Spring Security

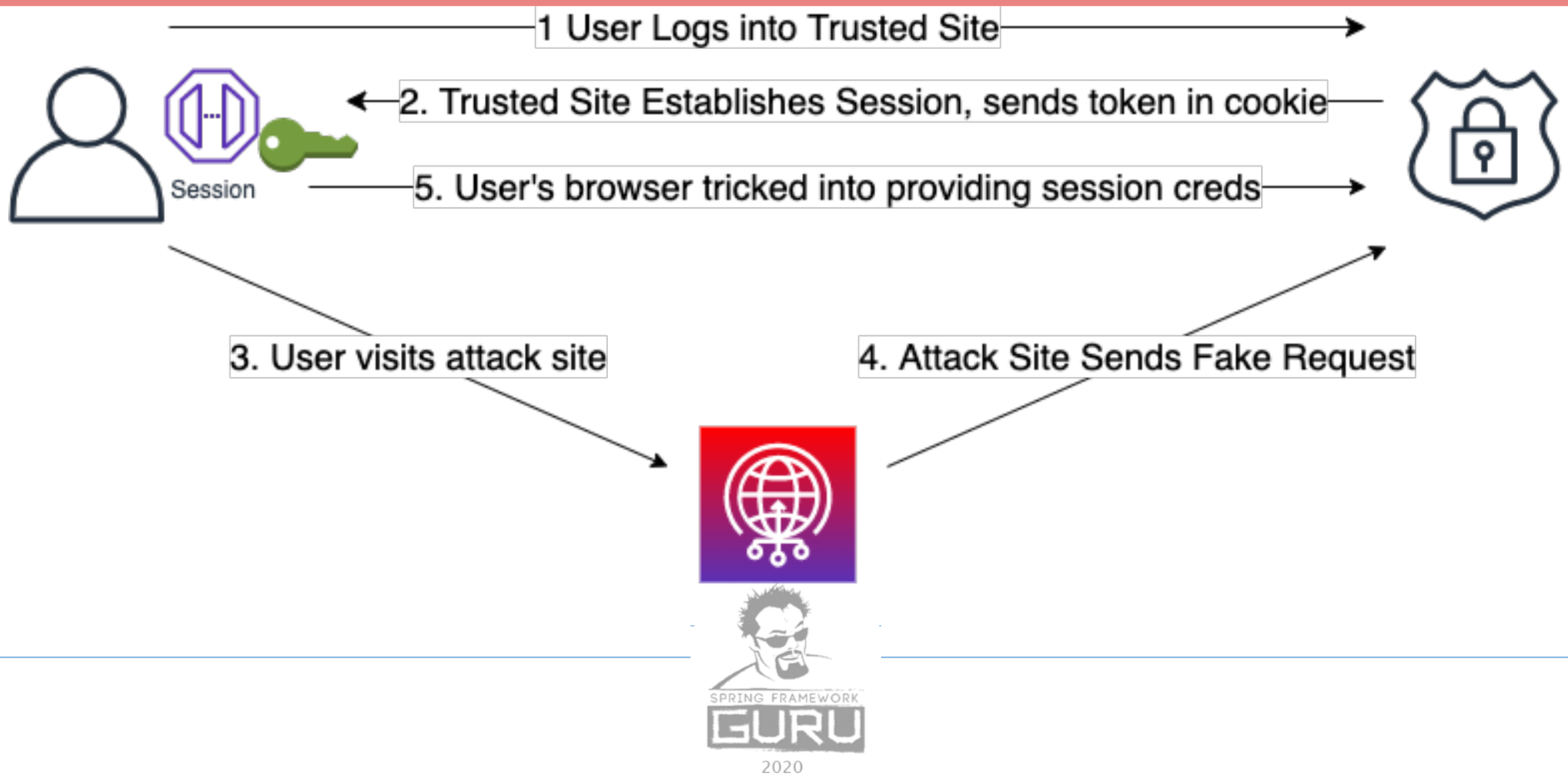
Beginner to Guru

Cross-Site Request Forgery

Cross-Site Request Forgery - CSRF

- A Cross-Site Request Forgery (CSRF) attack is when a site tricks the user's web browser to send a request to a site where the user is authenticated
- CSRF attacks work because the browser is tricked into sending session cookies to the trusted site
- Thus, the trusted site cannot distinguish the request is not from the authenticated user

Cross-Site Request Forgery - CSRF



CSRF - Synchronizer Token Pattern

- The Synchronizer Token Pattern requires in addition to the session cookie, a secure random CSRF token must be in request
- CRSF Token must be part of HTTP Request not automatically sent by browser
  - Do not store CRSF token in cookies
  - Use:
    - HTTP Headers
    - Hidden Form Fields

SameSite Cookie Attribute

- The SameSite cookie attribute can be set to tell browser to not send cookie when request is coming from other sites
- SameSite cookie attribute is supported on all modern browsers, older browsers might not support
- Supports - None, Lax (~subdomain), Strict
- Modern browsers transitioning from None to Lax if not explicitly set.
- Should not solely rely on SameSite attribute for CSRF prevention

When to Use CSRF Protection?

- Spring Security Team recommends:
  - Use CRSF when requests are processed by a browser by normal users.
    - ie - HTML, Single page apps (Angular, React, etc)
  - If only used by non-browser clients, disable CSRF
    - ie - programatic clients, like Spring RestTemplate or WebClient

SPRING FRAMEWORK

GURU