Introduction into Web Application Security

Tomáš Polešovský
14. 01. 2014
tomas.polesovsky@liferay.com

Introduction to Web Application Security

Agenda

- Security Overview, wider picture introduction
- Common concerns and general security concepts
- Most common vulnerabilities and their countermeasures

Security

- Is a very broad and deep topic,
  - from privacy and freedom to strict control and monitoring
  - from defacing the Birds site to breaking Iran's nuclear program using Stuxnet
- Is controversial, touches morale and ethics of each individual
- Is about assets, threats, vulnerabilities, attacks and protection
- Is a measure, from 0 to infinity
- Is an aspect of quality, must be balanced with other quality aspects like usability and performance
- Nothing and nobody is ever secure

Security & Hacking Terminology

- Hacking is seeking and exploiting a vulnerability to gain unauthorized access to protected asset
- Vulnerability is a bug in software, security control or missing security control at all. Mostly:
  - The code is not doing everything it should
    - Doesn't validate uploaded file name (../../my.doc)
  - The code is doing more than it was designed
    - Allows file upload, even JSP files (test.jsp)
  - Or Both
    - "`../../webapps/ROOT/test.jsp`" => Remote Code Execution

Security & Hacking Terminology

- Exploit
  - a script or tool exploiting one or more vulnerabilities
  - contains payload = attack vector
- Hacker is:
  - Newbie, Script Kiddie, White Hat (ethical hacker), Black Hat, or something between (Gray Hat)
  - Motivated by fun, money, power or other reasons
  - Very creative, thinks the way others don't
  - Most of the time sitting, reading, testing, reading, testing, reading, testing, reading ...

Security & Hacking Terminology

- Attack surface
  - List of all interaction interfaces between application and the rest of the world accessible by the attacker
    - ... where the attacker can hit the application
    - For example HTML interface, Remote APIs, File System
- Security Incident
  - A security event that compromises the integrity, confidentiality, or availability of an information asset
- Breach
  - An incident that results in the disclosure or potential exposure of data
- Data disclosure:
  - A breach for which it was confirmed that data was actually disclosed (not just exposed) to an unauthorized party

Security & Responsibility

- When you find a vulnerability:
  - Non Disclosure:
    - You find bug, tell vendor, he may fix it or ignore you
    - Or you find bug and don't tell vendor, sell the exploit
  - Responsible Disclosure:
    - You give a chance to vendor to fix it
    - You have the right to publish it when vendor ignores you or delays patches too much
  - Full Disclosure
    - You go public, vendor has the same reaction time as attackers
    - Customers have the chance to decide (if they are able to understand)
  - You may go into jail for hacking

Security & Hacking & Law

- Hacking is a crime in most civilized world countries

  Any sufficiently advanced technology is indistinguishable from magic

  A. C. Clark's Third Law

- People are afraid of what they cannot understand:

  -> 99% of hackers are bad guys

  -> Hacking is surrounded with a mystery, hackers are a secret society

  -> Witch-hunting for hackers across globe

  That's WRONG!

  => Hacking attracts children and teenagers that can easily cross the line

  => It's easy to be a bad guy and hard to be a good one

  - It's possible to report any bug without worries, except security bugs => **JAIL**

  => How can we protect internet for our children when we cannot help each other?

Hacking & Ethics

- Knowledge is a weapon you always have with you

  Q: Would you tell your children's school that you were able to “*get*“ their pupils records with photos and addresses?

Hacking & Ethics

Warning: This slide is my subjective point of view :)

Q: Would you tell your children's school that you were able to “*get*“ their pupils records with photos and addresses?

My answer: Yes. I would.

- What I believe in:
  - Moral principles should stand above law
  - *“Don't be afraid and don't steal”* (T.G.Masaryk, first president of Czechoslovakia)
  - No harm, no fool
  - Responsibility, better be safe than sorry
  - Adult is able to know when (s)he crossed the line, teenager/children not
  - Education that removes that magic cloud surrounding IT security

Web App Sec

End of introduction

Questions?

Next:
Common Concerns and General Security Concepts

Web Apps

- Web Applications are good targets
  - Are easy to attack, all you need is a browser
  - Are hard to secure, lots of different technologies
  - Are attractive because are visible

http://www.theguardian.com/angry-birds-defaced-nsa-spying-birds-user-data

Common Concerns and Risks

- Everything is driven by damage caused by security incidents
- The damage means losing important assets
  - User accounts
  - Legal documents
  - Bitcoins
- Solution is simply to be prepared for attacks
  - Minimize possible attack surface
  - Fix simple and known vulnerabilities
  - Stay aware for new vulnerabilities and attack attempts

General Attacks Description

- Anonymous attacks
  - Mostly exploits known vulnerabilities using automated scanners
  - Attack the web using the most common vulnerabilities
  - Targets end users and corporate network to:
    - Get user accounts, personal data and emails
      - To be sold (spam, phishing, malware distribution, fake driving licenses, credit cards, etc.)
    - Get into end users computers to plant trojans/malware
    - Abuse infrastructure to execute attack on other targets
- Direct attack on the company
  - Hard to challenge, tailored
  - Not very common - expensive
  - Attack surface is the whole company
    - Employees – social engineering
    - Mobile devices, laptops, computers
    - Infrastructure – internal and external systems
- Attackers
  -

General Web App Sec Concepts

- It's not possible to secure everything
  - Any security control can be defeated with enough resources
  - Security incident/breach is like Judgment Day. Nobody knows when it comes, but be sure it comes. Be prepared

- Security bugs will be found, don't try to be perfect
  - Give security researchers possibility to disclose issues responsibly (e.g. security@liferay encrypted emails)
  - Don't hide reported bugs, give credit (or bounty) to those who helped to protect your business
  - Offer security patches

- **Trust/credit is everything**
  - With trust and credit, imperfections are politely accepted
  - Once it's lost, there's long and painful way back, if any

Be open, honest, do your best and hope it will be enough

General Web App Sec Concepts

- Security by default
  - When application installs / starts it must run in secure configuration, it may provide options for users/customers to make it less secure
- Principle of least privilege
  - Code should run with the lowest possible permissions allowing it to execute given task
  - User should have the lowest possible permissions to perform the job
- Defensive programming
  - Design by contract – assumptions and undefined behavior
  - Check input and output
  - Security controls on application boundaries
  - Don't trust anything unless proved otherwise
    - Sanitized input, escaped output, signed code

General Web App Sec Concepts

- Whitelists, not blacklists
  - Specify what is allowed, not what is disallowed
- Chain is as strong as its weakest link
  - It's common to combine several small vulnerabilities
  - One small vulnerability mostly open door to other possibilities

General Web App Sec Concepts

- Prevention of security bugs in source code
  - Security aware design and programming
  - Internal security reviews and tools
  - Release security penetration tests
  - External Blackbox & Whitebox services (Veracode)
- Prevention of security bugs in 3rd party libraries
  - Make sure they care about security
  - Use latest secure version
  - Fix it yourself :(
- Mitigation of security incidents on production
  - Intrusion Detection System
    - Detects non-standard behavior and is able to report that activity
  - Intrusion Prevention System
    - Prevents exploiting vulnerabilities based on known attack payloads
  - Web Application Firewall
    - Complex rules, can be customized for every application
    - For example: http://en.wikipedia.org/wiki/ModSecurity
  - Installing patches and keeping systems up-to-date

General Web App Sec Concepts

- Types of security tests
  - Security review
    - Done manually as part of development lifecycle
  - Automated vulnerability scan
    - Black-box test – without source code
      - Fuzz testing
    - White-box test – with source code
  - Penetration testing
    - Simulates attack

Web App Sec

End of general and common concepts

Questions?

Next:
Vulnerabilities and countermeasures

Web App Vulnerabilities

- OWASP Top 10 (2013)
  - Open Web Application Security Project
  - Most frequent security issues
- CWE/SANS Top 25 (2011)
  - Common Weakness Enumeration
  - Top 25 Most Dangerous Software Errors

Injection

- SQL injection, OS Command Injection, others
- Validate and sanitize input

- “DELETE from invoices where id= “ + id;
  // id ... 1 **or 1=1**

```
conn.prepareStatement(query).setString(1, id);
```

- Runtime.exec(“avscan /tmp/“ + name”);
  // name ... test.doc**;rm -rf /**

```
if (name.matches(“[a-zA-Z0-9 \.]+”)) { … }
```

Cross-Site Scripting (XSS)

- Used to trigger an action in user's browser, for example alert(1)

  `payload = '"><svg onload=alert(1)>`

- Reflected XSS (from URL)

  http://site/editEntry?name=payload

  <h1><%= request.getParameter("name") %></h1>

- Stored XSS (saved in DB)

  <h1><%= entity.getName() %></h1>

- DOM XSS (executed by JS)

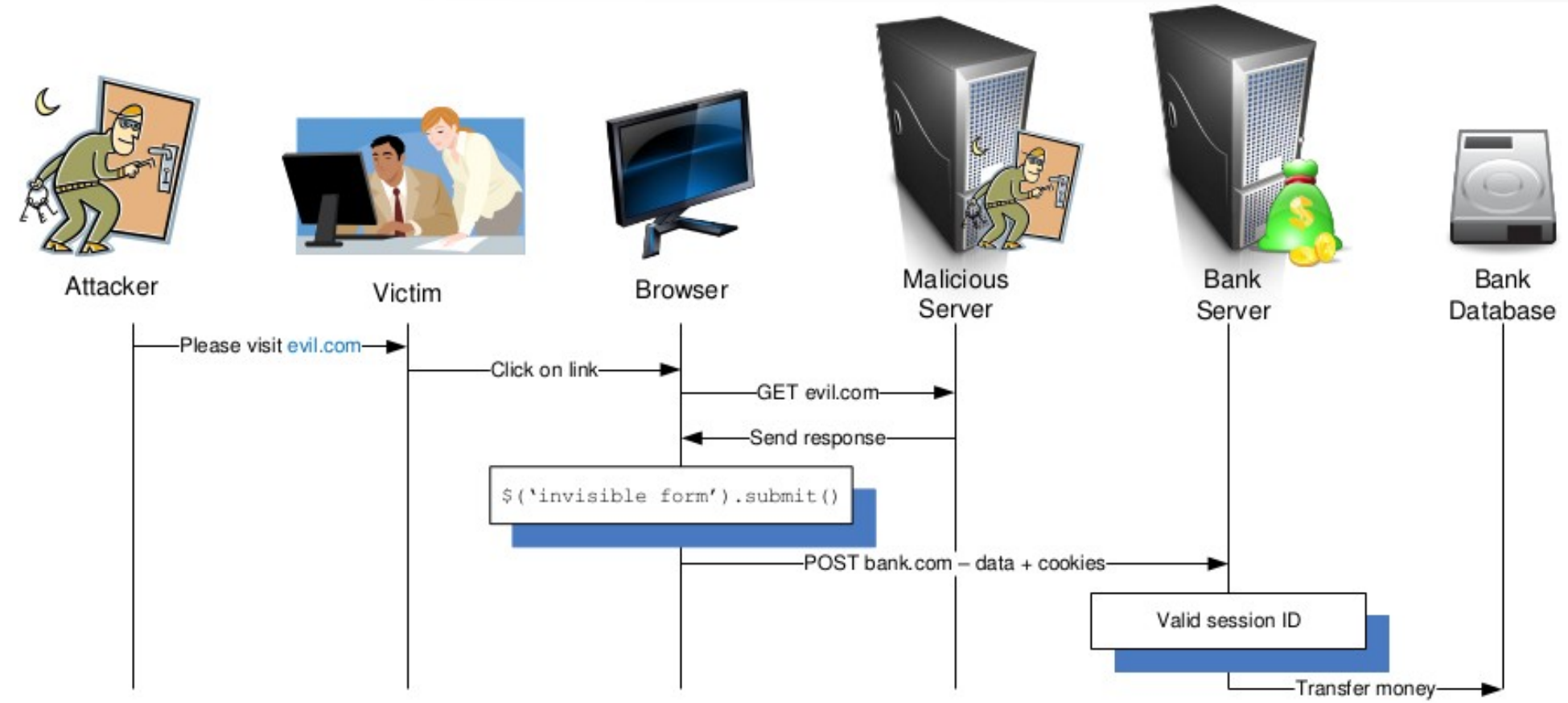
  http://site/search#payload

  document.getElementById("searchQuery").innerHTML = location.hash

XSS (Cross-Site Scripting)

- Countermeasure:
  - Context-sensitive output escaping
  - Escaping must be done in the place of writing output
  - Escape all output that is not static
- Example:
  - HTML Body: < → < “ → " ' → '
  - HTML attribute: &#xHH; (hex)
  - JavaScript: \uXXXX (unicode)
  - CSS: \HHHHHH (hex)
  - URL: %HH (hex)
- https://www.owasp.org/index.php/XSS_%28Cross_Site_Scripting%29_Prevention_Cheat_Sheet#XSS_Prevention_Rules_Summary

Cross Site Request Forgery (CSRF)



CSRF (Cross Site Request Forgery)

- CSRF or XSRF
- Countermeasure:
  - *Synchronizer Token Pattern* or Anti-CSRF token or simply CSRF token
  - Is present in (1) request and (2) cookie/session, server verifies that the tokens are the same
  - Attacker doesn't know the token => cannot create valid request
  - Is needed only for URLs that change state on server
  - Should not be inside URL … can be disclosed by Referer header
- https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet#General_Recommendation:_Synchronizer_Token_Pattern

Missing Security HTTP Headers

- Clickjacking
  - Countermeasure: X-Frame-Options: sameorigin
- XSS via Content Type sniffing
  - Countermeasure: X-Content-Type-Options: nosniff
- Application cookies can be read by browser (XSS)
  - Countermeasure: HTTPOnly cookie flag
- Cookies established for HTTPS are used for HTTP
  - Countermeasure: Secure cookie flag

Missing Authorization

- Private information is disclosed by public function
- Access to records using URL manipulation (changing “ID” in the URL)
- Accessing or modification of data using remote API
- Countermeasure:
  - Verify all accessible information have been checked for permissions

Broken Authentication

- Some pages/functions can be accessed without authentication … common issue with home routers
- Passwords are not stored safely (tokens leaks into server logs)
- Incorrect reset password functionality

  https://www.liferay.com/community/forums/-/message_boards/view_message/39345927

Unvalidated Redirects and Forwards

- http://site.com/redirect?url=evil.com/login
- Opens doors for phishing attacks

  `response.sendRedirect()`

  `RequestDispatcher.include/forward()`
- Countermeasure:
  - Validate input and whitelist allowed URLs

Path Traversal

- Also known as Local File Inclusion
- Allows to load and display a local file
- http://site/page/convert?file=../../../../etc/passwd
- Countermeasure:
  - Validate input and use File.getCanonicalPath()

Remote Code Execution (RCE)

- Application executes payload
- Examples:
  - Uploaded JSP file, WAR file
  - JVM Scripting executes forged validation routine
  - Freemarker/Velocity/XSLT scripting
- Countermeasure:
  - Validate uploads
  - Check all dynamic execution points
  - Apply whitelisting to allow only safe methods in scripting

Denial Of Service (DoS)

- Application eats resources and doesn't react
  - CPU – long operation blocking a thread on the server
  - Memory – too big variable in HTTP session
  - Disk – big log files
  - DB – tons of rows in one table
- Countermeasure:
  - Performance testing
- DDoS – Distributed DoS
  - Launched from a botnet
  - Launched using amplification attack
  - Countermeasure: network configuration and IPS

Sensitive Data Exposure

- Unencrypted communication (HTTP), mixed HTTPS
- Storing username+password into cookies unencrypted
- Sensitive data are not saved properly (credit card numbers leaking in log files)
- Countermeasure:
  - Employ HTTPS, everywhere if possible
  - Review code to free or encrypt sensitive data as soon as possible

Use of Hard-coded Credentials

- Credentials are stored in code
- Credentials are stored in configuration files, have default values
- Countermeasure:
  - Remove them from code and store safely!

Security misconfiguration

- Insecure configuration by default
- Disabled important security feature
- Countermeasure:
  - Review application configuration
  - Have up-to-date 3rd party frameworks and components

Where to continue / sources

- List of vulnerabilities / errors: http://cwe.mitre.org/
- https://www.owasp.org/index.php/OWASP_Cheat_Sheet_Series
- Security Conferences, videos
  - DEF CON Germany - https://www.youtube.com/user/defconvidoes
  - Black Hat USA - https://www.youtube.com/channel/UCJ6q9Ie29ajGqKApbLqfBOg
- Twitter and Blogs of security experts
  - search for “top 10/20/100 security experts to follow” :)
- Full disclosure mailing list, web app security mailing list
  - http://insecure.org/news/fulldisclosure/
  - http://www.securityfocus.com/
- Verizon data breach reports
  - http://www.verizonenterprise.com/DBIR/

Introduction to Web App Security

That should be enough for the start :)

Thank you.

Questions?