How To Break Software Security

James A. Whittaker, Ph.D.
Professor of Computer Science
Florida Tech
Hacking Software

♦ Your adversary is prepared...are you?
  – Thousands of freeware hacking tools
  – Tens of thousands of hacker sites with tips, hints and tutorials

♦ The cold, hard truth: Hackers have the advantage:
  – They know assembly and C
  – They have more time than we do
The Art of the Hack

- The hacking “process”
  - Target selection
    - what are we going to attack?
  - Entry point identification
    - how are we going to get in?
  - Vulnerability detection
    - What bug will get us in?
  - Exploitation
    - What damage will we cause?
Target Selection

♦ What makes your application a valuable target?
  – It can be used (and therefore exploited) remotely
  – It protects something of perceived value
  – It is ubiquitous
  – It is being used by a customer who is a target
    • The Pentagon, FBI, …
Entry Point Identification

♦ What are the entry points that hackers use to exploit your applications?
  – Its UI
    • Logon screens, web front-end, …
  – Exposed remote functionality
    • DCOM, RPC, Web Services, …
  – Its communication paths
    • Communication protocols
  – The files it reads
    • Script files, corrupt data files, …
Vulnerability Detection

- Intended Behavior
- Actual Behavior
- Traditional Bugs
- Most Security Bugs
It’s a TESTING Problem

Techniques:
- Exploit external dependencies
- Find unanticipated user input
- Expose insecure design
- Determine insecure implementation practices

Tools
- The software *Holodeck*
External Dependencies
Dependency attacks

1. Block access to libraries
2. Manipulate registry values
3. Force the application to use corrupt files (includes write protected, inaccessible, physically corrupt etc.) and file names
4. Replace files that the application reads from, writes to, creates and executes
5. Force the application to operate in low memory/disk space/ network availability conditions
Dependency attacks

1. **Block access to libraries**
2. Manipulate registry values
3. Force the application to use corrupt files (includes write protected, inaccessible, physically corrupt etc.) and file names
4. Replace files that the application reads from, writes to, creates and executes
5. Force the application to operate in low memory/disk space/ network availability conditions
Dependency attacks

1. **Block access to libraries**

**DEMO: Browser content controls**
Unanticipated User Input
User input attacks

6. Overflow input buffers

7. Examine all common switches, options etc.

8. Explore escape characters, character sets and commands
User input attacks

6. **Overflow input buffers**

7. Examine all common switches, options etc.

8. Explore escape characters, character sets and commands
User input attacks

6. **Overflow input buffers**

**DEMO:** Flash, PPT Viewer, Media Player
Design Attacks
Design attacks

9. Try common default and test account names and passwords
10. Use Holodeck to expose unprotected test APIs
11. Connect to all ports
12. Fake the source of data
13. Create loop conditions in any application that interprets script, code etc.
14. Use alternate routes to accomplish the same task
15. Force the system to reset values
Design attacks

9. Try common default and test account names and passwords
10. Use Holodeck to expose unprotected test APIs
11. Connect to all ports
12. **Fake the source of data**
13. Create loop conditions in any application that interprets script, code etc.
14. Use alternate routes to accomplish the same task
15. Force the system to reset values
Design attacks

9. **Fake the source of data**

DEMO: E-commerce FREE shopping
Implementation Attacks
Implementation attacks

16. Get between time of check and time of use
17. Create files with the same name as files protected with a higher classification
18. Force all error messages
19. Use Holodeck to look for temporary files and screen their contents for sensitive information
Implementation attacks

16. Get between time of check and time of use

17. Create files with the same name as files protected with a higher classification

18. Force all error messages

19. Use Holodeck to look for temporary files and screen their contents for sensitive information
Implementation attacks

16. Get between time of check and time of use

DEMO: Breaking thru Digital Rights Management
Questions?

James Whittaker

Snail Mail
Florida Tech
150 West University Blvd
Melbourne, Florida 32901

E-mail & Web
jw@se.fit.edu
www.howtobreaksoftware.com

Telephone
321 674-7638

Fax
321 674-7046