Unforgivable Vulnerabilities

Steve Christey
The MITRE Corporation
August 2, 2007

MITRE

Introduction

- Vulnerabilities are a fact of life
- Many vulnerabilities simply shouldn't be in software anymore
- Everything's obvious to smart people like us!
- How to identify the worst of the worst?
- What issues should give pause to consumers, and nightmares to vendors?

Criteria for an "Unforgivable" Vulnerability

- **Precedence: Many have made the same mistake** — *Required*
- **Documentation: The mistake is well-documented** — *Required*

- **Obviousness: The attacks are obvious**
- **Attack Simplicity: The manipulations are very simple**
- **Found in Five: Able to be found with 5 minutes of effort**

2 of 3 Required

The Lucky 13

AAA…AAA

User/Password

Filenames

Common Commands

<SCRIPT>

User/Password

Body, subject, title,

to, from

‘

User/Password

id or other

numeric field

http://example.com/c99

Any include/require

that includes

$_GET, $_POST, etc.

“../..” or “/full/path”

Get/Send Command

File sharing

-rwxrwxrwx myprog

Executables

Libraries

Configuration Files

The Lucky 13 (Continued)

http://example/admin/script.cgi

Direct requests to admin scripts

tebj lbhe bja pelcgb

Grow-your-own crypto

User: shhh
Pass: shhh

Hard-coded usernames and passwords

Help

Selected from privileged Windows executable

Cookie: authenticated=1

Or in a form field

Length: 99999
Width: 99999

Selected from privileged Windows executable

ln –s /tmp/App.txt /etc/passwd
sleep 100000

“Turtle” symlink attack

Shall We Play a Game?

Do you have any more nominations?

See how long it takes one of these problems to show up in your favorite software.

See how many Black Hat talks in new technologies demonstrate these problems.

Typical Vulnerability History of a Product

- Obvious vulnerability types in critical functionality
- Incomplete fixes for these vulnerabilities, often ignoring closely related vectors
- Variants of common vulnerability types
- Types restricted to rare environments, platforms, configurations, or features
- Elimination of most common types
  - Systematic code analysis or refactoring
- Susceptibility to rare or novel vulnerability types and attacks
  - Rarely detectable automatically
- Unique vulnerability types requiring expert analysis and lots of time

VAAL-idation

- Vulnerability Assessment Assurance Levels (Litchfield)
  - Based on an audit of a product
  - Depth of analysis: “how much effort was put into analysis?”
  - Level of confidence: “how secure is software X?”
- Moves away from those pesky vulnerability counts
- Communicate to consumers and each other

The professional research community needs to stop pretending that basic research is magic and become more consumer-friendly.

Proposed VAAL Dimensions

- Access Constraints
  - Privileges/restrictions needed for access
- Feature Frequency
- Potential Severity
- Novelty
  - How new/unusual is the vuln/attack?
- Vector Depth
  - How “close together” are the entry point with the vulnerability?
- Manipulation Complexity
  - <SCRIPT> or RSnake head-scratcher?
- Ubiquity
  - Configuration, Platforms, Environments
- Level of Effort
  - Shhhh, never let them see you sweat

Unforgivable Vulnerabilities in VAAL-speak

- Low access constraints
- Very high feature frequency
- Very low novelty
- Low manipulation complexity
- Low level of effort

- Not directly applicable
  - Potential severity
  - Vector depth
  - Ubiquity

Related Work

- Attack Surface Measurement (Howard, Manadhata, and Wing)
- Threat Modeling (*Trike*, *STRIDE*, Snyder)
- SAMATE: Software Assurance Metrics and Tool Evaluation (NIST)
- Security Quality Score (Wysopal/Veracode)
- CVSS: Common Vulnerability Scoring System (FIRST)

- See paper for details and more references

Questions?

Extra Slides

The Lucky 13

- **Buffer overflow using long strings of “A” characters in:**
  - **Username/password during authentication**
  - **Arguments to most common features of the product or product class**
- **XSS using well-formed SCRIPT tags, especially in the:**
  - **Username/password of an authentication routine**
  - **Body, subject, title, or to/from of a message**
- **SQL injection using ' in the:**
  - **Username/password of an authentication routine**
  - **“id” or other identifier field**

The Lucky 13 (Continued)

- Remote file inclusion (RFI) from direct input
  - include($_GET['dir'] . "/config.inc");
- Directory traversal in GET/SEND commands
  - “../..”
  - “/full/path/name”
  - File sharing, web server, chat client
- World-writable critical files
  - Executables
  - Libraries
  - Configuration files

The Lucky 13 (Continued)

- Direct requests of administrator scripts
- Grow-your-own crypto
- Authentication bypass using "auth=1" cookie/form field
- Turtle race condition (symlink)

The Lucky 13 (Continued)

- **Privilege escalation launching "help" (Windows)**
- **Hard-coded or undocumented account/password**
- **Unchecked length/width/height/size values passed to *malloc()*|*calloc()***