Web Application Vulnerability Assessment

Discovering and Mitigating Vulnerabilities in Web Applications

14 October 2005

CYBERTRUST

Introductions & Approach

- Kris Philipsen – Security Engineer at Ubizen/Cybertrust in Luxembourg (kristof.philipsen@cybertrust.com)

- Impossible to discuss all security issues affecting web applications in just 45 minutes !!!

  So … security issues depicted based on findings of selected real-life Web Application Vulnerability Assessment scenarios.

- Scaled down slides. Full slides available at:

  http://www.ubizen.com/download/Hack-LU_2005_K_Philipsen.pdf

Agenda

- Web Application Overview
- Web Application Vulnerability Assessments
- Discovering Security Issues in Web Applications
- Mitigating Security Issues in Web Applications

Web Applications – An Overview

- Found everywhere nowadays (e-Banking, e-Commerce, Corporate Internal Web Apps)

- Are non-trivial in nature and require specific security requirements.

- Generally based on 3-tier model:
  - Client – Browser software (end user system)
  - Web Application – Performs **gateway** functions between client and database
  - Data Provider – Usually **database** server containing user data.

- Web Applications come in two flavors:
  - Consulting Only
  - Transactional

What makes web applications vulnerable?

- Common availability and access
- Reliance on other architecture components
- Designed without a maximum of security considerations
- Abundance of programming languages
- “If it ain’t broken don’t fix it” / “If it’s still running don’t touch it” philosophy

Need for Web App Vulnerability Assessments

- Vulnerability Assessment Tools available for free.
- Tools do not detect unknown or less-known vulnerabilities.
- Issues difficult to detect by automated tools (just to name a few):
  - Authentication De-synchronization
  - Access Control Issues
  - Temporary and User-specific Files
  - Session State

Discovering Security Issues in Web Apps

- Automated Discovery (AD)
  - Automated tools look for known patterns and vulnerabilities
  - Well-known vulnerabilities, XSS, SQL Injection, Information Disclosure, Command Execution, Buffer Overflows, …
- Manual Discovery (MD)
  - Still uses tools but person decides on approach
  - Access Rights, Access Control, Session State, Temporary and User-specific Files, Trust Relationships, Database Audits, …
- Hybrid Discovery
  - A best breed combination of automated and manual discovery

Tools of the Trade

- Web Browser
  - Whichever you feel most comfortable using
- Web Vulnerability Scanners
  - Nikto, AppScan, Nessus, …
- Web Security Tools
  - Stunnel, OpenSSL, …
- Web Proxies
  - Paros, WebScarab, Achilles….
- Command-line Programs, Interpreters, and Compilers are your friend!
  - Netcat, Perl, …

CYBERTRUST

Before we begin … Web App Vulnerabilities

- Just some statistics for the last three days: 20+ Web App Vulnerabilities

- These are just a few of them …

  - [13 Oct 2005] YaPig Homepage Form Field HTML Injection Vulnerability
  - [12 Oct 2005] PHPWebSite Search Module SQL Injection Vulnerability
  - [12 Oct 2005] Sun Java System Application Server JSP Source Disclosure Vulnerability
  - [12 Oct 2005] VERITAS NetBackup Java User-Interface Format String Vulnerability
  - [12 Oct 2005] Zeroblog Thread.PHP Cross-Site Scripting Vulnerability
  - [11 Oct 2005] Accelerated E Solutions SQL Injection Vulnerability
  - [11 Oct 2005] VersatileBulletinBoard Information Disclosure Vulnerability

Injection Issues

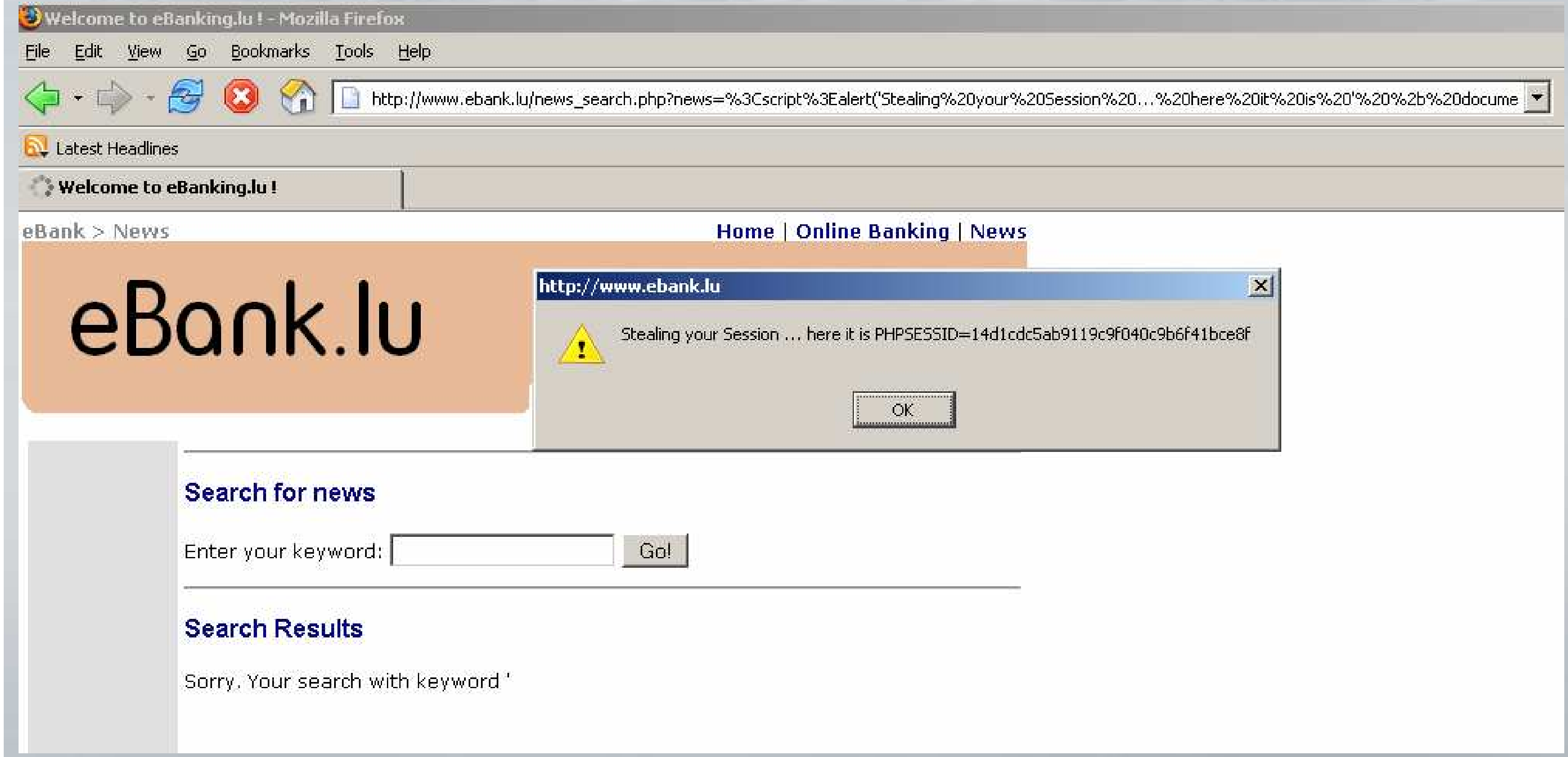
- Cross Site Scripting (AD)
  - Attacker can write content into document
  - Web Application is susceptible, but the user is the target
  - Due to improper input validation
  - Facilitates cookie stealing & session hijacking

- SQL Injection (AD)
  - Attacker can gain access to the database (SELECT, ALTER, DELETE,…)
  - SQL injection issues are often seen in:

    - Login Prompts

    - Database Generated Tables and Content (i.e. Transaction listings in e-banking systems).
  - Due to improper input validation and database access control issues
  - Facilitates authentication bypass, arbitrary content viewing, …

Example: Cross Site Scripting



Welcome to eBanking.lu ! - Mozilla Firefox

http://www.ebank.lu/news_search.php?news=%3Cscript%3Ealert('Stealing%20your%20Session%20...%20here%20it%20is%20'%20%2b%20docume

File Edit View Go Bookmarks Tools Help
Latest Headlines
Welcome to eBanking.lu !
eBank > News
Home | Online Banking | News
eBank.lu

http://www.ebank.lu

Stealing your Session ... here it is PHPSESSID=14d1cdc5ab9119c9f040c9b6f41bce8f

Search for news

Enter your keyword: Go!

Search Results

Sorry. Your search with keyword '

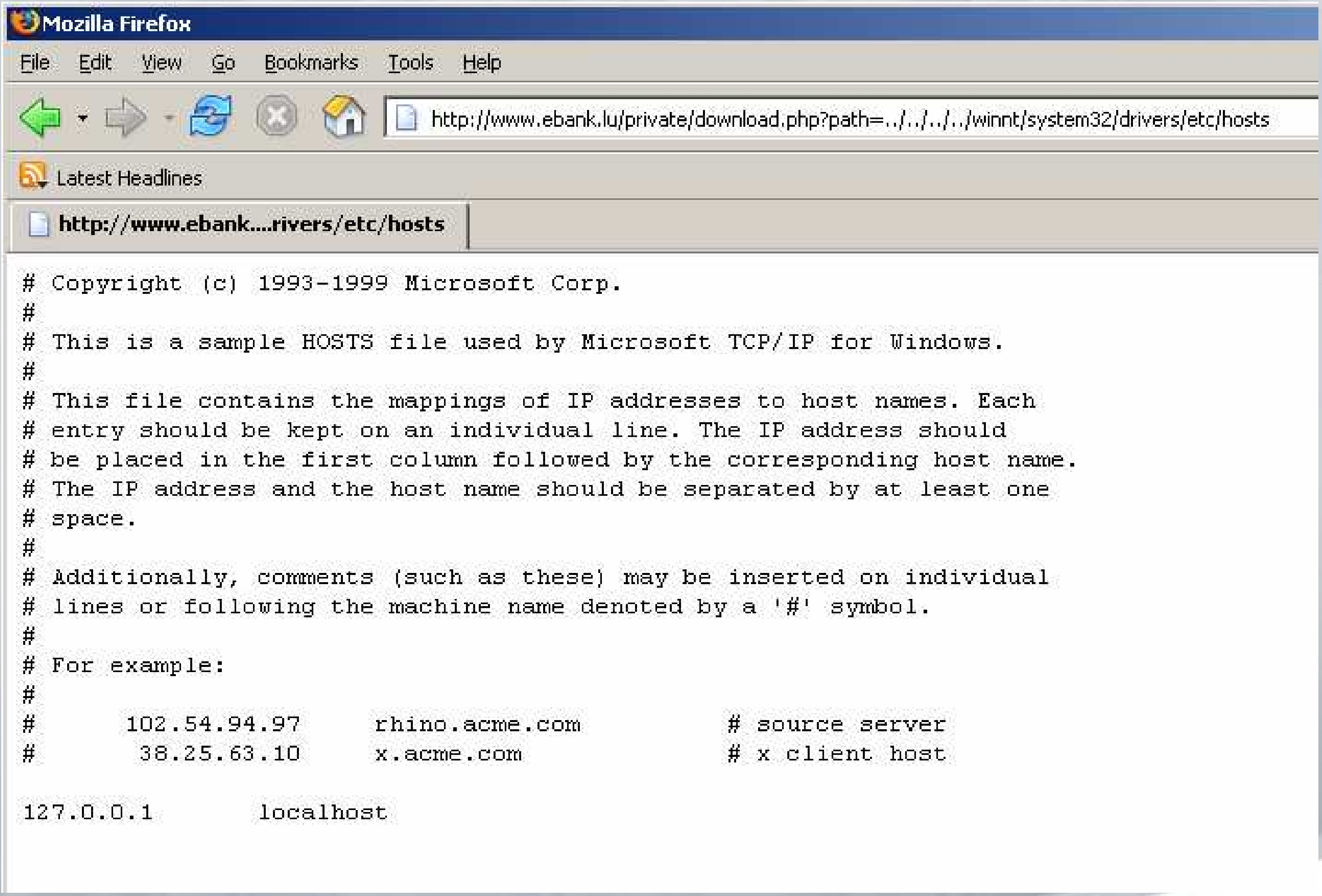
CYBERTRUST

Disclosure Issues

- Directory Indexing (AD)
  - List files in directory
- Directory Traversal (AD/MD)
  - Attacker can traverse through different directories
- Arbitrary File Disclosure (AD/MD)
  - Attacker can view files stored at arbitrary locations (i.e. “c:\winnt\system32\repair\sam”, “/etc/passwd”)
- Error Messages (AD/MD)
  - Attacker can gain potentially harmful information from system error messages and debug traces.

CYBERTRUST

Example: Arbitrary File Disclosure


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Latest Headlines
http://www.ebank....rivers/etc/hosts

http://www.ebank.lu/private/download.php?path=../../../../winnt/system32/drivers/etc/hosts

```
# Copyright (c) 1993-1999 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

127.0.0.1       localhost
```

Execution and Denial-of-Service Issues

- Command Execution (AD)
  - Attacker can execute commands on system through manipulation of URI parameters (i.e. “cmd.exe”,”/bin/ls”,”/bin/cat”).
  - Improper input handling and interpretation
  - We’ve all seen Unicode and PHF (old school!)
- Buffer Overflows (AD)
  - Attacker can execute commands or gain access to the system by overwriting portions of non allocated memory using attack code.
  - Improper buffer handling
  - Apache Chunked Encoding, Microsoft IIS .HTR Overflow
- Denial-of-Service (AD)
  - Attacker can render the entire or parts of the web application unusable through resource starvation. Other types of DoS exist!

Other Well-Known Issues

- Default Values and Scripts (AD)
  - Defaults are the mother of all evil!
  - Anyone can stage such an “attack”
  - Due to negligence

- Brute Forcing (AD)
  - Generally known for username/password brute-forcing…
  - However, why not brute force Session IDs, temporary files, etc...
  - Due to improper policy controls

Trust Issues

- Authentication Issues (MD)
  - Attacker exploits inherent holes in web application resulting in authentication mechanism bypass
  - Authentication De-synchronization Attack in Multi-level Authentication Systems
- Access Control Issues (MD)
  - Access control measures may not be implemented correctly allowing attackers to defeat authorization mechanisms.
  - Parameter Tampering in Resource Authorization Systems
- Trust Relationship Issues (MD)
  - Attacker takes advantage of the fact that one web application infrastructure component trusts another component entirely or to a certain degree.
  - Trust Delegation Attack in Web Application Database Environments

Session Issues

- Session Generation Issues (MD)
  - Predictability of Session IDs can lead to hijacking attacks
  - Different attacks against Session ID generation algorithms:
    - Statistical Time-based Attacks
    - Statistical Modulus-based Attacks
    - Predictability issues in Session ID Algorithms (i.e. WebSphere, iPlanet,…)
- Session Validation Issues (MD)
  - Poorly designed Session Validation mechanisms allow attacker to carry out hijacking attacks.
  - State of the entire original session is not validated upon each request.

Data Processing Issues

- User-specific Content and Files (MD)
  - Users can generate temporary contents (CSV, PDF, XLS, DOC) on web app
  - Improper data processing and access controls expose all!

- Temporary Application Content and Files (MD)
  - Temporary files may be generated during data processing
  - Improper clean up exposes sensitive information

Other Less Well-Known Issues

- Database Auditing (MD)
  - Take SQL injection one step further
  - Build a "pseudo-layout" of database
  - Assist in launching further attacks
  - Example:

```
Warning: Sybase: Server message: Server user 'webuser' is not a valid user
in database 'model'. (severity 14, procedure N/A) in /www/stockinfo.php on
line 64

DB errorServer user 'webuser' is not a valid user in database 'model'.
```

Mitigating Security Issues in Web Apps

- Core Web Application Security
  - Secure core components of web application infrastructure directly
  - Difficult to implement if not included in initial design phase
- Peripheral Web Application Security
  - Peripheral infrastructure security secures the core components
  - Can be implemented at later stages

Securing Core Authentication Processes

- Securing the Authentication Process
  - Multi-level User Authentication Systems
    - Available authentication methods (choose them wisely!)
    - Authentication synchronization (don’t make the common mistake!)
  - Centralized vs. Decentralized Authentication Systems
  - Authentication Mechanisms:
    - Username and Password based Authentication
    - Client Certificate based Authentication
    - Challenge Response, Time Synchronous, and One-Time Password Authentication
    - Biometric Authentication Systems (Privacy Issues?)

Securing Core Authorization Processes

- Securing the Web Application Authorization Process
  - Based on user profiles stored on app server or on centralized AAA server
    - Authentication Credentials and Type
    - Authorization Scope, Paths, Timeframe, Sources, …
  - Scalability issues with per-user authorization profiles – Solution: Group Profiles

- Securing the Data Provider Authorization Process
  - Based on “roles” defined in database or on centralized AAA server
  - More granular control, not only what a user has access to but also what kind of access:
    - Authentication Credentials and Type
    - Authorization Scope
    - Authorization Rights (INSERT, SELECT, DELETE, etc…)

CYBERTRUST

Securing Core Accounting Processes

- Implementing Accounting
  - Client Accounting (Difficult to implement? Privacy Issues?)
  - Web Application Accounting (Custom Application Logs)
  - Data Provider Accounting (Security and SQL Transaction Logs)
- Accounting Security Challenges
  - Account data encrypted?
  - Stored locally or centralized?
  - Who has access? How is access control enforced? (i.e. key splitting)

CYBERTRUST

Securing Interconnections & IPC

- Securing Communication Schemes
  - Application-specific guidelines on which process/function can talk with which other process/function based on predefined prerequisites
  - Define boundaries of communication channels on shared communication planes
- Securing Trust Relationships
  - One component will allow delegation of a specific action (authentication, authorization, data processing, …) to another if a certain set of criteria is met
  - Trust is a very dangerous thing! You need a **REALLY** good motivation!
- Securing Traffic Flows
  - Protocol-specific guidelines on inter-component communications to accommodate communication schemes and trust relationships over network infrastructures
  - Implement secure transport over insecure medium:
    - Transport Authentication, Confidentiality, Integrity & Replay Detection

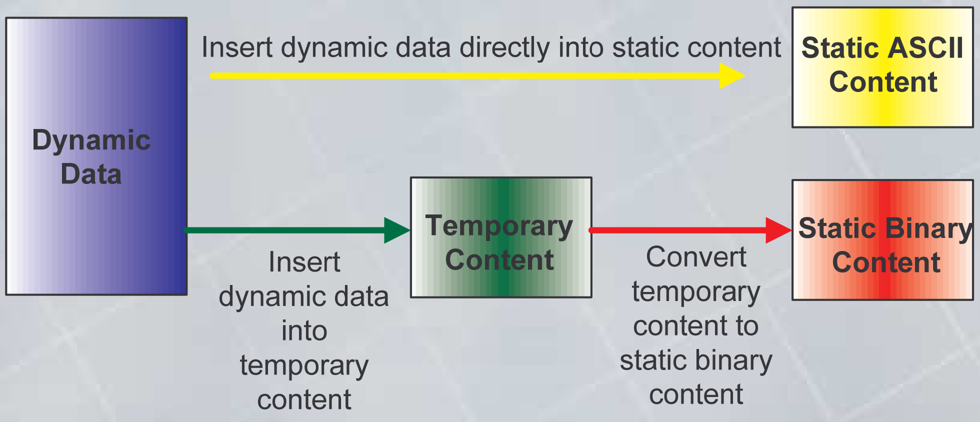
Securing Sessions

- Concept of Sessions
  - • Sessions manage unique user connections (Session table on server-side, Session ID/cookie on client-side).
- Securing Session ID Generation
  - • Use Session ID API functions available in web server:
    - - Easily susceptible to reverse engineering due to widely available web server platforms
  - • Challenges for writing Custom Session ID algorithms:
    - - Which cryptographic algorithms to use?
    - - How should session tables be constructed?
    - - Different Session ID for different parts of web app?
- Securing Session Validation
  - • Don’t underestimate the power of social engineering! (XSS, Phishing, …)
  - • Validate sessions to multiple parameters but beware!

CYBERTRUST

Securing Data Generation

- Dynamic Data Concepts
  - Web App transforms dynamic data into static ASCII content or binary content


Dynamic Data
Insert dynamic data directly into static content
Static ASCII Content
Temporary Content
Static Binary Content
Insert dynamic data into temporary content
Convert temporary content to static binary content

- Securing Dynamic Data Generation
  - Prevent temporary files on web application (delegate but with care!)
  - Ensure limited lifetime
  - Deploy end-to-end authorization for dynamic data generation

Securing Data Storage

- Securing Dynamic Data Storage
  - Apply access control (i.e. store data on the data provider (GLOBs) validate)
  - On-demand content, one-time mappings, pseudo-random file IDs, …
- Securing Client-Side Data Storage
  - Persistent vs. non-persistent data
  - Client-independent autonomous environments
- Securing Server-Side Data Storage
  - Protect against physical data compromise (logical/hardware encryption)
  - Protect against logical data compromise (confidentiality and integrity checks + limit access to system)

Securing Core Components

- Hardening Client Systems
  - Hybrid in nature, difficult to implement.
  - User awareness campaigns

- Hardening Web Application and Data Provider Systems
  - Security best practices should be followed.
  - As for the web application systems:
    - Use chrooted or self-contained environments on Unix systems
    - Don’t forget those permissions (especially on web server – mount read-only if possible!)
    - Perform proper authentication and access control
    - Set up proper audit trails

Secure Code Development

- Writing Secure Code
  - Avoid the common issues!
  - Tools for detecting common coding insecurities (ITS4/Lint, CodeSpy)
- Securing Input Functions
  - Secure HTTP Header Input (compliance, strip characters, limit methods)
  - Secure HTTP Request Input (check encodings, know your content length)
- Securing Output Functions
  - Secure Regular Application Output (Detect and drop anomalies)
  - Secure Error Output (Don’t give the attacker the edge or the incentive)
- Security Implications using Third Party Libraries
  - Vulnerabilities in 3rd Party Libraries may directly or indirectly affect the web application.

Secure Code Development (2)

- Using Security Libraries
  - Perform input and output security through a predefined set of libraries and APIs
  - Stringer for J2EE, PHPFilters for PHP.
- Code Auditing
  - Crucial part of Q&A testing
  - Have a fresh set of eyes look at the code

Peripheral Web Application Security

- Network-level Firewalls
  - Provide separation and layered approach to web application infrastructure.
- Reverse Proxies and Application-level Firewalls
  - Reverse proxies break flow between client and web applications. Provide data sanity and basic security checking.
  - Application-level Firewall go one step further; allow for granular rules to be created. Oftentimes support “learning” mode.
- Intrusion Detection and Prevention Systems
  - Intrusion Detection Systems take a passive approach and alert vulnerabilities
  - Intrusion Prevention Systems take an active approach and drop suspicious.
- Web Server Security Modules
  - Apache with mod_security / Microsoft IIS with SecureIIS

CYBERTRUST

Workshop

- Still not bored? ;) – Workshop at hack.lu
- What's it all about?
  - Hands-on experience with web application vulnerabilities and web application security on a mock-up e-banking infrastructure.
- Which vulnerabilities can be put to the test?
  - SQL Injection
  - Cross Site Scripting (XSS) + Session Hijacking
  - User Input Validation Issues
  - Parameter Tampering
  - Server Mis-configuration
  - Data Generation Issues

  … and more!

Q&A