CENTER FOR CYBER SECURITY TRAINING

Burp Suite Pro

Abstract

This training isn't about Web hacking. Instead, this training is for Web hackers who want to master their toolbox.

Burp Suite Pro is the leading tool for auditing Web applications at large, but also a complex beast where new features get added every few weeks. Mastering Burp Suite Pro, including its newest features, allows testers to get the most out of the tool, optimizing time spent auditing and testing. Work will be faster (hotkeys!) and much more efficient (more tools, more possibilities!). Attendees will also learn to measure and assess the quality of their attacks, a crucial skill in real-life engagements that can make the difference between a false-negative and a critical finding.

Learning Objectives

Menial tasks (like sharing requests among the different tools, applying common encodings or navigating the GUI) should be as fast and transparent as possible, in order to free time and brain power for harder subjects. Recurrent tasks (like brute-forcing a CSRF-protected form, frobbing an opaque blob of data, logging-in automatically or doing 1-byte fuzzing of a specific parameter) should be executed without having to think too much about it, thanks to prior rehearsals. Advanced tasks (like managing a complex state, dealing with a custom format or testing authorizations) should be doable exclusively in Burp Suite Pro, possibly with the help of session handling rules or specific extensions. These tasks require testers to live-assess themselves, in order to detect as early as possible any error and to allow for correction and self-improvement.

Every trainee goes through the main set, composed of nearly 100 challenges. Plenty of additional ones are available, depending on your speed, taste, skills and professional needs. No way to get bored! Among the available challenges: complex brute-force, data extraction, support of custom formats, automatic management of anti-CSRF tokens, WebSockets, weak

cryptography, webhooks, NoSQL injections, authorizations bugs, aggressive disconnection, JWTauthenticated APIs, arbitrary Java deserialization, blind stored XSS, instrumented Java applications, strict workflows, etc. The challenges are hosted in a Docker infrastructure (around 20 containers) which is made available to all trainees right after the training session. It's super easy to use: install Docker Compose, run a few commands, enjoy the challenges!

Target Audience

The training is aimed at Web application penetration testers and bug hunters and will provide them with significant automation capabilities. We aim at a fast and comfortable testing workflow with as-short-as-possible feedback loops.

Course Outline

1. After an introduction to the training platform and its challenges, this day is spent on well-defined tasks where the goal is to find flags, like in CTF contests. We practice basic automation using tools like Proxy, Repeater and Intruder. The goal is to improve the speed of our interactions with the tool, while monitoring and self-assessing our attacks.

    • Introduction: rules and advice, connecting to the network, description of the training platform and its challenges

    • Getting started: navigating the GUI, loading custom options, using hotkeys, sorting and filtering data

    • Match & Replace: well-known examples, live traffic modifications

    • Repeater: keyboard-only usage, replaying WebSockets traffic, dealing with streamed data

    • Intruder: coverage of all attack types and most payload types, automatic processing of results with "Grep – Match" and "Grep - Extract", data extraction, managing CSRF-tokens without session handling rules, atypical injection points, frobbing and fuzzing

2. Challenges get more realistic: solving them requires a good understanding of the underlying application and the usage of multiple Burp Suite tools, possibly including extensions. Additionally, we keep working on the efficiency of the testing workflow (using shortcuts or extensions) and on self-monitoring (now with Logger++). The latter skill will prove itself invaluable when working on session handling rules.

• Traffic interception: HTTP exchanges and WebSocket messages are intercepted and modified on the fly, in order to bypass client-side protections or to subvert the logic of (emulated) mobile apps. That's the only section where "Intercept is On" isn't a problem

• Macros and session handling rules for Web applications: terminology, basic setups, common use-cases (like managing CSRF tokens or logging-in automatically), applying session handling rules to third-party tools like sqlmap. Note that dealing with Web services (either SOAP or REST) is quite different and is covered in the separate section, on the third day

• Extensions: review and testing of the most useful and/or popular extensions (Logger++, Hackvertor, JSON Beautifier, Paramalyzer, Turbo Intruder, etc.)

3. Next, we dig deeper in advanced subjects. That covers authorization testing, custom active scanning, Web Services and much more! Built-in features are pushed to their limits, and extra ones provided by extensions are commonly used.

   • Authorization testing: from quick tests w/o specific configuration to deep tests requiring business-specific knowledge (extensions "Authz", "AutoRepeater", "SessionAuth" and "AuthMatrix" are covered)

   • REST and SOAP WebServices: why is a specific toolbox needed, generating requests from definition files (WSDL, OpenAPI, etc.), using session handling rules to manage authentication in cookie-less environments

   • Two-way communication with the target: deploying and using a private Collaborator instance, patching the target byte-code with Infiltrator in order to receive additional details (filename, line number, etc.), running an Infiltrator-only active scan

   • Scans and live tasks: differences between v1 and v2 (terminology, GUI, usage), using the scanner like in v1, description and testing of the much-improved crawler, configuring and running specialized scans, observing the oriented-graphs generated during crawling, using these graphs with "Crawl and Audit" (in order to audit CSRF-protected forms without macros)

   • Headless usage: driving Burp Suite Pro via a REST API (provided by 3rd-party extensions)

   • Vuln-specific tooling: Blind XSS, from builtin features to 3rd-party tools like Sleepy Puppy

REQUIREMENTS

Student Machine/ Laptop Requirements

- Computer (with appropriate WiFi connectivity)
- 64-bit OS supported by Burp Suite Pro (Linux, Windows or Mac)
- Administrative privileges (in order to configure network settings)
- Recent version of the 64-bit Oracle JVM (possibly installed using the Burp bundles)
- Burp Suite Pro license (I can provide temporary ones)
- Modern browser (no IE6, no Epiphany)

Students Knowledge Pre-Requisites:

Students must have the following:

- Basic knowledge of Burp Suite (UI navigation, traffic interception and replay)