.4** (Freeware)
  - www.filelibrary.com: 8080/cgi-bin/freedownload/New_Files/n/150/iebooster.zip

- **Version 2.x** (Shareware – 30 day trial)
  - www.paessler.com/iebooster
Technique – Brute Force Authentication

• **Brutus**
  - [www.hoobie.net/brutus/index.html](http://www.hoobie.net/brutus/index.html)

• **Platform: Win32 GUI**

• **Purpose: Brute force web logins (both kinds – Country & Western)**
  - HTTP Basic Authentication
  - Form-based Authentication
    - GET or POST

  - Brute forces other protocols too
    - FTP, telnet, POP3, SMB...

Copyright 2002-2003 - David Rhoades
Brute Force Tool – Brutus Features

- **Brute force many types of auth**
  - web forms and Basic auth
  - POP, telnet, FTP, SMTP

- **Exhaustive word list generation**
  - all lower case character strings 6 to 8 characters long

- **HTML form viewer**
  - to assist in form based brute force

- **Built in script maker**
  - to learn new protocol for brute forcing

- **Word list permutations**
  - password -> pa55w0rd
Other Brute Force Tools for Web Apps

- **Win32: wwwhack**
  - [http://packetstormsecurity.org/Crackers/wwwhack.zip](http://packetstormsecurity.org/Crackers/wwwhack.zip)

- **UNIX: Authforce**
  - [kapheine.hypa.net/authforce/index.php](kapheine.hypa.net/authforce/index.php)

- **Win32: Brutus**
  - [http://www.hoobie.net/brutus/index.html](http://www.hoobie.net/brutus/index.html)

- **UNIX: THC Hydra**
  - [www.thc.org/releases.php](www.thc.org/releases.php)

- **Nessus (specific plugin)**
  - “Unknown CGIs arguments torture”
  - Brute forces CGI parameters in general, not just authentication

- **Screaming Cobra [cobra.lucidx.com](http://cobra.lucidx.com)**
  - no SSL; not being updated; but nice proof-of-concept (crawl and fuzz)

Copyright 2002-2003 - David Rhoades
Other Brute Force References

- **Word Lists**
  - [www.packetstormsecurity.nl/Crackers/wordlists/](http://www.packetstormsecurity.nl/Crackers/wordlists/)

- **Build word variations**
  - [sourceforge.net/projects/variation](http://sourceforge.net/projects/variation)
Technique – Decompiling Java Applets

- **Compiled into byte-code, but can be decompiled**
- **Java Applets from...**
  - Client-side code
  - Stolen from server
  - Lots of apps (WebProxy) are Java
- **May contain sensitive info**
  - username / password
  - “secret” URLs
  - undocumented features
Tools – Java Decompiling

- **JAD**

- **Mocha**
  - [http://www.brouhaha.com/~eric/computers/mocha.html](http://www.brouhaha.com/~eric/computers/mocha.html)

- **Sourcetech**
  - [http://www.srctec.com/decompiler/index.htm](http://www.srctec.com/decompiler/index.htm)
Technique – Mirror/Crawl Web Site

- **Automated Mirror**
  - Use web mirroring software (AKA. robots, crawlers, spiders, offline browsers) to download the site onto your hard drive.
  - Search the captured files for...
    - HTML and script comments
    - Inappropriate use of the GET method (versus POST)
    - GENERATOR tags (e.g. FrontPage)
  - Try to capture HTTP headers for more info...
    - X-Accelerated-By: PHPA/1.3.3r1
    - Server: Apache/1.3.19 (Unix)
    - X-Bender: Care to contribute to the Anti-Mugging-You Fund?
Tools – Mirror/Crawl Web Sites

**Freeware**

- UNIX/Windows: HTTrack (open source and free)
  - Can override robots.txt restrictions
  - Not supported by ads; not spy ware
  - Mozilla extension (Spiderzilla) available
  - UNIX: wget
    [freshmeat.net/projects/wget/](http://freshmeat.net/projects/wget/)

**Commercial**

- Windows: BlackWidow
  [www.softbytelabs.com](http://www.softbytelabs.com)
  - HTTP, HTTPS, and FTP

Copyright 2002-2003 - David Rhoades
Attack Agenda Roadmap – Authentication

• The Problem
• Tools
  ➢ Points of Attack
• Resources

• Some points of attack
  ➢ Authentication
    – Session Tracking
    – Unexpected Input
    – Application Logic
DEMO – Attacking Authentication

- **wwwhack**
  - [http://packetstormsecurity.org/Crackers/wwwhack.zip](http://packetstormsecurity.org/Crackers/wwwhack.zip)
  - **NOTE:** Shareware? Porn ads?

- **Demo Site**
  - [http://www.vaporware.usa/cgi-bin/calendar.pl?calendar=vaporexternal&template=login.html](http://www.vaporware.usa/cgi-bin/calendar.pl?calendar=vaporexternal&template=login.html)
  - **NOTE:** key phrases (Pick something that is unique to the FAILED attempt)
Authentication Attack – Attacking Locked Accounts (PIN Harvest)

• **Q:** Locking accounts will prevent brute force attacks….right?

• **A:** Not always.

• **There is username harvesting…**
  – Bad login reveals valid user names

• **But what about password/PIN harvesting?**
  – Locked account + error message = correct PIN revealed
Authentication Attack – PIN Harvest Real World Example

Real example found in major consumer banking application in Europe a few years ago.

- **Example:**
  - When trying the wrong PIN for a locked account, the web application returned:
    - Leider ist diese PIN falsch. [Unfortunately this pin is wrong.]
  - When trying the correct PIN for a locked account, the web application returned:
    - Leider ist Ihre PIN nicht mehr gültig. [Unfortunately your pin is no longer valid.]
Authentication Attack – Bypass Authentication

- *If you cannot beat the authentication perhaps you can bypass it.*

- Viewing public calendar without login we see:
  - http://vaporware/cgi-bin/calendar.pl?calendar=vaporexternal

- *Demo: See Mar 2002 for calendar=secret*
Attack Agenda – Session Tracking

- The Problem
- Tools
  - Points of Attack
- Resources

**Some points of attack**
- Authentication
  - Session Tracking
- Unexpected Input
- Application Logic
Session Tracking Intro

• **Session Tracking**
  - Session ID is unique identifier
  - Embedded into traffic via URL or Cookie

Set-cookie: CGISessionID=1344107640;path=/

• **Forms of attack:**
  - Predict, Brute Force, or Pinch (i.e steal)
Session Cloning via Prediction

Session ID Attacks:
- Predict
- Brute Force
- Pinch

Steps for Prediction Attack
- Determine how & when session ID is assigned
  - E.g. before login via cookie
- Collect several session IDs
  - Rapid fire: one after another
- Analyze for pattern or predictability
  - Based on time stamp? Source IP? MD5 checksum of both?
(Tool) iDefense Intro: Cookie Collecting Made Easy

- **iDefense Web Application Session Auditor**
  - Win32 GUI
  - for the coding impaired 😊

- **URL**
  - [www.idefense.com/idtools/Session_Auditor.zip](http://www.idefense.com/idtools/Session_Auditor.zip)

- **Version 1.0**
  - Cookie brute-force does NOT work
  - It tries to send Set-Cookie, rather than Cookie:
**DEMO – Session Tracking: Collect & Analyze Session ID**

**Session ID**  
**Attacks:**  
- Predict  
- Brute Force  
- Pinch

**Tool – iDefense**  
- **WebMaven – Buggy Bank**  
  - SessionID assigned before login via cookie

- **VaporWare Calendar**  
  - similar data for recent audit of online reservation system  
  - looks random but...  
  - Worse example: credit union software

---

**Sample Data**

---

Copyright 2002-2003 - David Rhoades
Session Cloning via Brute Force

Session ID
Attacks:
- Predict
- Brute Force
- Pinch

- Sometimes the session ID is from a small range of choices

- Attack: Request all/most possible combinations
DEMO – Brute Force Session ID

Session ID Attacks:
- Predict
- Brute Force
- Pinch

• Tool – *iDefense Web Application Session Auditor*
  – *ideal if session ID is inside the URL*
  – *cookie brute force feature is broke in v1.0*

• Site *WebMaven-BuggyBank*
  – *session ID embedded in cookie before login*
Command Line Kung Foo – cURL Intro

--silent = hide curl status junk
--include = show HTTP headers
    --cookie = add your own cookies
    --data = add POST data

Target URL

$ curl --silent --include --cookie
    'SessionID=1059750438' --data
    'from=1234567890123750&to=1234567890123751&amount=100000000&transaction=transfer2'
http://webmaven.usa/cgi-bin/wm.cgi?transaction=transfer
DEMO – Brute Force Session ID from Command Line

- $ curl --silent --cookie 'SessionID=1059777280' http://www.webmaven.usa/cgi-bin/wm.cgi?transaction=summary | grep -o -P 'Account Summary for .*?\<'

- $ perl -e 'for ($x=875;$x<=975;$x++) {print "Session ID 1059835\$x"; system ("curl --silent --cookie 'SessionID=1059835\$x' http://www.webmaven.usa/cgi-bin/wm.cgi?transaction=summary");} | grep -o -P 'Account Summary for .*?\<|Session ID .*?\<' | grep -B 1 Account
Session Cloning via Pinching

Session ID Attacks:
- Predict
- Brute Force
- Pinch

• Steps for Cookie Pinch Attack
  - *Session ID is very robust* – difficult or impossible to predict
  - *Therefore, try stealing valid session IDs via Cross Site Scripting (XSS)*
DEMO – Session Cloning via XSS Cookie Pinch (Looky, looky, I got your cookie!)

Session ID
Attacks:
- Predict
- Brute Force
- Pinch

- Define XSS
  - User input and/or web app output not filtered; might contain client-side code; browser is attacked

- Simple demo
  - http://localhost/cgi-bin/testcgi?
    <script>alert("Hello")</script>

- See Vaporware app

- If Session ID is in cookie then it can be sent to remote site
  - <SCRIPT>
  </SCRIPT>

Copyright 2002-2003 - David Rhoades
Attack Agenda – Unexpected Input

• The Problem
• Tools
  ➢ Points of Attack
• Resources

• Some points of attack
  – Authentication
  – Session Tracking
  ➢ Unexpected Input
    ➢ SQL Injection
    ➢ Buffer Overflow
    ➢ Command Injection
    ➢ etc...
  – Application Logic
Unfiltered User Input

- Lots of names for this concept
  - SQL Injection
  - Buffer Overflow

- Unexpected input might cause error
  - Special characters
  - Too big
  - Alternate choice

Error Occurred While Processing Request

Error Diagnostic Information

ODBC Error Code = 37000 (Syntax error or access violation)

[Microsoft][ODBC SQL Server Driver][SQL Server]Line 1: Incorrect syntax near 'action'.

Data Source = "secretDB"

SQL = "SELECT * FROM request WHERE sessionID = 123 AND action = 'sum:mary"

The error occurred while processing an element with a general identifier of (CFQUERY), occupying document position (6:1) to (6:50) in

the template file D:\Itempub\databases\bankusers\user_access.cfm
DEMO – Unfiltered User Input / Web Server Output

- **Error message too detailed**
  - **SQL / ODBC Errors**
    - How: account number during login
    - Result: Access to entire DB
  - **Aux. Program Errors**
    - How: Semicolon (%3B) in the “Account” cookie
    - Result: run commands

- **XSS**
  - **Seen earlier**
  - Result: Attack, eavesdrop, and clone user’s session ID (cookie-based)
Command Injection Attack

- **Found in online banking app (very large bank)**
- **Cookie held encrypted account number**
  - Cookie used to speed-up login process
  - `Account=pCqzl3mSxE8gD3aQfHeKHOmBJCyGca7M6mtaLPn6zINsScc3l%2FF5FdGUi0Kg%3D%3DvV3i`
Command Injection – The Encrypted Account Cookie

- **Browser**
  - **First time**
    - User enters full 16 digit account number
    - “Account” cookie is stored for future visits
  - **Return Visits**
    - “Account” cookie sent
  - **Useful where many accounts were used**

- **Server**
  - Encrypts account # with PGP
  - Embeds encrypted account # into cookie
  - Account cookie sent to browser
  - Account cookie decrypted
  - HTML for login screen shows last four digits in drop down menu

Copyright 2002-2003 - David Rhoades
Manipulating the cookie value (e.g. inserting semi-colon) revealed this error:

– PGP v2.6 error

How was our cookie data getting fed to PGP?

– Maybe
  # pgp $COOKIE_DATA
– So, then our data is passed across a command line? :-)
– What if $COOKIE_DATA = junk ; netstat
Command Injection Results

PGP v2.5 error

Active Connections Proto Local Address Foreign Address State TCP 0.0.0.0.0:21 0.0.0.0.0 LISTENING TCP 0.0.0.0.0:80 0.0.0.0.0 LISTENING TCP 0.0.0.0.0:81 0.0.0.0.0 LISTENING TCP 0.0.0.0.0:135 0.0.0.0.0 LISTENING TCP 0.0.0.0.0:1026 0.0.0.0.0 LISTENING TCP 0.0.0.0.0:3851 0.0.0.0.0 LISTENING TCP 0.0.0.0.0.0.0.0.0 LISTENING TCP 0.0.0.0.0.0.0.0.0 LISTENING TCP 127.0.0.1.80 127.0.0.1.3564 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3571 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3585 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3592 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3599 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3606 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3613 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3620 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3627 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3634 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3641 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3648 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3655 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3662 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3669 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3675 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3683 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3690 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3697 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3704 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3711 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3713 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3725 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3732 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3739 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3746 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3753 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3760 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3767 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3782 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3785 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3787 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3788 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3791 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3792 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3794 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3796 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3799 TIME_WAIT TCP 127.0.0.1.80 127.0.0.1.3800 TIME_WAIT TCP 127.0.0.1.80
Attack Agenda – Application Logic

• The Problem
• Tools
  ➢ Points of Attack
• Resources

• Some points of attack
  – Authentication
  – Session Tracking
  – Unexpected Input
  ➢ Application Logic
  • Application performs steps in the wrong order, or some other flaw in the underlying logic or design

Copyright 2002-2003 - David Rhoades
Buggy Bank Demo: Viewing Other Account Balances

Proper Sequence:
A Authorized to take money from?
B Authorized to put money in?
C Enough balance?

• **View the balance of other accounts**
  – Discovered a few years ago in credit union software
  – Web app did step C first

• **Attempt transfer of funds between accounts**
  – Change the FROM account to someone else’s
  – Small amount...transfer is prevented
  – But, make amount very large...Result: account balance error
DEMO – Attack Application Logic: Collecting Balances

- **Tool: Custom Perl script**
  - Brutus and others might work too.

- User can change FROM account to someone else’s account when transferring funds

- Can also collect valid account numbers too.

Copyright 2002-2003 - David Rhoades
Conclusion

Closing Thoughts & Resources
Conclusion – Limitation of Tools

- **The Problem**
- **Tools**
- **Points of Attack**
  - Resources

- **Brain & clues not included**
  - You have to know what you’re looking for (e.g. view account balances)

- **No one tool does it all...(yet?)**

- **Some tools don’t support SSL**
  - Try stunnel to wrap in SSL

- **For thorough testing you will need to code/script your own tools.**
Resources – Beyond Point & Click Tools

• **Elza** – scripting language for interacting with web sites and apps
  – Poor man’s Perl...in fact, Elza is a Perl script
  – Easier than learning Perl (?)

• **cURL** - command line tool for **HTTP(S)**

• **Perl with libwww-perl (LWP)**

• **Regular Expressions (regex)** – take the red pill
  – But if you do, there’s no going back...

I know Kung Foo

Copyright 2002-2003 - David Rhoades
Resource – (aka Buggy Bank)
WebMaven: Web App Audit Trainer

• “Give a man an audit and he will be secure for a day. Teach a man to audit and he will be secure for the rest of his life.” - David Rhoades

• Fake web app that emulates vulnerabilities.

• Run it on your own web server
  – safe & legal way to practice audit techniques & learn
  – benchmark audit tools

• http://webmaven.MavenSecurity.com
Resources –
Web App Security Resources

• **OWASP – Open Source Web App Security Project**
  – [www.owasp.org](http://www.owasp.org)
  – Lots of projects, papers, etc.

• **WebApp Sec mailing list**
Questions? Fill out Evals! Download slides!

- **Fill out the course eval**
- **These slides (and others) are online at** [www.MavenSecurity.com](http://www.MavenSecurity.com) (under **Resources** section)

- **Contact me at**
  - David Rhoades
  - david.rhoades@mavensecurity.com

- **Thank you**
www.MavenSecurity.com

Auditing web apps since 1996