Security in Computing & Information Technology

Lecture 9
Web Security

Lecture Schedule

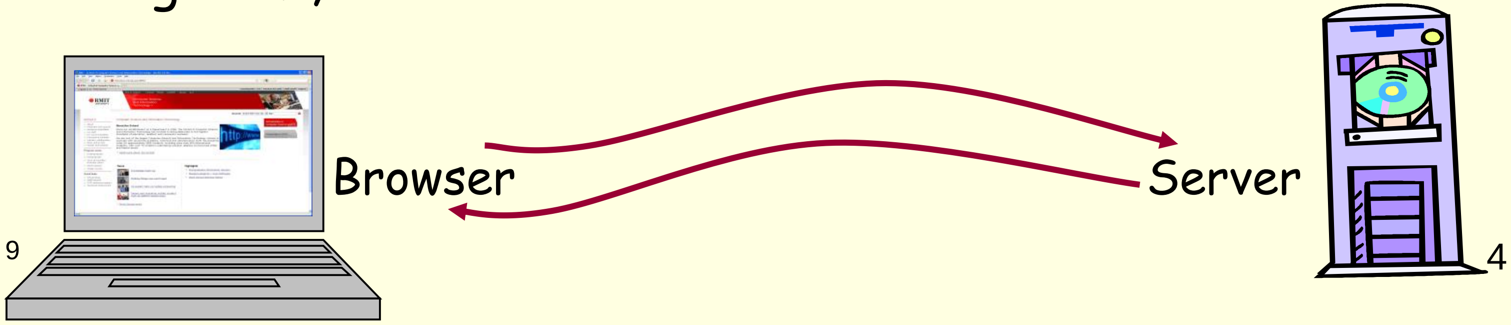
Lecture Topics

- Web basics: pages & cookies
- Web browser security policies
- Web page attack techniques

Web Page Processing

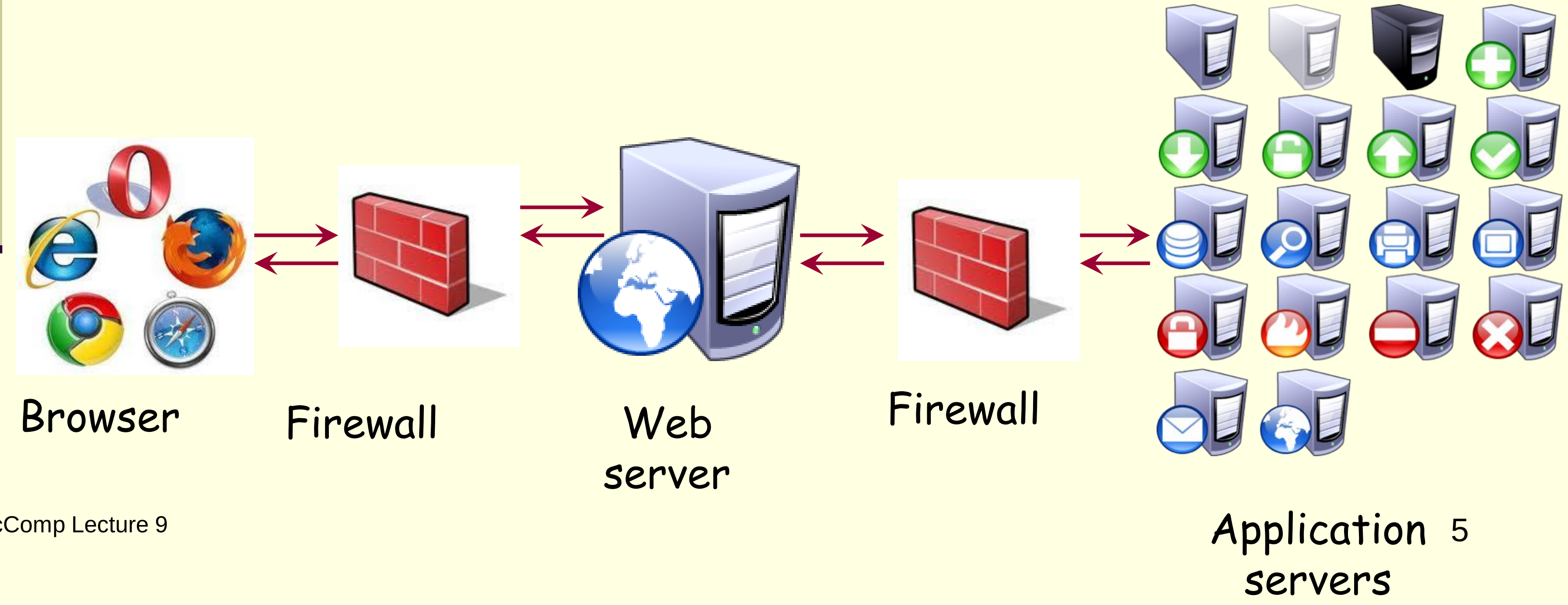
- Web page display (static)
  - Browser sends a request to a web site
  - The web server replies with a document (web page) encoded in the HyperText Markup Language (HTML)
  - The web browser interprets the document and displays it after rendering
- A web page may be processed by programs other than the browser

  E.g. PDF, …



Basic Web Architecture

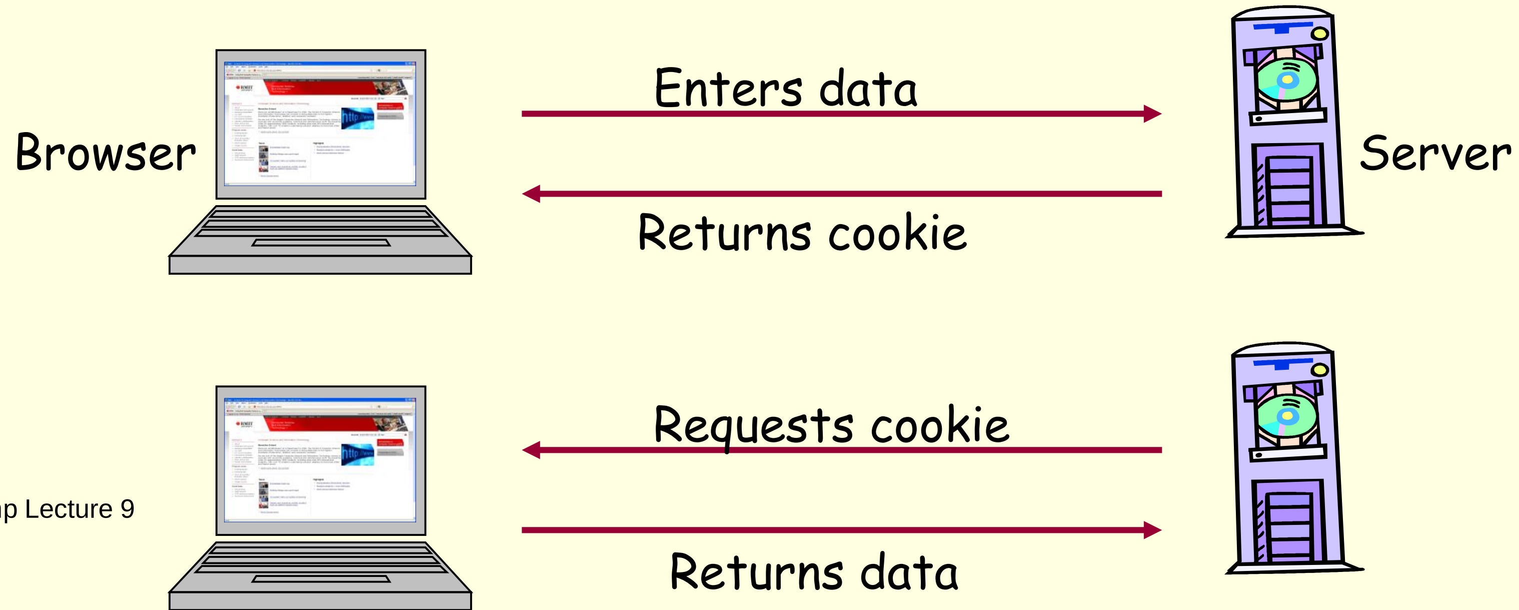
- Web servers offer different applications
- The applications can be hosted on different servers
- Security problems may effect both the Web server and the application servers
- This lecture focuses on the Web server



Cookies (1)

- Store
  - server-related information
  - at the client machine

  E.g. user ID, user status
- If stolen, can cause harm (e.g. login information)
- Can violate privacy (e.g. information collection)



Cookies (2)

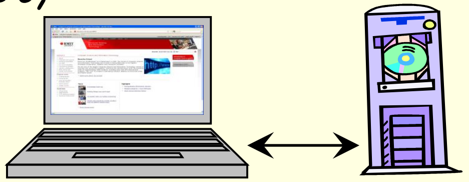
- Cookie Ownership
  - Once a cookie is saved on your computer, only the Web site that created the cookie can read it.
- Variations
  - Temporary cookies
    - Stored until you quit your browser
  - Persistent cookies
    - Remain until deleted or expire
  - Third-party cookies
    - Originates from a Web site other than the web page's URL domain (e.g. advertisers) – browsers allow you to stop them

Active Content

- Dynamic HTML (Web page generated on the fly)
  - Allows the user to interact with the web page and change the look and content of it
  - Needed for many pages to work (difficult to disable)
  - Can contain malicious code

Client side
JavaScript,
VBScript

Server side
ASP, JSP,
PHP



- Components
  - Client-side scripting
    - executes in the browser (JavaScript, VBScript, ActiveX, Flash or any other browser-supported technology)
    - affects the presentation of the web page
  - Server-side scripting
    - executes in the web server
    - used for content creation

Rendering Content and Events

- Basic browser execution model
  - Each window or frame
    - Loads content
    - Renders
      - Processes static and dynamic HTML
        - May involve pages, scripts, images, subframes, etc.
    - Displays the page (parts outside the visible window need to be scrolled to)
    - Responds to events
- Events can be
  - User actions: OnClick, OnMouseover, OnKeyPress, OnBlur
  - Rendering: OnLoad, OnUnload
  - Timing: setTimeout(), clearTimeout()
- Pages can contain content from many sources

```
<script  src="http://source.com/script.js"> </script>
<iframe  src="http://source.com/frame.html"> </iframe>
<img src="http://source.com/picture.jpg" height="50"
  width="100">
```

Embedded Content Issues

Embedded content can

- communicate with other sites
  - Use scripts to pass on local information to any site
- hide resulting image
  - Contain a link to an attack site (clickjacking)
- be used to spoof other sites
  - Use logos e.g. to impersonate bank web sites

Remote Scripting

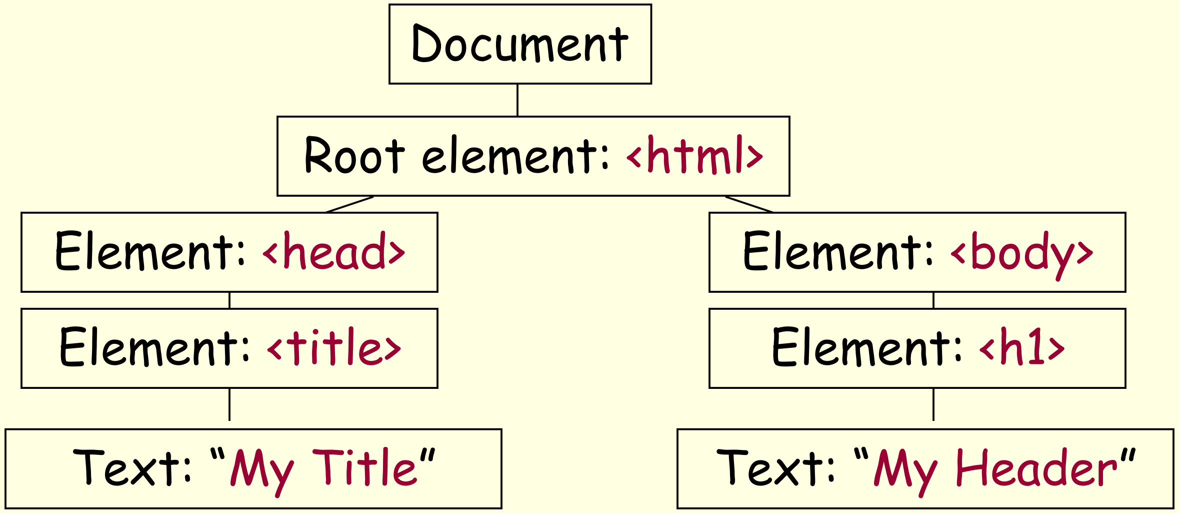
- Exchanging information between browser-side scripts (programs) and servers without reloading the page
  - Avoids reloading the page after some user action
- Local scripts invoke scripts on the remote side
  - Remote Procedure Call (RPC): XML-RPC, AjaxRPC
- Technologies: Java Applet, Active X control, Flash
- Can be used for unattended installation of software

A web site can maintain bidirectional communication with the browser until the browser closes

Document Object Model (DOM)

A standard way for accessing and manipulating HTML and XML documents

- Defines
  - Objects and properties
  - Methods to access them
- It is platform and language neutral
- Presents the document as a tree structure

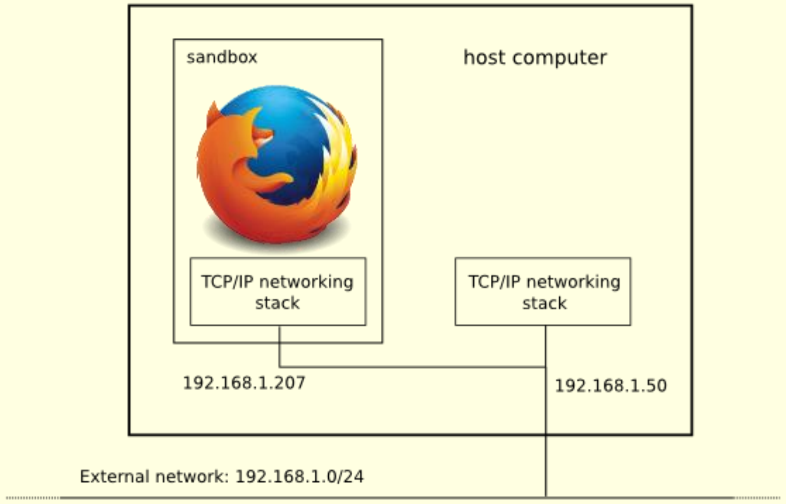


Web Security Policies

- Sandbox
  - Limits access to local resources
- Same origin policy
  - Documents from the same site are treated the same way

Sandbox

- The browser and any application running in it have limited access to most local resources



- Sites are put in different security contexts
- Different contexts have different protection

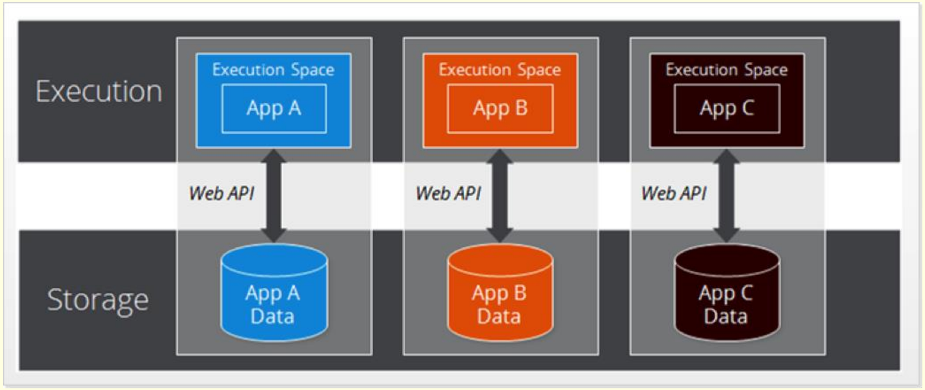


Image sources: https://l3net.wordpress.com/2015/09/21/firejail-a-security-sandbox-for-mozilla-firefox-part-3/
https://developer.mozilla.org/en-US/docs/Mozilla/B2G_OS/Security/Security_model

Same Origin Policy

- Documents and scripts originating from the same site can access each other's methods and properties (e.g. HTTP PUT)
- They cannot do the same with documents from different origins
- One-size-fits-all approach, introduced with the appearance of active content long ago
- Several attack types have been developed to circumvent this policy (including some using Adobe Flash)

Web Security Techniques

- Access control
  - Authentication
    - Basic: sends the password in plaintext
    - Digest: sends a hash of the username/password and additional information (e.g. server time)
- TLS/SSL (https)
  - Authentication certificates + encryption
- Firewalls, proxies
  - Examine, interpret messages before passing them on

Web Browser Security

- Basic action: client browser sends a request, web server returns the requested page
- Finding a web page
  - User types in / cuts & pastes the URL
  - User clicks on a URL/hypertext link
  - Problems
    - Is the URL genuine? (or point to a malicious site)
    - Does it point to the intended web page? (or to something else)
    - …
  - Web form processing
    - Plain text input: data is visible in the channel and in web logs
- Browser displaying a web page
  - Visited page may contain malware ("drive-by download")

Web Browser Security Settings

- Browsers allow users to set preferences
  - Security preferences
    - Receive warning messages

      E.g. if a site wants to install programs (add-ons)
    - Block operations

      Known attack sites, known web forgeries
    - Some settings present a vulnerability

      E.g. remember passwords
  - Privacy preferences
    - Accept cookies (and from whom)
    - Browsing history

Web Threats

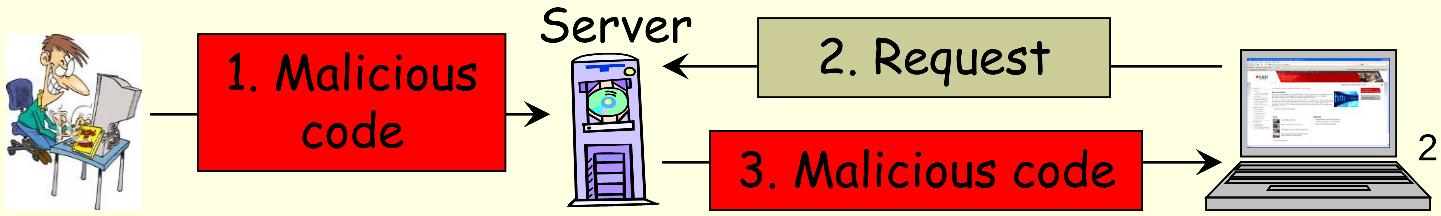
- Attack sites
  - Attacker has full control over the site
  - Site may have SSL certificate
  - Drive-by download
    - Attack is performed when site is visited
    - No user interaction is required
- Malware escaping browser protection mechanisms
  - Attacks may use legal sites with exploitable problems to inflict damage
  - Browsers (like any software) contain bugs
  - Specially crafted attacks can circumvent protection

Web Page Attack Techniques

- Rated as widespread
  - Cross-Site Scripting (XSS sometimes wrongly abbreviated as CSS) – more than 20 years old
    - (Google search: 9,511,000 results - Jan 2017)
    - Exploits user's trust in the web site
  - Cross-Site Request Forgery (CSRF)
    - (Google search: 2,030,000 results - Jan 2017)
    - Exploits the web site's trust in the user's browser
- Newer attacks
  - Clickjacking (also called UI redressing) – fairly recent
    - (Google search: 391,000 results)
- Easy attack
  - Spoofing
    - Imitating the look and feel of another web page
    - Name spoofing: e.g. g00gle.com

Cross-Site Scripting (XSS)

- Malicious browser-side scripting embedded in web pages (very frequent: ~2/3 of the whole vulnerability population)
- Attacker uses the web site to send malicious code to a different web-page user
- Vulnerable site
  - Accepts input from the user, and incorporates it in the response sent back to the user - without proper checking (e.g. search pages)
  - Sites affected in the recent past include MySpace (Samy Worm - 2005), Google, Yahoo, Facebook, Paypal (2010)



XSS Types

- Stored attack
  - The malicious code is permanently stored on the web server (e.g. in a database, message forum, visitor log)
  - Victim retrieves the malicious code when accessing the infected information
- Reflected attack
  - The user is tricked into clicking on a malicious link or submit a specially crafted form to the vulnerable web server that will send it back to the user's browser (e.g. search engine shows the query)
- Distributed object model (DOM) based
  - The malicious payload is sent as part of the URL
  - Web page refers to the supplied URL (e.g. uses data from `document.location` or `document.URL` or `document.referrer` in JavaScript))

XSS – Execution

- Script executes with the user's privileges
- Can lead to
  - Cookie theft

    Taking over the user's cookie
    - session hijack/compromise
    - complete account compromise
  - Disclosure of end user files
  - Installation of Trojan horses
  - Redirection to other sites
  - Modification of presentation or content (e.g. stock price)
- Common occurrences
  - On-line forums, message boards
  - Links attached to email
  - Search engines
  - Error messages
  - Worms

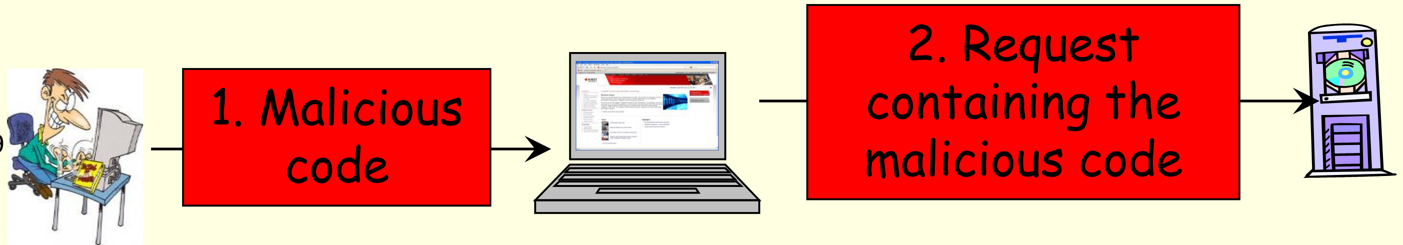
XSS Prevention and Mitigation

- Responsibility is shared between site maintainer and user
  - Web site maintainer
    - Validate (check and sanitise) input
    - Ensure characters are treated as data, not relevant to the interpreter’s parser
    - Protect your cookies from client-side scripts (use the HTTPOnly flag)
  - Browser / user
    - Be careful with spam and forums
    - Some browser extensions (add-ons) can help
      http://www.noxss.org/
- For details see the
  - Cheat sheet at http://www.owasp.org/index.php/XSS_%28Cross_Site_Scripting%29_Prevention_Cheat_Sheet
  - browser security handbook at http://code.google.com/p/browsersec/wiki/Main

‘(‘ character shown as its Unicode

Cross-Site Request Forgery (CSRF)

- Aka 'sea surf'
- Steps
  - The victim is tricked into accessing a malicious web page that contains the attack script
  - The attack script executes with the victim's identity and privileges to perform an undesired function
  - If the victim has been authenticated to a server (has a valid, non-expired cookie), the attack script can access the victim's account on the server (e.g. bank account)
- Consequences
  - Almost no limit on impact of CSRF
  - One of the most dangerous vulnerabilities (software bugs) ever



CSRF

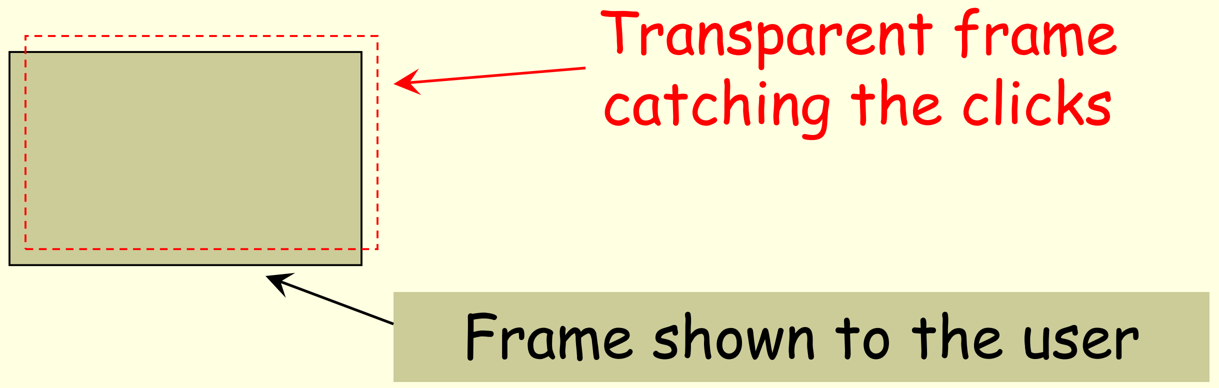
- Most frequent variants
  - Cross posting

    An HTTP POST request is sent to the web page (Data writing)
  - Cross authentication

    The attacker can perform actions at a site in the victim's name (by using the victim's cookies)
- Mitigation
  - User
    - Logging out of sites (no valid cookies)
    - Don't click on links in spam
  - Web server
    - Several methods to validate requests (authentication in each HTTP request, check HTTP Referrer, …)

Clickjacking (1)

- Multiple transparent and opaque layers of a web page result in a click on a concealed and unintended link
- HTML `<iframe>` tag defines an inline frame that includes another document
  - It can be made
    - transparent, so the background of the parent frame is visible
    - borderless and use the parent's colour to blend into the web page



Clickjacking (2)

- Examples
  - One attack tricked the victim into altering Adobe's Flash settings, e.g. animations could switch on/utilise the computer's camera and microphone
  - Registering clicks on advertisements fraudulently

    Advertisers pay by the number of clicks on an ad
- Prevention
  - Web site: not allowing pages appear in a frame (Framekiller)
  - Browser: NoScript - scripts disabled in general, explicit permission can be granted by the user for individual sites

Other Cross-Site Attacks

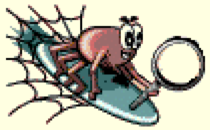
- Cross site framing
  - Site A (attacker) includes site B in the page
  - Victim believes to be operating on site B
  - Site A can use the victim's credentials on site B
- Cross site double clicking
  - Two pages pop-up together, first is hidden below the second
  - The second click will be passed on to the first pop-up, without the victim noticing it

Information Gathering on the Web Crawlers

AKA indexer, spider, robot

- Automated program that methodically browses the Web
- Creates an index of data
- Used by
  - search engines (Google, Yahoo etc) to collect data for quick response to search queries, linguists for text analysis, market research, malicious data collection

    E.g. email harvesting robots
- Downloads and caches web pages



Crawlers and Web Servers

- Policies
  - Selection: which pages to download (e.g. which hyperlinks to follow)
  - Politeness: how many pages of a website to download simultaneously to avoid web server overload
  - Parallelisation: improve performance
- Robots Exclusion Protocol (/robots.txt file, <robots> meta tag)

  Convention to prevent the access of web pages by honourable crawlers

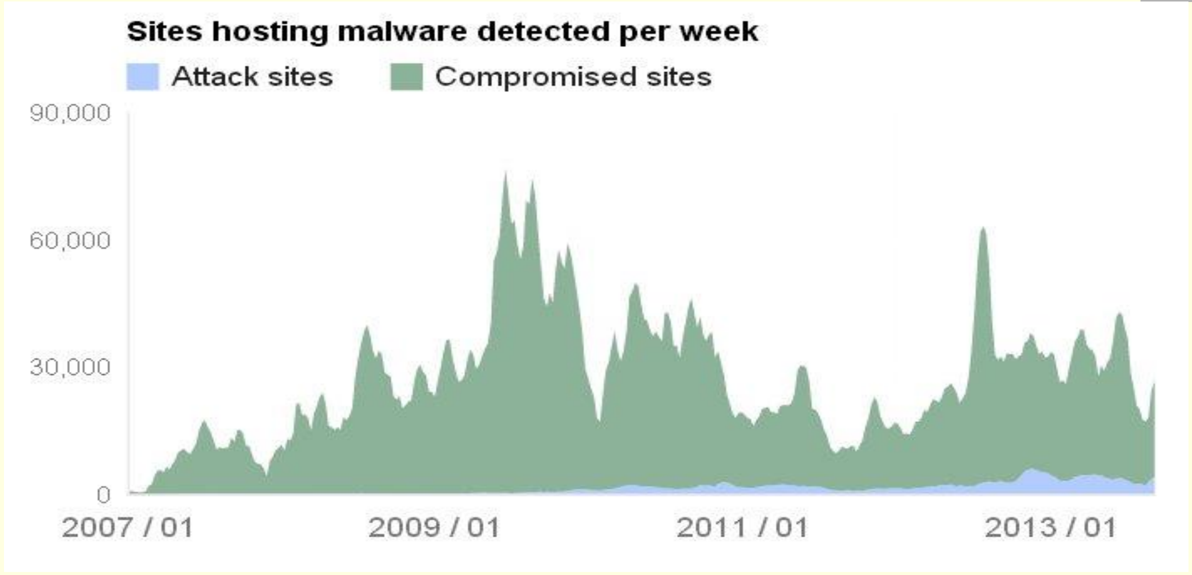
  E.g.

`User-agent: *`	*Meaning: applies to all crawlers*
`Disallow: /~joe/stuff/`	*Do not visit this page*

Spamdexing

- AKA search spam, search engine spam
- “Blackhat methods” to achieve higher search ranking
  - Cloaking: presenting different content to URL users and search engines
  - Keyword stuffing: inserting irrelevant keywords
  - JavaScript redirects: crawlers may not execute scripts
  - Doorway pages: large sets of pages, each optimised for a different keyword, all pages redirect to one page
  - Link farming: group of web pages having links to other web pages in the group

Malware Hosting and Webmaster Response Time



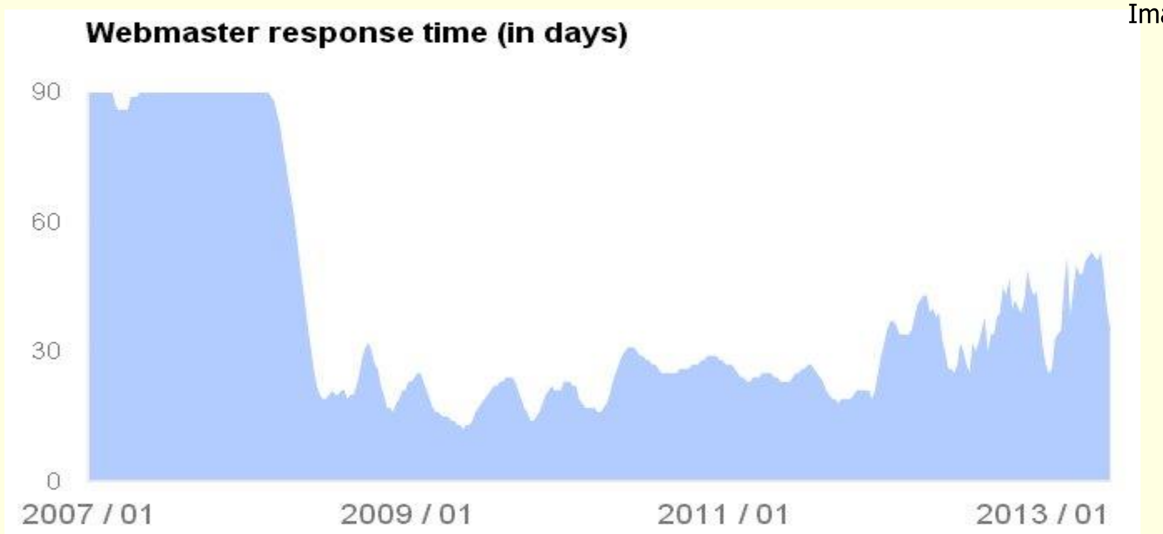
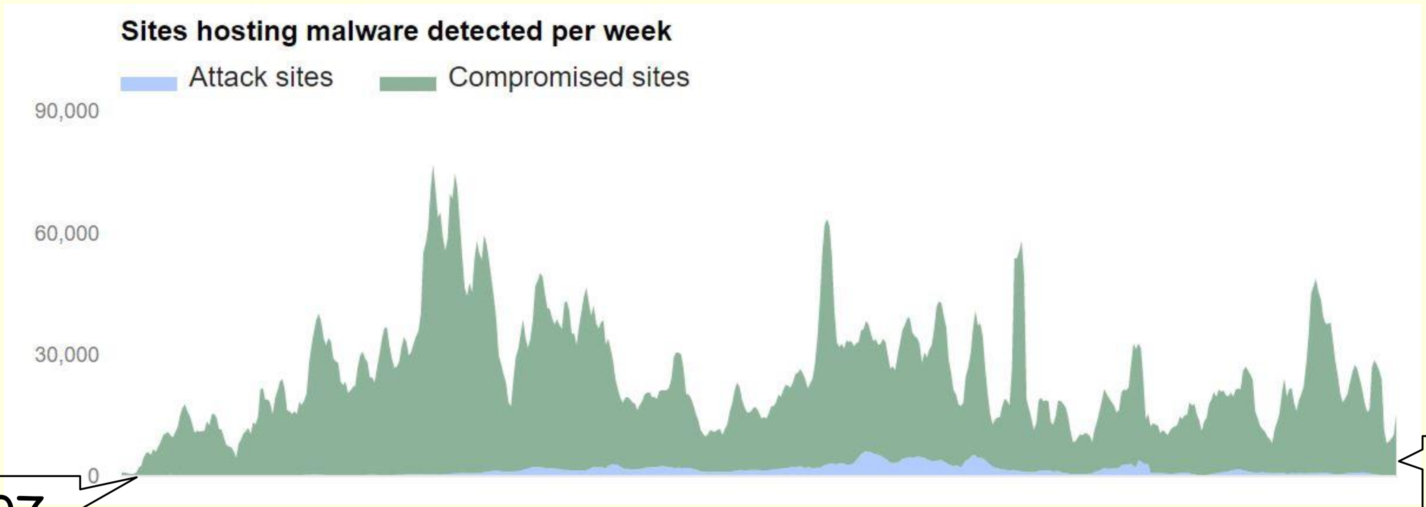


Image source
http://www.google.com/transp

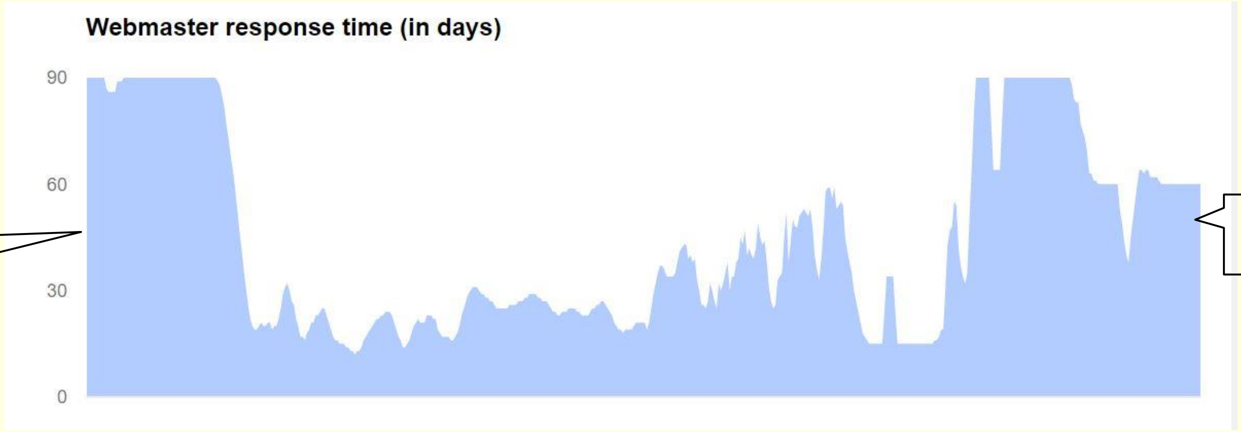
arencyreport/safebrowsing/

Malware Hosting and Webmaster Response Time



Jan 2007

Jan 2017



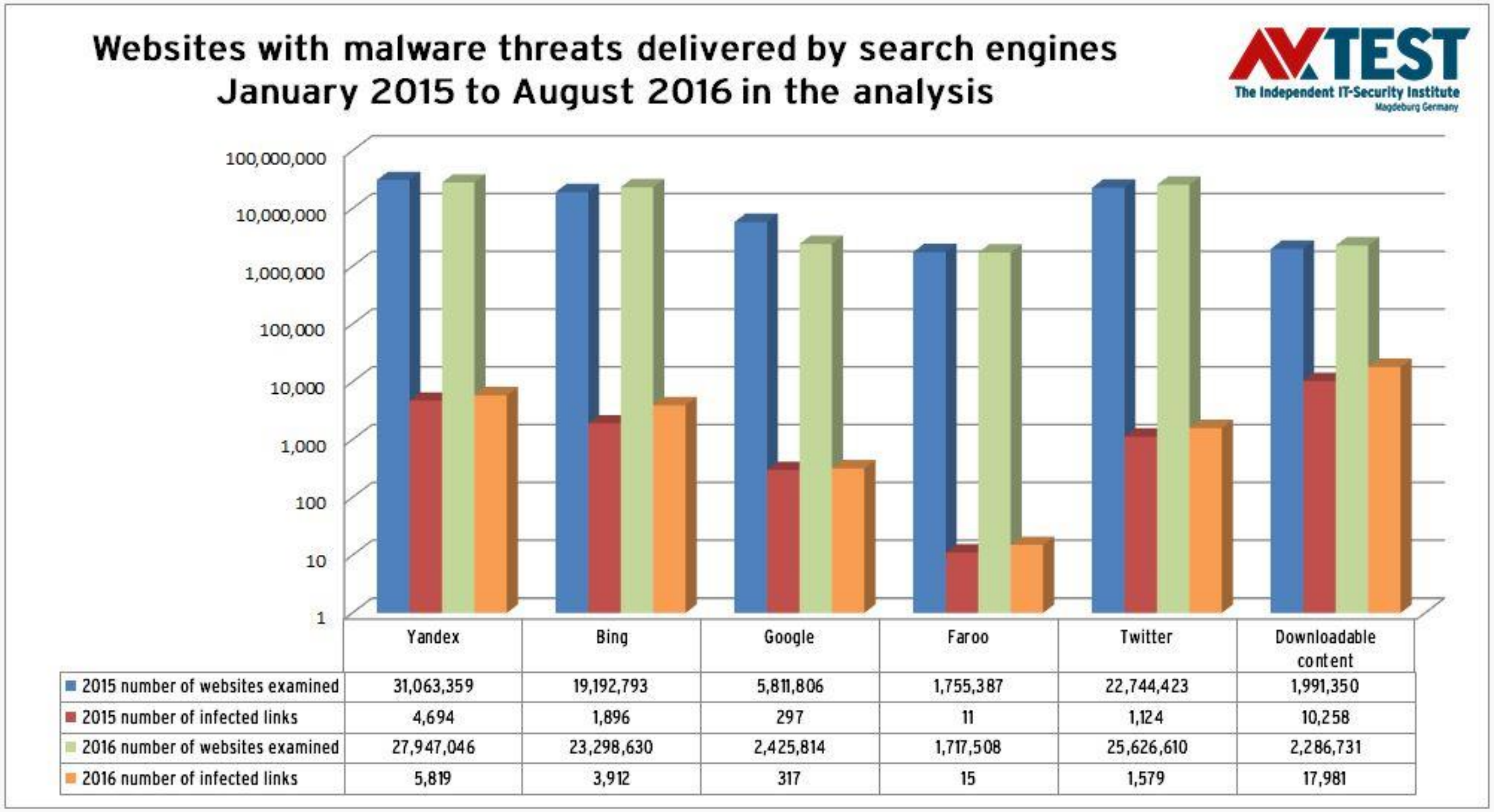
Oct 2006

Oct 2016

Image source https://www.google.com/transparencyreport/safebrowsing/

Web Search Threats

- Legitimate search can return malicious URLs



	Yandex	Bing	Google	Faroo	Twitter	Downloadable content
🟦 2015 number of websites examined	31,063,359	19,192,793	5,811,806	1,755,387	22,744,423	1,991,350
🟥 2015 number of infected links	4,694	1,896	297	11	1,124	10,258
🟩 2016 number of websites examined	27,947,046	23,298,630	2,425,814	1,717,508	25,626,610	2,286,731
🟧 2016 number of infected links	5,819	3,912	317	15	1,579	17,981

Image source https://www.av-test.org/en/news/news-single-view/analysis-of-160-million-websites-are-google-and-other-search-engines-platforms-for-distributing-mal/

Poisoned Search Results (1)

Search engine optimisation (SEO) poisoning

1. A fake site is set up to serve
   - legitimate content to crawlers and
   - malicious content to users.
2. The hacker creates a link farm to the fake site to be picked up by a crawler.
3. A search spider crawls the link farm.
4. The fake site appears in the search results. The fake site is built around themes of the day to increase the number of searching users.
5. A user clicks on the search result link leading to the hacked site, and is redirected to the malicious page.

Poisoned Search Results (2)

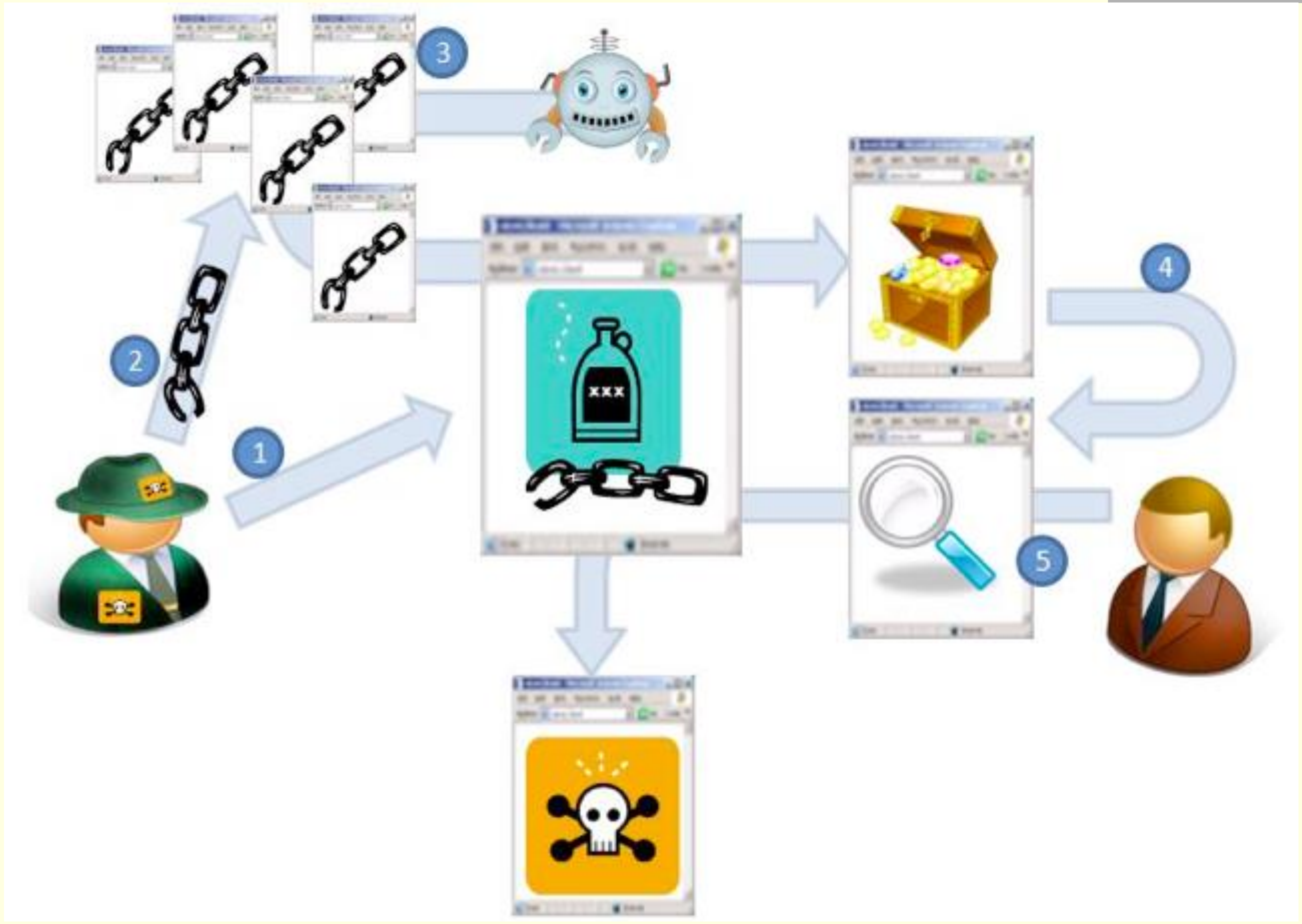


Image source http://www.symantec.com/connect/blogs/iframes-please-make-way-seo-poisoning

Summary

- Active Web-page content can be a vehicle to deliver malware
- Web browser security policies need security-conscious users
- Web-based attacks have a very wide range, including malicious scripts and search results