Burp Session Handling Rules

OWASP Stammtisch Ruhrpott, 26.07.2016

Thomas Patzke

Agenda

- Motivation/Overview
- Building and Debugging Session Handling Rules
- Use Case: Keep Login Session
- Use Case: CSRF Tokens
- Use Case: Workflows
- Use Case: Deletion of Generated Data
- Complex Rules with Extender API

Motivation

- Modern web applications can be quite complex
- Requests require a login session
- A request requires data from previous responses (e.g. CSRF tokens)
- Data received in a particular request can be processed some requests later (e.g. checkout process in webshops)
- Requirement for unique names/IDs in creation requests
- Fuzzing generates a lot of displayed data and slows down the application

Burp Session Handling Concepts

- Macros
  - Sequence of requests
  - Definition of source and destination of parameter values
- Cookie Jar
- Session Handling Rules
  - Scope: tools, URLs and parameters (optional)
  - Actions:
    - Set cookies or parameters
    - Run macros before and after request
    - Check session validity and restore login sessions
    - Prompt user to perform actions
    - Run custom code
- Session handling tracer: debugging tool for session handling rules

Use Case: CSRF Tokens

- Challenge: CSRF token must match and be submitted with each POST request
- Solution:
  - Capture request with form containing CSRF token
  - Create “Run a Macro” session handling rule
    - Scope to CSRF token parameter
    - Only update CSRF token in current request

Use Case: Keep Login Session

- Challenge: web application terminates login session with probability of 5%
- Solution:
  - Capture login request in macro
  - Create “Check Session is Valid” rule
    - Issue current request
    - Check response body for user name (session valid)
    - OR: Check redirection target for login URL (session invalid)
    - Run captured macro on session invalidity

Use Case: Workflows

- Challenge: Multiple request until final processing appears
- Solution:
  - Execute all requests until fuzzed one with macro
  - Finalize workflow with “Run a Post-Request Macro” that returns current request
  - Define a tight scope (URL, parameters)

Use Case: Deletion of Inserted Data

- Challenge: Only 3 entries allowed, “add” should be fuzzed.
- Solution: delete just added record afterwards
  - Post-request macro that extracts object id and performs deletion workflow

Session Handling with Extender API

- ISessionHandlingAction
  - Execution after session handling actions
  - Can change current request
  - Gets previously executed macro request and response
- IHttpListener
  - Useful for extraction of data from arbitrary requests

Examples for Usage of Extender API

- Randomizer
  - Randomizes parts of requests, e.g. to create unique names
  - https://github.com/thomaspatzke/Burp-Randomizer
- CSRFToken.py
  - Watches responses for CSRF tokens that are used later
  - https://gist.github.com/thomaspatzke/386098fb7348606b295a