Imperial College
London

Network and Web Security

Same origin policy

Dr Sergio Maffeis
Department of Computing
Course web page: https://331.cybersec.fun

The Same Origin Policy

- The Same Origin Policy (SOP) is the key security policy in the browser
- Recall: URLs denote origins
  - scheme+host+port

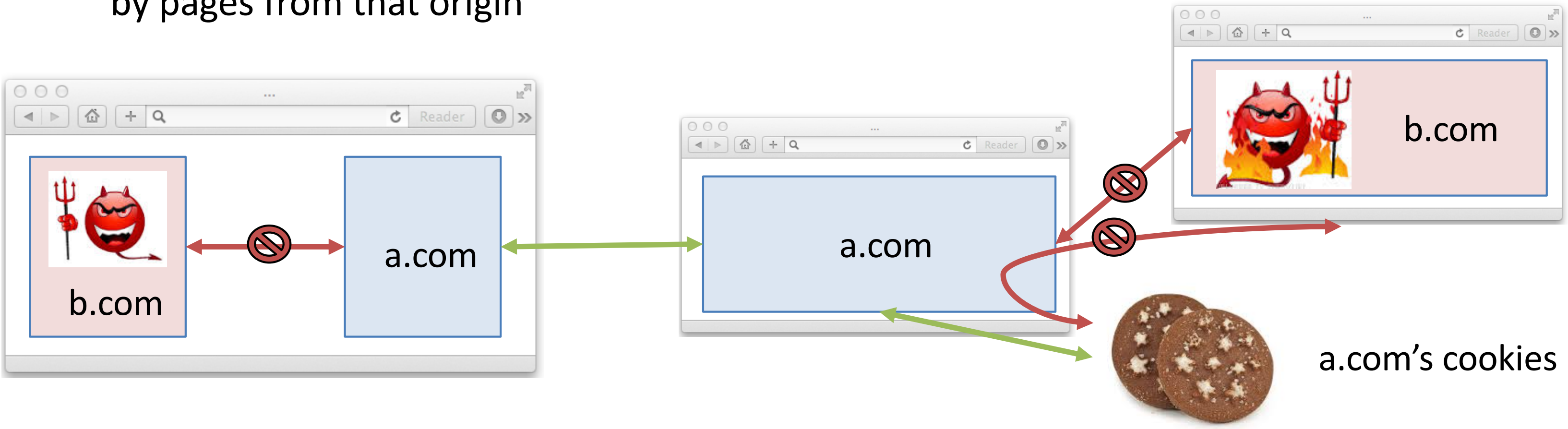
    https://www.example.com:443/path?query
- Most resources are associated to the origin where they were loaded from
- Scripts are associated to the origin of the browsing context that loaded them
- Rules for cookies are slightly different
  - See *Browser storage* module

1. page: http://9gag.tv
2. CSS: http://9gag.tv
3. script: http://9gag.tv
4. iframe: https://www.facebook.com
5. image: https://fbcdn-profile-a.akamaihd.net

```
1 <HTML>
  <head>
  …
2 <link rel="stylesheet"
   href="//cdn-jarvis-ftw.9gaging.com/…/….css”>
  <script type="text/javascript" …
3  src="//apis.google.com/js/platform.js”></script>
  </head>
  <body>
  …
  <iframe id="f15caa161c” …
4  title="Facebook Social Plugin” …
   src="https://www.facebook.com/plugins/
        comments.php?api_key=…&amp;width=617”>
4 <form … method="post"
   action="/ajax/connect/feedback.php" … >
     <input type="hidden" name="lsd"
      value="AVrPxCTM" autocomplete="off”>
     <textarea title="Add a comment..." … >
     </textarea>
     <input value="Comment" type="submit” … >
     …
  </form>
5 <img class="img" width="50px" height="50px"
   src="https://fbcdn-profile-a.akamaihd.net/….jpg”>
  …
  </iframe>
  …
  </body>
  </HTML>
```

The Same Origin Policy

- Main goals
  - Pages in different browsing contexts can interact with each other if and only if they have the same origin
  - Persistent resources (cookies, storage, …) are associated to origins, and can be accessed only by pages from that origin

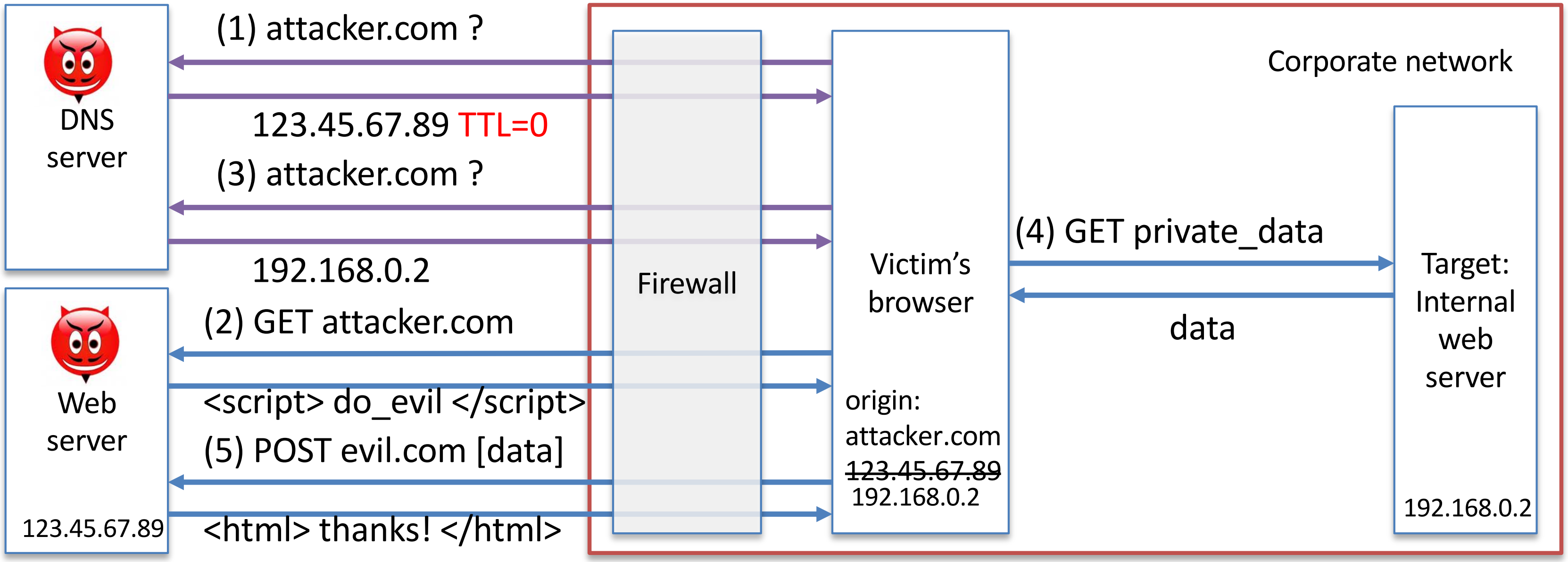


- In practice
  - Although the key guarantees are reliable, there are lots of exceptions and corner cases
    - See *Browser Security Handbook*
  - We will later see how to make the SOP more fine grained
    - More restrictive in some cases (CSP)
    - Less restrictive in others (CORS)

Attack: DNS rebinding

Imperial College London

- The SOP restricts a page to interact with other pages from the same origin (1,2)
- DNS rebinding gets around the SOP by associating the attacker DNS name to the target's IP address (3)
- Main use is to circumvent firewalls and access corporate networks through victim's browser (4,5)



- Countermeasures
  - DNS pinning: bindings cannot be changed too quickly
  - Prevent external DNS queries to resolve to internal addresses (192.168.0.2, 10.0.0.5, …)

SOP revisited

- The SOP prevents pages from different origins from tampering with each other, and confines resources based on origins
- Some accesses are allowed by default
  - Write only: `window.location.href`
  - Read only: `window.{frames,parent,top,…}`
  - Call methods: `window.{blur,close,focus,postMessage}`
  - Example: frame-busting code to prevent click-jacking

    ```
    if (top!=self) {top.location.href = location.href}
    else {… keep loading page …}
    ```
- The SOP does not prevent outbound communication
  - Ok to do cross-domain post requests
  - Page from `http://source.com sends a message to` *`target`* `server`

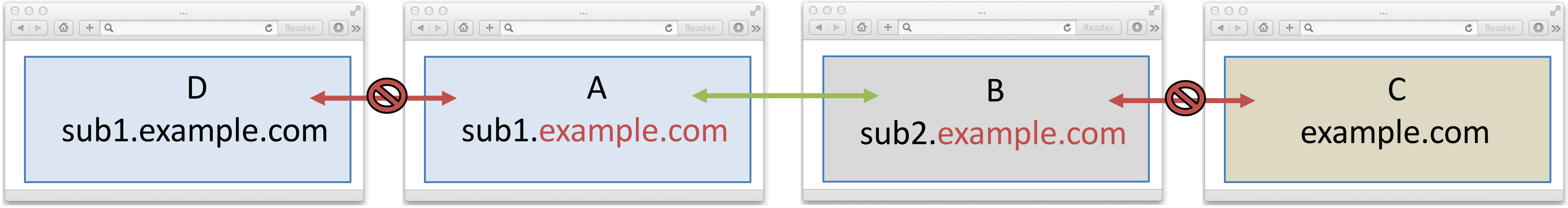
    ```
    <img src="http://target.com/img.png?msg-from-source-to-target">
    ```
- The SOP does not even prevent bidirectional communication
  - Page from `http://source.com` adds to its own DOM the HTML tag

    ```
    <script src="http://target.com/script.js?out_msg">
    ```
  - Server at `http://target.com` replies with `script.js` containing

    ```
    var in_msg = … data …
    ```
  - Server must be willing to communicate this way
  - Client must trust server, who could inject arbitrary code

Domain relaxation

Imperial College London



- Domain relaxation
  - A page can change its origin by “relaxing” the host part of its origin to become a suffix of its domain
    - The Public Suffix List is used to keep track of valid suffixes
  - Page A from `sub1.example.com` can set `document.domain=example.com`
  - Page B from `sub2.example.com` can set `document.domain=example.com`
  - The SOP now lets A and B communicate
  - Page A can no longer talk to page D from `sub1.example.com`
  - Page C from `example.com` remains isolated
- The origin is really: **scheme, host, port, relaxed_domain_flag**