Mag. iur. Dr. techn. Michael Sonntag

Securing webpages

An Overview

i-Society 2010, 28-30.6.2010, London

Institute for Information Processing and
Microprocessor Technology (FIM)
Johannes Kepler University Linz, Austria

E-Mail: sonntag@fim.uni-linz.ac.at
http://www.fim.uni-linz.ac.at/staff/sonntag.htm

Introduction

- E-Learning occurs predominantly on the WWW
  - → Which is notoriously insecure in various aspects, especially regarding copyright violations and privacy
- Many stakeholders require security, …
  - → Producers of learning material: No unauthorized copying
  - → Teachers: No cheating (from materials, other learners, …)
  - → Learners: Not everyone should see all their actions in detail

… but it is a different kind for each!

- Solution: Technological protection measures
  - → But there's a snag: Producing circumventions for them is often difficult, but using them is typically trivial!
    - » So legal protection exists (EU directive 2001/29/EC), but this requires a protection measure to be "effective"

What technological measures exist and how good are they?

Targets of web site protection

- High-level targets of web site protection are:
  - → Obtaining, keeping, reducing decrease of revenue
    - » Ensuring advertisements are shown/clicked
  - → Preventing others from obtaining, keeping, reducing decreases of revenue through employing your web site
    - » Copying content, framing, link stealing, …
  - → Protecting security measures
    - » Encrypting JavaScript code which prevents cheating in exams
  - → Keeping the content "secret" (→ better: restricted)
    - » Only certain people are allowed to access it, e.g. learning materials you have to pay for
  - → Marking content for tracing
    - » Discovering the source of infringement, like who distributed the recording of an online lecture; combating plagiarism
- Many stakeholders exist; they vary in their interests

Protecting a web page's elements

- Media files and apps: Images, videos, animations, applets,…
  - → Typical problem: Copying, saving for offline use
- HTML source code: Interesting for designers and attackers
- CSS: Good designs are hard to develop
- JavaScript code: Functionality, security
  - → Ajax is especially important: Method/content of communication
- E-Mail addresses: Harvesting for spam
- Page text: Copying content for exercises or reuse
  - → Especially problematic: Fully automated crawling & extraction
- Drag&drop / copy&paste: Copy solution instead of formulating it yourself
- Advertisements: Circumvent ad-blockers, overlapping
  - → Techniques legitimately used in educational content!

Integrity of presentation

- Preventing whole page/site download
  - → No offline use - only online; security often doesn't work offline
- No copies: Preventing printing, screenshots, caching
  - → Privacy → Embarrassing chat messages/pictures/…
- Keeping page existence secret (→ solutions)
- Avoid content filtering matches
  - → Includes presenting special pages to indexing bots
- Only registered/human users
  - → Preventing visits by bots
  - → Declining automated content submission
- Deny access to certain browsers
  - → Because of bugs or commercial reasons
- Cookie extraction: Preliminary for impersonation

Protecting links and framesets

- Preventing deep links: No bypassing of home page ads!
  - → Or being a link target from certain sources at all
- No direct links: Images, videos etc.
  - → Direct linking to the content to show it in a different context or as “own” service
  - → Learners have to discover the way, not receive a direct link
- Keeping link targets secret: No link stealing
  - → When the worth of a site is the collected links
  - → Not wanting search engines to know the link structure
    - » Or only if the targets have paid for this!
- No automated link following: Interstitials, click-fraud, ads, …
- Prevent subframes to appear in different context/alone
  - → Surrounding by ads, exploiting it as own service, …

Technical protection measures (1)

- Transmitting encrypted data + JavaScript decryption
- Obtaining content dynamically, e.g. by Ajax
- Hooking actions: Right-click, swallowing “Ctrl” (Ctrl-P=Print)
- Inserting “problematic” data: False tags, wrong nesting, …
- Overlaying data: Cover by transparent frame/image/…
- Position by CSS: No proximity in HTML code (→scraping)
- Alternative versions: Stylesheet for printing
- Alternative rendering software: Modified browser or plugin
- Captchas: Determine if visitor is a human
- User login: Require registration and credentials
  - → Knowledge (pwd), possession (token) or being (biometrics)
- Protocol features: Caching, X-FRAME-OPTIONS (IE 8) etc.
- Server timing: Too rapid/regular requests

Technical protection measures (2)

- Unique IDs: For every rendering of a form (→ CSRF)
- Referrer check: Where is the user coming from?
- Content queuing: Only send content after ad was delivered
- Endless data loops: Dynamically created pages for crawlers
- Incompatible code: Applets/flash using bugs in cert. versions
  - → Note: Potentially illegal and very problematic!
- Framebuster scripts: Breaking out of “incorrect” framesets
- Page structure scrambling: Each rendering is different
- New browser window without menu/status bar
- Text inside (usually) ignored tags: For JavaScript, SEO, ..
- Watermarks: Texts, images, videos, …; visible/hidden
- Low-quality content: Publicly show only a part of what exists
- No links: URL must be know/transmitted externally

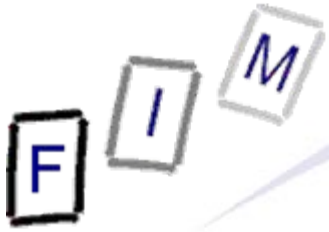
Classification of protection measures: Difficulty of circumvention

Protection measure	Circumvention-Classification
1. Transmitting encrypted data	Medium
2. Obtaining content dynamically	Medium to complex
3. Hooking actions	Simple to medium
4. Inserting “problematic” data	Simple
5. Overlaying data	Simple
6. Positioning to hide relations	Medium
7. Alternative versions	Simple to medium
8. Custom viewer software	Complex
9. Captchas	Complex
10. User login	Complex
11. Protocol features	Medium to complex
12. Server timing	Medium
13. Unique IDs	Medium
14. Referrer check	Simple to medium
15. Content queuing	Medium
16. Endless data loops	Simple
17. Incompatible code	Complex
18. Framebuster scripts	Medium
19. Page structure scrambling	Complex
20. New browser window	Trivial
21. Text inside ignored tags	Trivial
22. Watermarks	Complex
23. Low-quality content	Complex
24. No links	Trivial to Complex

- Trivial: Accidentally possible
- Simple: No special knowledge; detailed procedure on web
- Medium: Special knowledge, but readily found; effort needed
- Complex: Detailed & uncommon knowledge, custom programs, …

Conclusions

- Strong protection of EL content is possible for some aspects
  - → But it might require a large effort and introduce problems
    - » Requiring additional software, cumbersome, accessibility, …
  - → But most can only be protected very weakly
- Against amateurs protection can be moderately effective
  - → No accidental circumvention possible
    - » Legally important!
  - → No prevention, only some (slight) hindrance
- Content providers to protect their investment: Sorry!
  - → Very complex and even then not secure
- Teachers to enforce rules/didactics: Ok!
  - → Moderate effort can discourage learners to cheat, copy, shortcut, … in normal learning situations
    - » But not for exams!



Questions?

Thank you for your attention!