BURP SUITE REPORT

Ameer Hamza 63582

Nouman Naeem 65589

Syed Mujtaba Zaidi 62081

Muhammad Sagheer 65647

Muhammad Maaviyah Rehman 64760

TABLE OF CONTENTS

INTRODUCTION

1.WHAT IS BURP SUITE?

Definition
Burp Suite is an integrated platform of tools for testing web-application security. It sits between your browser and the target web app as an HTTP(S) proxy, letting you inspect, modify and replay traffic, plus run automated scanners and manual attack workflows. Think of it as a security testing IDE for web apps.

Primary purpose(s)

- **Intercept & inspect traffic**: view every request/response, headers, cookies, bodies.
- **Manipulate traffic**: edit requests/responses on the fly to test input validation, session handling, auth, etc.
- **Discover vulnerabilities**: via automated crawling/scanning and human-driven testing.
- **Exploit & verify**: tools for fuzzing, brute force, session analysis and manual verification of issues.
- **Automate and integrate**: support for scripted workflows, extensions, and CI/CD integrations (enterprise features).

2.HISTORY & DEVELOPER (PORTSWIGGER) EVOLUTION AND MAJOR ADDITIONS

Developer:
Burp Suite is developed by **PortSwigger Ltd.**, a UK company founded by security researcher **Dafydd Stuttard** (co-author of The Web Application Hacker's Handbook). PortSwigger produces Burp Suite and extensive learning resources (Web Security Academy).

Evolution:
Burp began as a set of developer/security utilities for intercepting web traffic and gradually matured into a full commercial product with modular tools, automation and enterprise features. Over many releases the product expanded from a manual proxy-focused toolkit to a hybrid platform that supports full automated scanning, collaboration, extensibility, and CI/CD integration.

3.IMPORTANCE IN WEB APPLICATION PENETRATION TESTING

Burp Suite is widely used in the industry for several reasons here are the core ones.

1. Centralized workflow (manual + automated in one UI)
You can discover issues automatically with the scanner, then immediately use Repeater/Intruder/Proxy to validate and exploit them. That tight feedback loop speeds up testing and reduces tool-switching overhead.

2. Deep traffic inspection & manipulation
Burp's proxy lets you observe and edit live requests/responses (headers, cookies, bodies, JSON, multipart, WebSocket, etc.), which is fundamental to most web attacks: modifying parameters, altering cookies, replaying requests, tampering with JWTs, etc.

3. Powerful automation where it helps
The automated scanner and crawler accelerate discovery of low-hanging vulnerabilities so testers can focus time on complex business-logic and chained vulnerabilities that automation misses.

4. Extensibility
The Extender API + BApp store allows bespoke tooling and community-driven additions. This is crucial

because web apps have so many custom issues, frameworks, and protocols — extensibility keeps Burp useful across many scenarios.

5. Out-of-band (OOB) detection
The Collaborator service is unique/valuable for detecting blind vulnerabilities that don't immediately return an observable response — e.g., blind SSRF, blind command injection that triggers DNS/HTTP callbacks.

6. Token/session analysis
Tools like Sequencer and the suite's session-handling features help analyze randomness/entropy of session tokens and cookies — important for evaluating authentication/session security.

7. Industry adoption & community
Burp is a de-facto tool for many professional pentesters, so training material, tutorials, integration examples, and community help are abundant. That lowers the learning curve and makes results reproducible across teams.

BURP SUITE ARCHITECTURE AND CORE COMPONENTS.

Burp Suite is a tool in Kali Linux which is designed as an integrated platform that brings multiple security tools together in a single interface. It is a very useful tool for cyber security. Every tool in Burp Suite has its own purpose, but they are connected with each other so they can exchange information. This connection makes the workflow faster. Burp Suite also updates all the tools automatically whenever it detects changes. At the center of Burp Suite is the Intercepting Proxy, which is like the heart of the whole system. This proxy sits between the browser and the website. Whenever a user sends a request to a website, Burp Suite captures that request and lets you view or modify it before sending it forward. In the same way, it also captures the response coming back from the server. Due to this, Burp Suite gives you complete control over the communication happening between the user and the website. This same feature also helps in performing a Man-in-the-Middle (MITM) attack. On top of this proxy, Burp Suite provides useful tools like Spider, Scanner, Intruder, Repeater, Decoder, and Comparer. Each tool performs its specific task. For example, the Scanner checks vulnerabilities automatically, the Repeater allows manual testing, and the Intruder helps perform automated attacks such as brute force or fuzzing. Even though all these tools have different functions, they use the same data coming from the proxy, which keeps everything organized and connected. Burp Suite's architecture is also very flexible. Testers can choose whether they want to work manually, automate some steps, or use a mix of both. All captured requests, cookies, parameters, and sessions are stored in one place, so switching between tools becomes very easy. This is very helpful for testers because it reduces the chance of missing important information, especially when testing large websites or complex applications. In simple words, the structure of Burp Suite is designed to give penetration testers an easy and smooth workflow. By bringing all tools under one platform, it helps analyze traffic, find weaknesses, and understand how a web application works. Whether someone is performing basic request interception or advanced vulnerability scanning, Burp Suite's connected tools ensure that the testing process is efficient and effective

BURP PROXY (DETAILED EXPLANATION) & ITS ROLE IN MITM

The burp proxy is an important and commonly used feature of burp suite. It acts like a proxy between the target and the website. When the proxy is active, every request from the user passes through the proxy and then reaches the website. In the same manner, when the website generates the response, it passes through the proxy and then reaches the user. This gives the tester full and visibility control over the communication between users and the website. The proxy works by intercepting traffic. When interception is turned on, the proxy stops and displays it to tester. This allows the tester to analyze the data such as parameters, cookies, headers, and other information. After this tester can change and analyze the data according to it before sending it to forward. This feature is very powerful because it helps testers understand how a web application responds to different inputs. Burp Proxy also allows testers to capture responses coming from the server. The response can show HTML content, hidden fields, session tokens, error messages, or backend information that is useful for security analysis. By studying these responses, testers can identify weaknesses such as insecure cookies, misconfigurations, or sensitive information leakage. All these captured requests and responses are saved in the history tab, so the tester can review them later. Another major feature of Burp Proxy is its ability to create a Man-in-the-Middle (MITM) position. In a MITM setup, the proxy stands between the user and the website, allowing it to read, modify, or forward all communication. This is the same technique you performed during your practice. Since Burp acts as the middle-man, it can decrypt HTTPS traffic using its own certificate. When the user installs this certificate in the browser, Burp can examine even encrypted communication. This greatly helps in finding vulnerabilities that appear only in secure traffic. Burp Proxy also supports custom rules and filters. For example, testers can choose which types of requests they want to intercept—POST, GET, file uploads, or API calls. They can also choose to ignore some requests, like images and CSS files, to avoid unnecessary noise. This makes the testing more focused and cleaner. In simple words, the Burp Proxy is the backbone of Burp Suite. Without the proxy, the other tools would not receive traffic to analyze. It gives full control over web requests and responses, helps in performing MITM attacks, and supports both manual and automated testing. Whether someone is checking for simple issues or testing a full web application, Burp Proxy always acts as the starting point for the entire penetration-testing process

COMPARISON BETWEEN BURP SUITE EDITIONS

A comprehensive explanation of the capabilities, limitations, and intended use-cases of each edition.

1.BURP SUITE COMMUNITY EDITION (FREE)

The Community Edition provides the foundational manual testing tools of Burp Suite. It is primarily intended for learners, hobbyists, and testers who do not require automation or high-performance features.

Included Features

1.Proxy:

The core interception tool that captures, inspects, and modifies HTTP/S traffic between the browser and target application.

2.Repeater (Basic):

Allows manual modification and resubmission of individual HTTP requests. Useful for verifying vulnerabilities and analyzing application behavior.

3.Decoder:

Supports multiple encoding and decoding formats, including Base64, URL encoding, hexadecimal, HTML entities, and more.

4.Comparer:

Enables side-by-side comparison of requests or responses to identify differences that may indicate filtering, validation, or logic changes.

5.Intruder (Limited):

Available but intentionally throttled. Speed, threads, and performance are significantly restricted, making it unsuitable for large-scale fuzzing or brute-forcing.

Major Limitations

- No automated vulnerability scanning
- No Burp Collaborator client for out-of-band (OOB) detection
- Limited Intruder performance
- No advanced crawling capabilities
- No project saving, automated reporting, or macros
- No productivity enhancements found in the Professional edition

Target Audience

- Students and beginners learning web application security
- Individuals performing small-scale manual testing
- Users who do not require automated scanning or OOB detection

2.BURP SUITE PROFESSIONAL EDITION (PAID – FOR PENTESTERS)

The Professional Edition is the industry standard for web penetration testing. It includes all Community features and adds advanced automation, scalability, and productivity tools. This edition is designed for consultants, security engineers, and serious bug bounty researchers.

Additional Features Beyond Community

1.Automated Vulnerability Scanner

A comprehensive active scanner capable of detecting a wide range of vulnerabilities, including injection flaws, cross-site scripting, access control issues, server misconfigurations, and DOM-based weaknesses. Results include detailed descriptions, remediation steps, and proof-of-concept payloads.

2.Full-Speed Intruder

Removes all throttling. Supports multi-threaded, high-performance fuzzing, brute-forcing, parameter discovery, and large payload sets. Includes additional attack types, payload processing rules, and session-handling capabilities.

3.Burp Collaborator

Enables detection of out-of-band vulnerabilities, such as blind SSRF, blind XSS, out-of-band XXE, and asynchronous callbacks. This is a significant advantage over Community Edition.

4.Enhanced Crawler

Capable of deep crawling, including dynamic and JavaScript-heavy applications. Improves coverage for both manual and automated testing.

5.Project Management and Reporting

Allows project saving, long-term testing workflows, exporting of detailed reports, and tracking issues discovered during engagements.

6.Productivity Enhancements

Includes numerous improvements such as:

- Faster and more detailed proxy inspection
- Enhanced Repeater features
- Macros and session-handling rules for complex authentication
- Built-in wordlists and payload generators
- Better organization of site maps, requests, and issues

Target Audience

- Professional penetration testers
- Bug bounty hunters
- Security researchers
- Red team members
- Consultants performing regular assessments
- Individuals who require both manual and automated testing capabilities

3.BURP SUITE ENTERPRISE EDITION (PAID – FOR ORGANIZATIONS)

The Enterprise Edition is designed for automated, large-scale, and continuous security scanning across an organization's entire application estate. Unlike the Professional Edition, it is not used for manual pentesting. Instead, it enables automated scanning and integration with development pipelines.

Key Features

1.Automated, Scalable Scanning:

Runs Burp's automated vulnerability scanner across many applications simultaneously. Supports large fleets of scanning agents to handle enterprise workloads.

2.Scheduling and Orchestration:

Allows scheduled scans (daily, weekly, monthly) or on-demand scanning. Useful for routine assessments, regression testing, and continuous monitoring.

3.Centralized Dashboard:

Offers a unified view of vulnerability findings, application health, and scan history. Provides analytics and reporting at the organizational level.

4.Multi-Instance Scanning:

Distributes scans across multiple scanners or nodes to optimize performance and reduce time required for large-scale assessments.

5.Team and Role Management:

Supports role-based access control, allowing separation of duties between security teams, developers, and management.

6.CI/CD and DevOps Integration:

Integrates with Jenkins, GitHub Actions, GitLab, Azure DevOps, and other pipeline systems to trigger scans during application development. Findings can be forwarded to ticketing systems such as Jira or ServiceNow.

Target Audience

- Medium and large enterprises
- DevSecOps teams
- Organizations practicing continuous integration and delivery
- Companies managing many web applications
- Security teams requiring centralized reporting and automated risk assessment

Summary Comparison Table

Feature	Community	Professional	Enterprise
Manual Proxy Tools	Yes	Yes	No
Repeater / Decoder / Comparer	Yes	Yes	No
Intruder	Limited	Full-performance	No
Automated Scanner	No	Yes	Yes (at scale)
Collaborator (OOB Testing)	No	Yes	Yes
Advanced Crawler	No	Yes	Yes (automated)
Project Saving	No	Yes	No
Reporting	No	Yes	Yes (centralized)
Dashboard / Team Management	No	No	Yes
CI/CD Integration	No	No	Yes
Designed for Manual Pentesting	Yes	Yes	No
Designed for Large-Scale Automated Scanning	No	No	Yes

INSTALLATION, SETUP & CONFIGURATION

1. SYSTEM REQUIREMENTS

Burp Suite can operate on a wide range of systems because it is built on Java. However, optimal performance, especially for the Professional Scanner, requires sufficient memory and CPU.

Minimum Requirements

- **CPU:** Dual-core processor
- **RAM:** 4 GB (Community) / 8 GB (Professional recommended)
- **Storage:** 1 GB available
- **Java:** Bundled JRE (no external installation needed)
- **Display:** 1024×768 resolution

Recommended Requirements (for Professional scanning)

- **CPU:** Quad-core 64-bit
- **RAM:** 16 GB or higher
- **Storage:** SSD recommended for faster project loading
- **OS:** Windows 10/11, macOS, or any Linux distribution

2. INSTALLATION STEPS (WINDOWS, MACOS, LINUX)

Installation on Windows

- Download the Windows installer from PortSwigger’s official website.
- Run the .exe installer.
- Choose installation path and allow the installer to package its own JRE.
- After installation, launch Burp Suite from the Start Menu.
- Select either **Temporary Project** or **New Project on Disk** when starting.

Installation on macOS

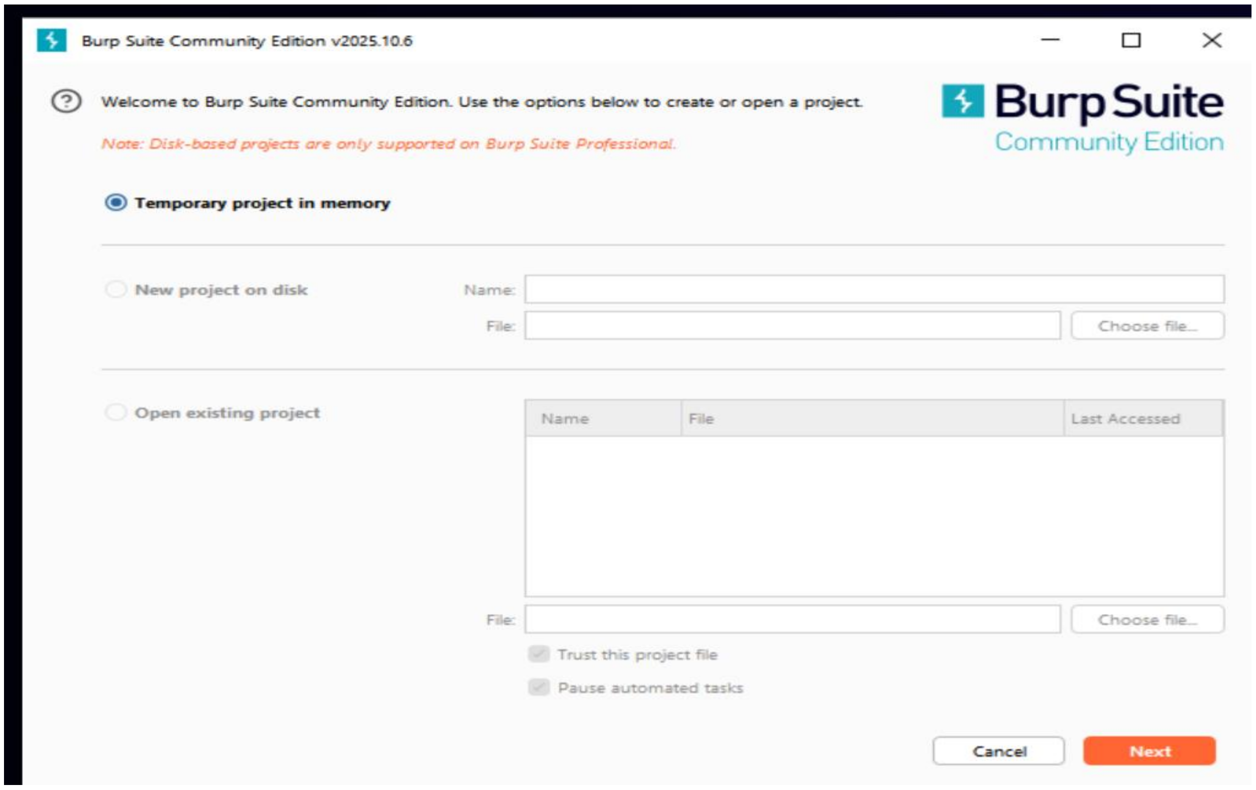
- Download the .dmg file from PortSwigger.
- Drag Burp Suite into the Applications folder.
- Launch it (macOS may require allowing it in Security & Privacy settings).
- Continue to project selection.

Installation on Linux

- Update package lists:
  ```
  sudo apt update
  ```
- Install Burp Suite:
  ```
  sudo apt install burpsuite
  ```
- Launch Burp Suite:
  ```
  sudo burpsuite &
  ```

3. INITIAL PROJECT SETUP

When Burp Suite is launched, it displays the **Project Selection** screen. Here the user chooses how Burp will store data.



Burp Suite Community Edition v2025.10.6

Welcome to Burp Suite Community Edition. Use the options below to create or open a project.

Note: Disk-based projects are only supported on Burp Suite Professional.

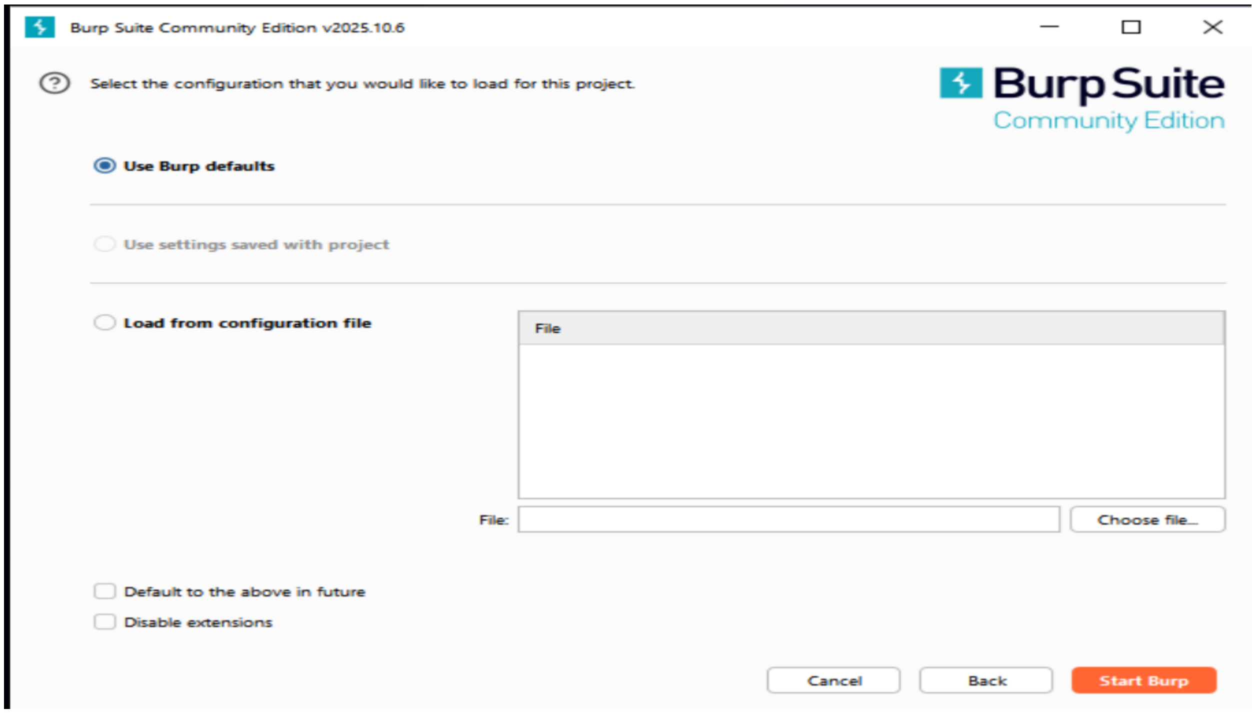
[✓ Temporary project in memory]

New project on disk — Name: — File: — Choose file...

Open existing project — Name | File | Last Accessed — File: — Choose file...

Trust this project file

Pause automated tasks



Burp Suite Community Edition v2025.10.6

Select the configuration that you would like to load for this project.

[✓ Use Burp defaults]

Use settings saved with project

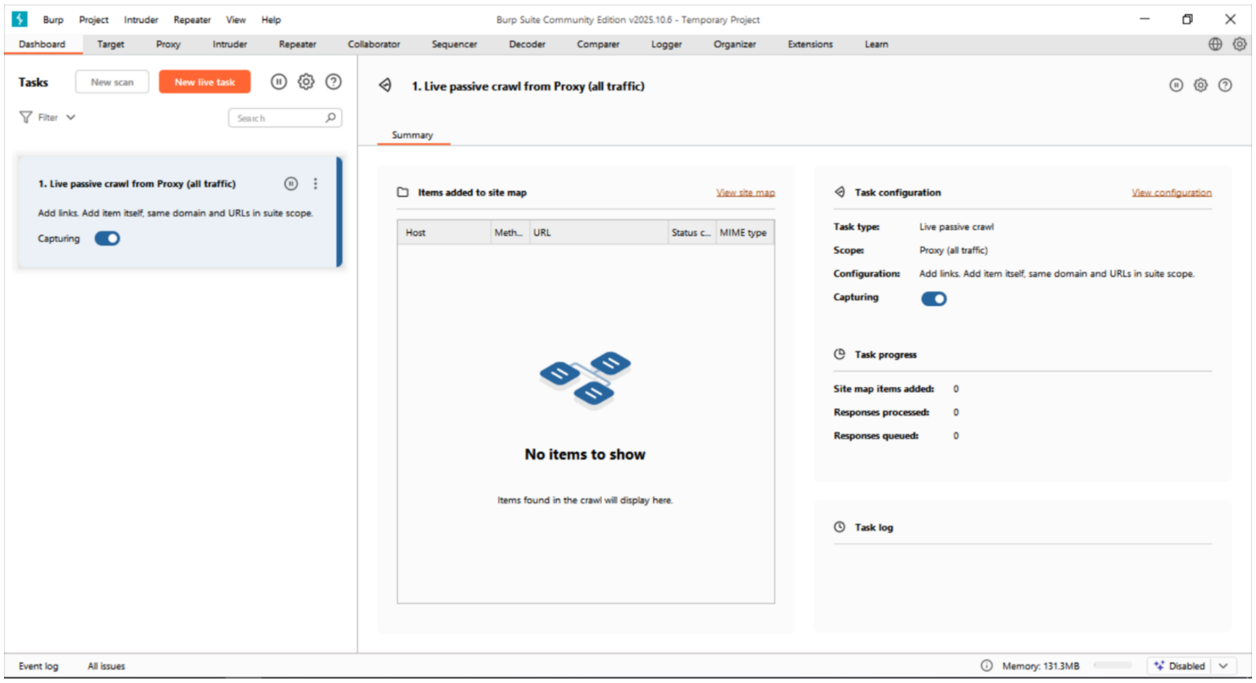
Load from configuration file — File — File: — Choose file...

Default to the above in future

Disable extensions

4.DASHBOARD

The **Dashboard** in Burp Suite Professional serves as the central control interface for managing and monitoring security testing activities. It provides a comprehensive overview of ongoing scans, task queues, system events, and real-time vulnerability detection. This panel is essential for testers who rely on automated or continuous scanning functions.



1. Live passive crawl from Proxy (all traffic)

Add links. Add item itself, same domain and URLs in suite scope.

Capturing

Items added to site map

Host	Meth...	URL	Status c...	MIME type

No items to show

Items found in the crawl will display here.

Task configuration

Task type: Live passive crawl
Scope: Proxy (all traffic)
Configuration: Add links. Add item itself, same domain and URLs in suite scope.
Capturing

Task progress

Site map items added: 0
Responses processed: 0
Responses queued: 0

Task log

Dashboard Components

1. Task Launcher

The Task Launcher enables quick initiation of various automated activities:

- **New Scan**
  Launches a full automated vulnerability scan on the configured target scope.
- **Live Scan**
  Provides continuous monitoring and automated testing of selected targets as new content is discovered.
- **New Crawl**
  Initiates an application crawl to map site structure without performing a security audit.
- **From Configuration**
  Allows users to load and execute predefined scan configurations for standardized testing workflows.

2. Event Log

The Event Log offers real-time visibility into Burp Suite's operational state. It displays:

- Ongoing scan progress and status updates
- Error, warning, and performance notifications

- Crawl and audit activity details
- Timing information for tasks and interactions

This component helps testers maintain awareness of system behavior and troubleshoot issues during scanning.

3. Scan Queue

The Scan Queue manages all scheduled and active automated tasks. It allows testers to:

- View multiple scans running concurrently
- Prioritize certain scanning tasks over others
- Monitor CPU, memory, and thread utilization
- Pause, resume, or cancel scans as required

Effective use of the Scan Queue ensures optimized resource usage and efficient workflow management.

Scan Configuration Management

Burp Suite offers a range of predefined and customizable scan configurations tailored to different testing needs.

Built-in Configurations

- **Lightweight Audit**
  Performs a minimal-impact security audit with reduced scan depth.
- **Fast Application Scan**
  Balances performance and coverage for quick assessments.
- **Deep Application Scan**
  Conducts an extensive and comprehensive audit, generating the highest coverage (but requires more time).
- **Crawl Only**
  Maps application structure without performing vulnerability analysis.

Custom Configurations

Users can create custom scan profiles to suit specific testing requirements. Options include:

- Defining crawl limits and behavior
- Selecting specific audit checks
- Adjusting performance, throttling, and resource usage
- Exporting or importing configurations to maintain team consistency

Custom profiles enhance precision and standardization in professional testing environments.

Practical Usage Example

- **Initial Assessment**
  Initiate a new scan on the defined and authorized scope using the Task Launcher.

- **Progress Monitoring**
  Review real-time updates and logs within the Event Log panel.
- **Issue Review**
  Select identified vulnerabilities for detailed analysis, evidence review, and remediation guidance.
- **Report Generation**
  After completion, export the scan results in a structured report format for documentation or client delivery.

Edition-Specific Availability

- **Professional Edition**
  Full access to dashboard functionality, scan management, and live scanning.
- **Community Edition**
  Limited dashboard capabilities with basic task initiation and no automated auditing.
- **Enterprise Edition**
  Provides a centralized, organization-wide dashboard for managing multiple scanning nodes and large-scale automated testing.

5. PROJECT TYPES (TEMPORARY, DISK-BASED)

Temporary Project

Suitable for quick tests, demos, or small tasks.

- No file saved on disk
- Data exists only during the active session
- Faster to load and close

New Project on Disk

Best for full security assessments.

- Saves all traffic, tools activity, configurations, crawl data, and scanner issues
- Enables backup and long-term analysis
- More stable for large engagements

6. CONFIGURATION WIZARD

Burp Suite shows a Configuration Wizard (Professional edition primarily) after project setup. It allows:

- Selecting default scan configurations
- Enabling updates
- Setting performance limits (threads, memory usage)
- Importing custom CA certificates and extensions
- Loading pre-configured profiles

This helps standardize penetration testing workflows.

7. CRITICAL FIRST-TIME CONFIGURATION

Define Target Scope

- Limit Burp to test only allowed domains
- Prevent accidental scanning of external sites
- Improves performance of automated features

Enable/Disable Intercept

- Interception ON to capture requests
- Interception OFF for browsing normally

Configure Burp Proxy listener

Default listener:

- 127.0.0.1
- Port 8080

Install CA Certificate (required for HTTPS interception)

Discussed later in detail.

Update extensions (optional)

8. INSTALLING BURP’S CA CERTIFICATE (FOR HTTPS INTERCEPTION)

Purpose

HTTPS uses encryption. Burp Suite needs its own root certificate so that it can decrypt HTTPS traffic while acting as a man-in-the-middle (MITM).

Steps

- Open Burp Suite
- Go to: Proxy → Options → Import / Export CA Certificate
- Export certificate as **DER** format
- Import it in the browser’s certificate store
- Mark it as trusted for web traffic

9. CONFIGURING BROWSER PROXY (127.0.0.1:8080)

Steps

- Open Firefox Settings
- Scroll to Network Settings
- Choose Manual Proxy Configuration
- Set:
  - **HTTP Proxy:** 127.0.0.1
  - **Port:** 8080
- Enable “Use this proxy for all protocols”
- Save settings

Outcome

All browser traffic now routes through Burp Suite for interception and analysis.
Your PDF confirms this process successfully.

10. DISABLING OR MANAGING BROWSER CA CERTIFICATES

To remove Burp's certificate:

- Open Firefox: Settings → Privacy & Security → Certificates
- Go to "View Certificates"
- Find "PortSwigger CA"
- Delete or untrust it

Why is this needed?

- To prevent Burp intercepting traffic when work is completed
- To restore normal browser security
- To avoid MITM warnings when Burp is not running

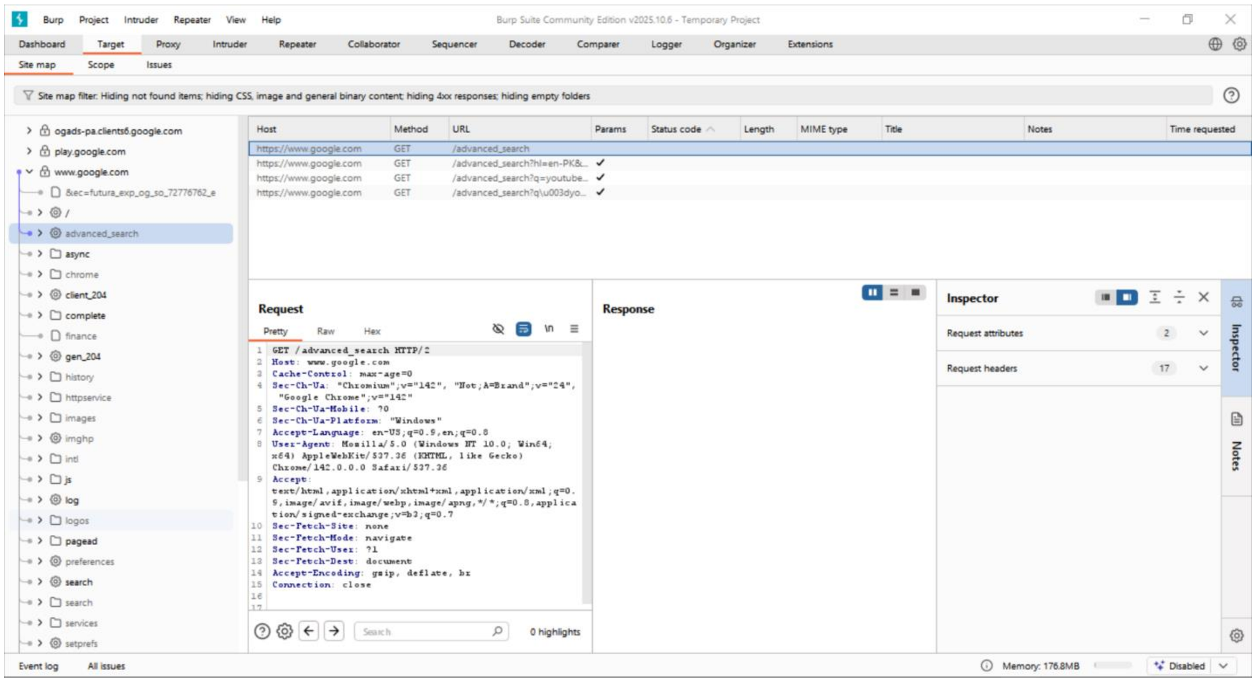
TARGET TAB AND SITE MAP MANAGEMENT

The **Target** tab in Burp Suite serves as the central repository for all information collected about the target application. It provides a structured, hierarchical view of the application's attack surface and acts as the foundation for systematic and organized penetration testing.

1.SITE MAP COMPONENTS

The Site Map presents a detailed breakdown of all discovered content within the target application. It typically includes:

- **Hosts and Domains**
  Displays all identified domains and subdomains in a tree-structured format.
- **URLs and Directories**
  Shows the entire structural layout of the application, including folders, endpoints, and resources.
- **Parameters**
  Lists all detected GET and POST parameters with their associated values.
- **Cookies**
  Displays session identifiers, authentication cookies, and other stored cookie values.
- **Requests and Responses**
  Stores full HTTP request and response data for review, replay, and analysis.


Burp Suite Community Edition v2025.10.6 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Site map Scope Issues
Request
Pretty Raw Hex
Response
Inspector
Request attributes 2
Request headers 17
Notes
0 highlights
Event log All issues
Memory: 176.8MB
Disabled

Site map filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host	Method	URL	Params	Status code	Length	MIME type	Title	Notes	Time requested
https://www.google.com	GET	/advanced_search							
https://www.google.com	GET	/advanced_search?hl=en-PK&...	✓						
https://www.google.com	GET	/advanced_search?q=youtube...	✓						
https://www.google.com	GET	/advanced_search?q\u003dyo...	✓						

```
GET /advanced_search HTTP/2
Host: www.google.com
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="142", "Not;A=Brand";v="24",
 "Google Chrome";v="142"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/142.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.
9,image/avif,image/webp,image/apng,*/*;q=0.8,applica
tion/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Connection: close
```

ogads-pa.clients6.google.com
play.google.com
www.google.com
&ec=futura_exp_og_so_72776762_e
/
advanced_search
async
chrome
client_204
complete
finance
gen_204
history
httpservice
images
imghp
intl
js
log
logos
pagead
preferences
search
search
services
setprefs

2.SCOPE CONFIGURATION

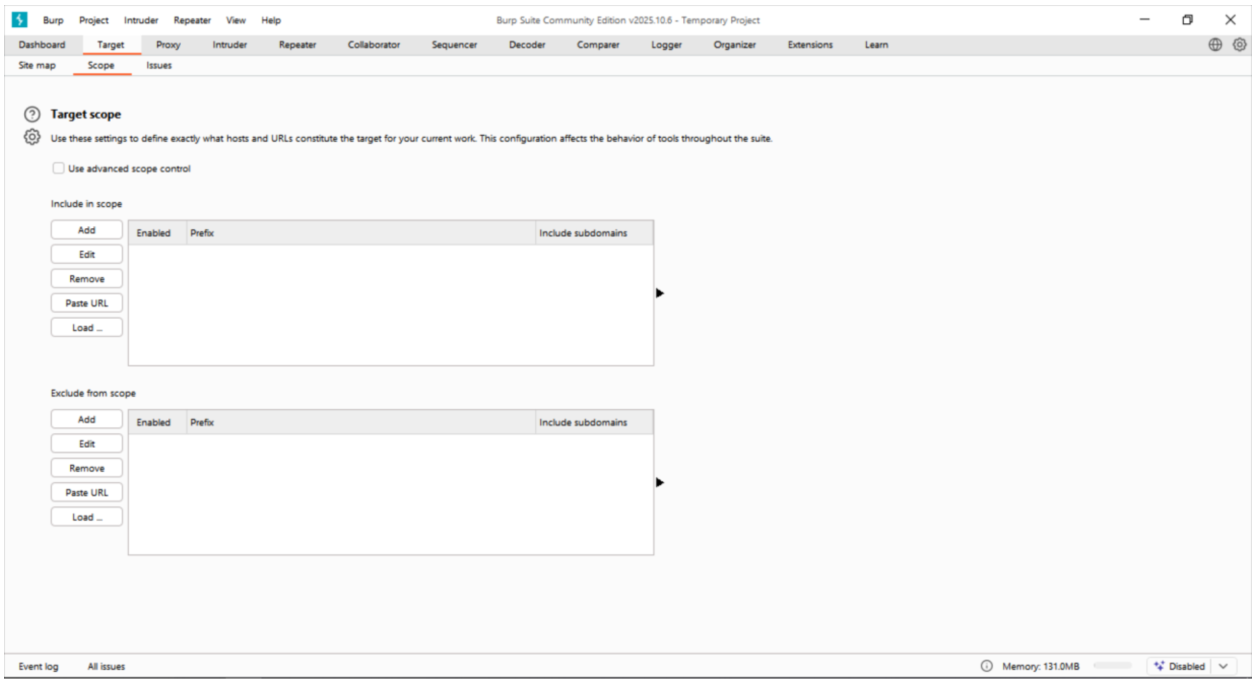
Scope configuration is an essential step in professional penetration testing. Proper scope definition ensures that testing activities are restricted to authorized assets and prevents accidental interaction with third-party systems.

Adding Items to Scope

- Right-click on a host, directory, or URL within the Site Map.
- Select **"Add to scope"**.
- Define inclusion rules based on host, directory, or specific file paths.

3.SCOPE SETTINGS (TARGET → SCOPE)

- **Include in Scope**
  Identifies the exact domains, folders, and endpoints that are authorized for testing.
- **Exclude from Scope**
  Prevents Burp Suite from sending traffic to unauthorized systems or unrelated external resources.
- **Advanced Scope Control**
  Allows the use of regular expressions for fine-grained scope definition.



Target scope

Use these settings to define exactly what hosts and URLs constitute the target for your current work. This configuration affects the behavior of tools throughout the suite.

Use advanced scope control

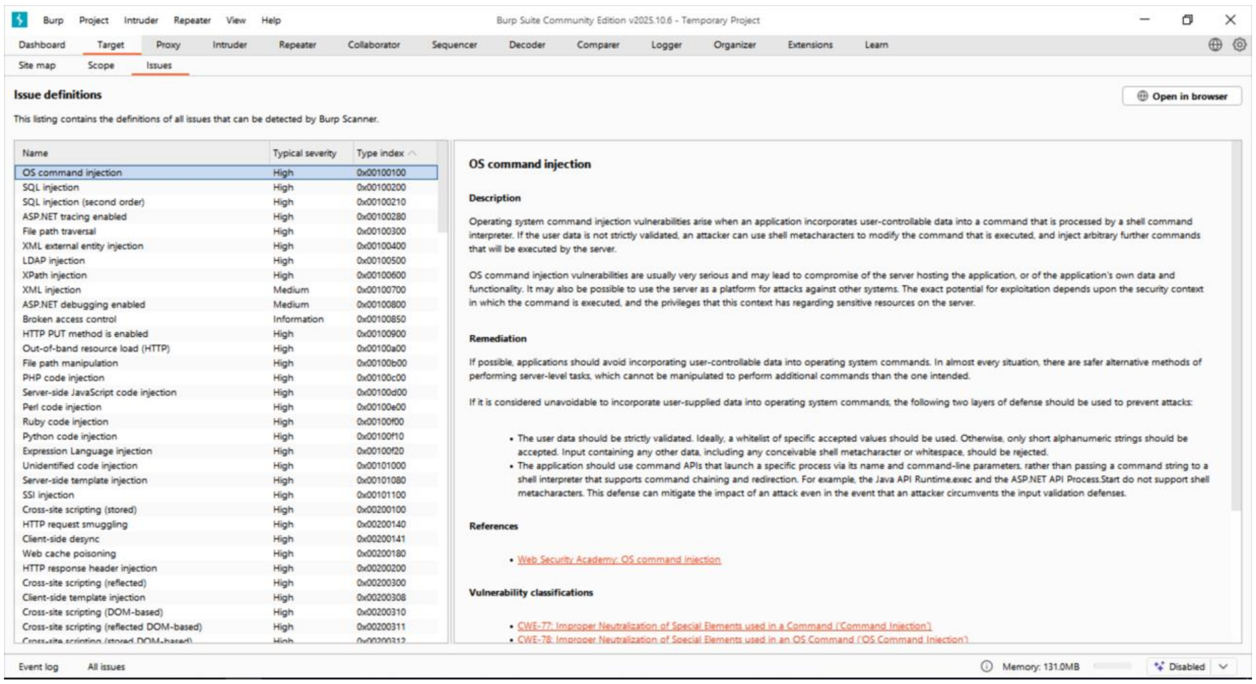
Include in scope

Exclude from scope

4.ISSUE DEFINITIONS

Burp Suite provides a comprehensive vulnerability taxonomy to maintain consistency and clarity when reporting findings. Each identified issue includes:

- **Standardized Classifications**
  References to industry standards such as CWE and WASC.
- **Severity Ratings**
  Categorized as Critical, High, Medium, Low, or Informational.
- **Remediation Guidance**
  Clear, actionable recommendations for developers to correct the vulnerability.


Burp Suite Community Edition v2025.10.6 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Site map Scope Issues
Open in browser
Event log All issues
Memory: 131.0MB
Disabled

Issue definitions

This listing contains the definitions of all issues that can be detected by Burp Scanner.

Name	Typical severity	Type index
OS command injection	High	0x00100100
SQL injection	High	0x00100200
SQL injection (second order)	High	0x00100210
ASP.NET tracing enabled	High	0x00100280
File path traversal	High	0x00100300
XML external entity injection	High	0x00100400
LDAP injection	High	0x00100500
XPath injection	High	0x00100600
XML injection	Medium	0x00100700
ASP.NET debugging enabled	Medium	0x00100800
Broken access control	Information	0x00100850
HTTP PUT method is enabled	High	0x00100900
Out-of-band resource load (HTTP)	High	0x00100a00
File path manipulation	High	0x00100b00
PHP code injection	High	0x00100c00
Server-side JavaScript code injection	High	0x00100d00
Perl code injection	High	0x00100e00
Ruby code injection	High	0x00100f00
Python code injection	High	0x00100f10
Expression Language injection	High	0x00100f20
Unidentified code injection	High	0x00101000
Server-side template injection	High	0x00101080
SSI injection	High	0x00101100
Cross-site scripting (stored)	High	0x00200100
HTTP request smuggling	High	0x00200140
Client-side desync	High	0x00200141
Web cache poisoning	High	0x00200180
HTTP response header injection	High	0x00200200
Cross-site scripting (reflected)	High	0x00200300
Client-side template injection	High	0x00200308
Cross-site scripting (DOM-based)	High	0x00200310
Cross-site scripting (reflected DOM-based)	High	0x00200311
Cross-site scripting (stored DOM-based)	High	0x00200312

OS command injection

Description

Operating system command injection vulnerabilities arise when an application incorporates user-controllable data into a command that is processed by a shell command interpreter. If the user data is not strictly validated, an attacker can use shell metacharacters to modify the command that is executed, and inject arbitrary further commands that will be executed by the server.

OS command injection vulnerabilities are usually very serious and may lead to compromise of the server hosting the application, or of the application's own data and functionality. It may also be possible to use the server as a platform for attacks against other systems. The exact potential for exploitation depends upon the security context in which the command is executed, and the privileges that this context has regarding sensitive resources on the server.

Remediation

If possible, applications should avoid incorporating user-controllable data into operating system commands. In almost every situation, there are safer alternative methods of performing server-level tasks, which cannot be manipulated to perform additional commands than the one intended.

If it is considered unavoidable to incorporate user-supplied data into operating system commands, the following two layers of defense should be used to prevent attacks:

- The user data should be strictly validated. Ideally, a whitelist of specific accepted values should be used. Otherwise, only short alphanumeric strings should be accepted. Input containing any other data, including any conceivable shell metacharacter or whitespace, should be rejected.
- The application should use command APIs that launch a specific process via its name and command-line parameters, rather than passing a command string to a shell interpreter that supports command chaining and redirection. For example, the Java API Runtime.exec and the ASP.NET API Process.Start do not support shell metacharacters. This defense can mitigate the impact of an attack even in the event that an attacker circumvents the input validation defenses.

References

- Web Security Academy: OS command injection

Vulnerability classifications

- CWE-77: Improper Neutralization of Special Elements used in a Command ('Command Injection')
- CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')

5.PRACTICAL USAGE WORKFLOW

- **Initial Reconnaissance**
  Use the Spider or Crawler to automatically discover application content. Findings appear in the Site Map.
- **Scope Definition**
  Add only authorized domains and paths to the scope before any active scanning.
- **Focused Testing**
  Use the "In Scope Only" filter to narrow the Site Map and avoid unnecessary noise.
- **Issue Management**
  Review, validate, and organize vulnerabilities reported within the Target tab.

6.BEST PRACTICE

Always define scope **before** running automated scans. This prevents Burp Suite from interacting with unauthorized systems and ensures ethical, controlled, and legally compliant testing.

MAIN TOOLS IN BURPSUITE AND THEIR FUNCTIONS

Burp Suite includes several built-in tools that help in performing different types of webapplication security testing. Each tool has its own purpose, but they all work together using the data collected by the Burp Proxy. Understanding these tools is important because they form the core of the testing process. Below are the main tools in Burp Suite and their roles in burpsuite.

- Spider
- Scanner
- Intruder
- Repeater
- Sequencer
- Decoder
- Comparer

1. Spider:

The Spider tool is used for crawling and mapping websites. When a tester starts the Spider, it automatically collects links, pages, and directories of a web application. This helps in understanding the structure of the website. The Spider is useful during the beginning of testing because it shows what areas of the website need to be analyzed. It also finds hidden pages or parameters that might be missed manually.

2. Scanner:

The scanner is the most powerful and useful tool in brupsuite. It automatically tests application for security vulnerabilities such as SQL injection, XSS, insecure cookies and server misconfigurations. The Scanner sends different payloads to the server and observes how application responses. This saves time for tester and finds common vulnerabilities quickly.

3. Intruder:

The Intruder tool is designed for automated attacks. It sends multiple requests with different inputs to test how the server reacts. This tool is commonly used for brute force attacks, fuzzing, testing login forms, and checking for input validation weaknesses. Intruder allows the tester to modify payloads, choose attack positions and set different attack modes. It is a very powerful tool when the tester wants to automate repetitive tasks.

4. Sequencer:

The Sequencer tool analyzes the randomness and predictability of session tokens, cookies, and other values. Web applications should generate secure and u npredictable predictable tokens to prevent attacks like session hijacking. S equencer checks how strong and random these tokens are. If the tokens are weak and predictable, then the application becomes vulnerable.

5. Repeater:

The Repeater tool is used for manual testing. It allows the tester to send the same request multiple times with different modifications. When a tester wants to analyze how the server responds to specific inputs, they use the Repeater. This tool is great for testing parameters, trying different values, and manually checking for vulnerabilities. Repeater gives full control to the tester and is often used along with the Proxy.

6. Decoder :

Burp Decoder is a data transformation tool that converts information between various encoding and decoding formats. It supports multiple encoding schemes including URL encoding, Base64, hexadecimal, HTML entities, and various cryptographic hash functions (MD5, SHA-1, SHA-256, etc.). This tool is essential for analyzing encoded parameters, cookies, tokens, and preparing payloads for security testing. Unlike Repeater, which focuses on request manipulation, Decoder specializes in data format conversion.

7. Comparer:

Burp Comparer is a simple yet highly effective tool used for analyzing differences between two items, such as requests, responses, tokens, encoded strings, error messages, or any data collected during testing. Even small differences can reveal important security information, and Comparer helps testers identify these variations clearly and accurately.

TOOLS IN DETAIL

1.BURP SEQUENCER (DETAILED EXPLANATION)

Burp Sequencer is a specialized tool designed to analyze the randomness and predictability of session tokens, cookies, CSRF tokens, and other security-critical values. It performs statistical analysis to determine whether these values are generated with sufficient entropy to prevent attacks like session hijacking, token prediction, or replay attacks.

How Sequencer Works

- Token Capture: Sequencer captures live tokens from the application through manual browsing or automated sessions
- Statistical Analysis: It applies various statistical tests to evaluate the randomness of the token sample
- Entropy Calculation: Measures the information entropy of the tokens (bits of randomness)
- Result Presentation: Provides detailed reports on token quality with pass/fail recommendations

Key Features

- Live Capture: Real-time token collection from application traffic
- Manual Analysis: Option to load token samples from files
- Comprehensive Testing: Multiple statistical tests including chi-square, Monte Carlo, and correlation analyses
- Detailed Reporting: Graphical representation of randomness analysis

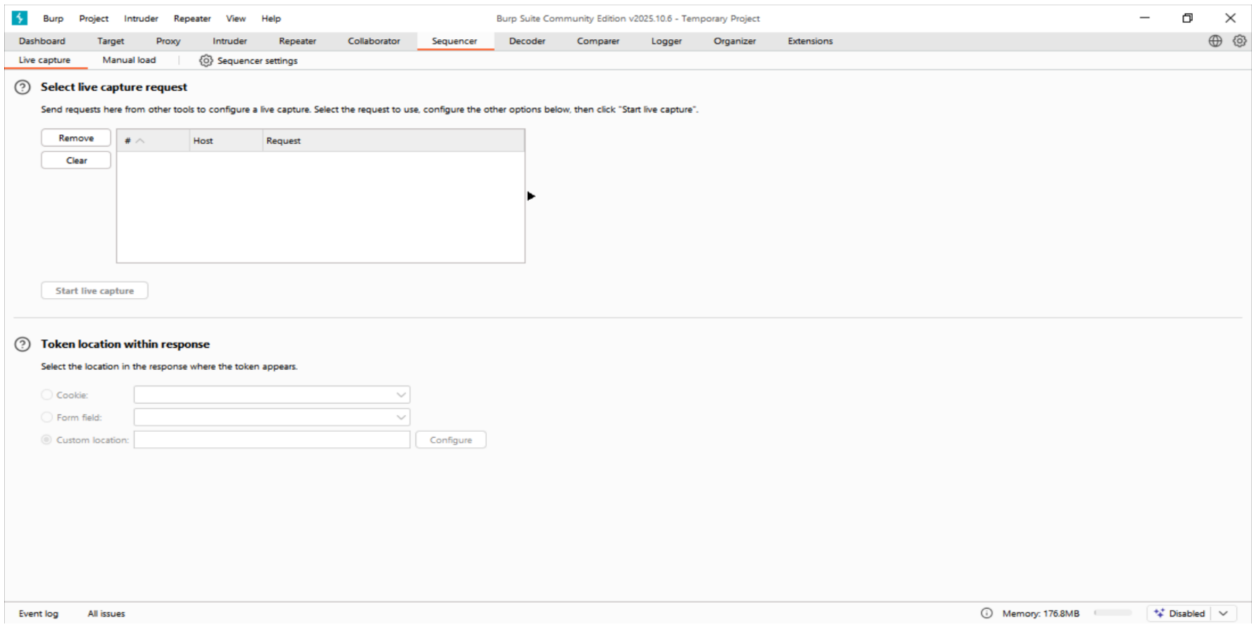
Practical Application

A web application that generates predictable session tokens (e.g., incremental numbers) can be exploited through session hijacking. Sequencer helps identify such weaknesses by analyzing token patterns and providing entropy scores. Generally, tokens should have high entropy (70+ bits) to be considered secure.

Workflow Example

- Browse the application and capture session cookies via Proxy
- Send the cookie values to Sequencer
- Configure capture settings (auto or manual)

- Analyze the results to determine token randomness
- Use findings to assess session management security



Select live capture request

Send requests here from other tools to configure a live capture. Select the request to use, configure the other options below, then click "Start live capture".

Token location within response

Select the location in the response where the token appears.

2.BURP SPIDER (DETAILED EXPLANATION)

The Spider tool in Burp Suite is designed to identify and map the complete structure of a web application. It performs automated web crawling by systematically visiting all accessible links and resources within a site. During this process, it gathers essential information such as URLs, links, parameters, forms, buttons, scripts, and hidden directories. This collected data allows the tester to understand the full scope of the application and determine which areas require detailed security testing.

Burp Spider is particularly valuable during the initial phase of penetration testing. Many applications contain pages that are not directly accessible from the homepage or are only reachable through complex navigation paths. The Spider tool follows every link it encounters, including links inside HTML forms, JavaScript elements, and dynamically generated content. This helps uncover hidden or lesser-used pages that may expose weaknesses or sensitive functionalities.

Another important capability of Spider is its ability to detect input fields and parameters. Since most web vulnerabilities originate in user-input points, identifying all parameters is crucial. Once Spider completes the crawling process, all gathered information becomes available for other Burp Suite tools such as Scanner and Intruder. In summary, Burp Spider enhances coverage, reduces manual effort, and ensures that no significant part of the application is overlooked during testing.

3.BURP SCANNER (DETAILED EXPLANATION)

Burp Scanner is one of the most sophisticated and powerful components of Burp Suite. It performs automated vulnerability assessment by actively testing web applications for security weaknesses. The Scanner sends crafted test requests to the server and analyzes the responses to identify inconsistencies or anomalies that indicate potential vulnerabilities.

The Scanner is capable of detecting a wide variety of common and high-impact vulnerabilities, including:

- SQL Injection

- Cross-Site Scripting (XSS)
- Command Injection
- Directory Traversal
- Insecure Cookies
- Server Misconfigurations
- Broken Authentication
- Sensitive Data Exposure

It analyzes parameters, forms, headers, cookies, and URL structures to identify issues. For example, if the Scanner sends a payload and the server returns an unusual error message or unexpected behavior, it flags this as a potential vulnerability.

One of the strengths of Burp Scanner is its detailed reporting capability. It generates reports that categorize issues by severity (High, Medium, Low, Informational), explain the impact, provide request/response details, and include remediation steps. This assists both penetration testers and developers in addressing security flaws efficiently.

Although the Scanner performs a large number of automated tests, manual validation remains important. Testers typically confirm identified issues using tools like Repeater or Intruder to ensure accuracy and prevent false positives.

Burp Scanner Workflow

- **Target Selection**
  Before scanning, the tester selects the target domain, specific web pages, or individual parameters. Burp Suite allows precise scoping to ensure that only authorized and intended assets are tested.
- **Mapping the Application**
  The Scanner begins by crawling the application to identify its structure. It collects links, forms, cookies, input fields, and scripts to develop a complete understanding of the application's attack surface.
- **Passive Scanning**
  During normal browsing, Burp automatically inspects traffic for security weaknesses. It identifies issues such as insecure headers, exposed tokens, outdated technologies, and missing security configurations. Passive scanning is non-intrusive and does not affect the target system.
- **Analyzing the Results**
  After scanning, Burp Scanner displays a categorized list of findings with color-coded severity levels. Each finding includes detailed technical analysis, evidence, and remediation guidance. Testers can review and verify each issue by examining the associated HTTP requests and responses.

4.BURP INTRUDER (COMPLETE EXPLANATION)

Burp Intruder is a high-performance automation tool used to perform a variety of attacks on web applications. It sends multiple payloads to the server in a systematic manner to analyze how the application responds. Intruder is widely used for testing login pages, forms, search bars, input fields, and any functionality that accepts user input.

Main Purpose of Intruder

The primary objective of Intruder is to automate the process of sending numerous inputs to a web application and monitoring the responses. Intruder helps identify:

- Weak or guessable passwords
- Hidden or undocumented parameters
- Injection points
- Rate-limiting vulnerabilities
- Authorization and access-control weaknesses

Types of Attacks in Intruder

- **Sniper Attack**
  Tests one parameter at a time by injecting payloads individually.
- **Battering Ram**
  Sends the same payload across multiple positions simultaneously.
- **Pitchfork**
  Sends payloads across multiple positions in parallel using synchronized lists.
- **Cluster Bomb**
  Generates combinations of payloads across multiple positions for exhaustive testing.

Payload Options

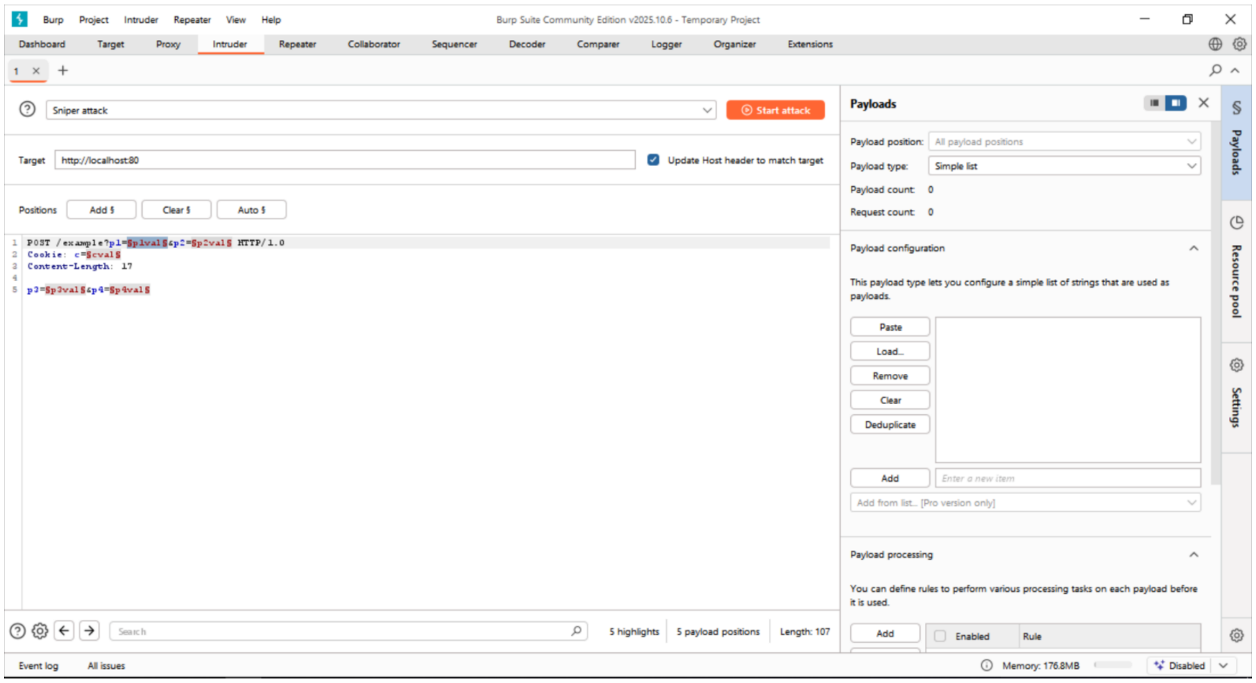
Testers can configure Intruder to use various payload types, including:

- Common password lists
- SQL injection payloads
- XSS payloads
- Custom wordlists
- Numbers, characters, and user-generated data

How Burp Intruder Works

- A request is captured through the Proxy tool.
- The tester sends the request to Intruder.
- Attack positions are selected.
- Payloads are inserted.
- The Intruder attack is launched.
- Responses are analyzed to identify anomalous behavior or vulnerabilities.

Intruder's automation capabilities make it an essential tool for testing input validation, authentication, and application logic.


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Start attack
Add § Clear § Auto §
Paste Load... Remove Clear Deduplicate Add
Event log All issues

Sniper attack

Target: http://localhost:80

Update Host header to match target

```
POST /example?p1=§p1val§&p2=§p2val§ HTTP/1.0
Cookie: c=§cval§
Content-Length: 17

p3=§p3val§&p4=§p4val§
```

5 highlights 5 payload positions Length: 107

Payloads

Payload position: All payload positions
Payload type: Simple list
Payload count: 0
Request count: 0

Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

Enter a new item
Add from list... [Pro version only]

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Enabled Rule

Memory: 176.8MB Disabled

5.BURP REPEATER (DETAILED EXPLANATION)

Burp Repeater is a manual testing tool that allows testers to resend and modify HTTP requests to the server. Unlike Intruder, Repeater provides complete manual control, making it ideal for detailed and precise testing of individual functionalities.

How Repeater Works

- A request is captured using the Proxy.
- The tester sends the request to Repeater.
- The request is manually modified by adjusting parameters, headers, cookies, or payloads.
- The modified request is sent to the server.
- The server's response is analyzed to determine the effect of the changes.

Common Uses of Repeater

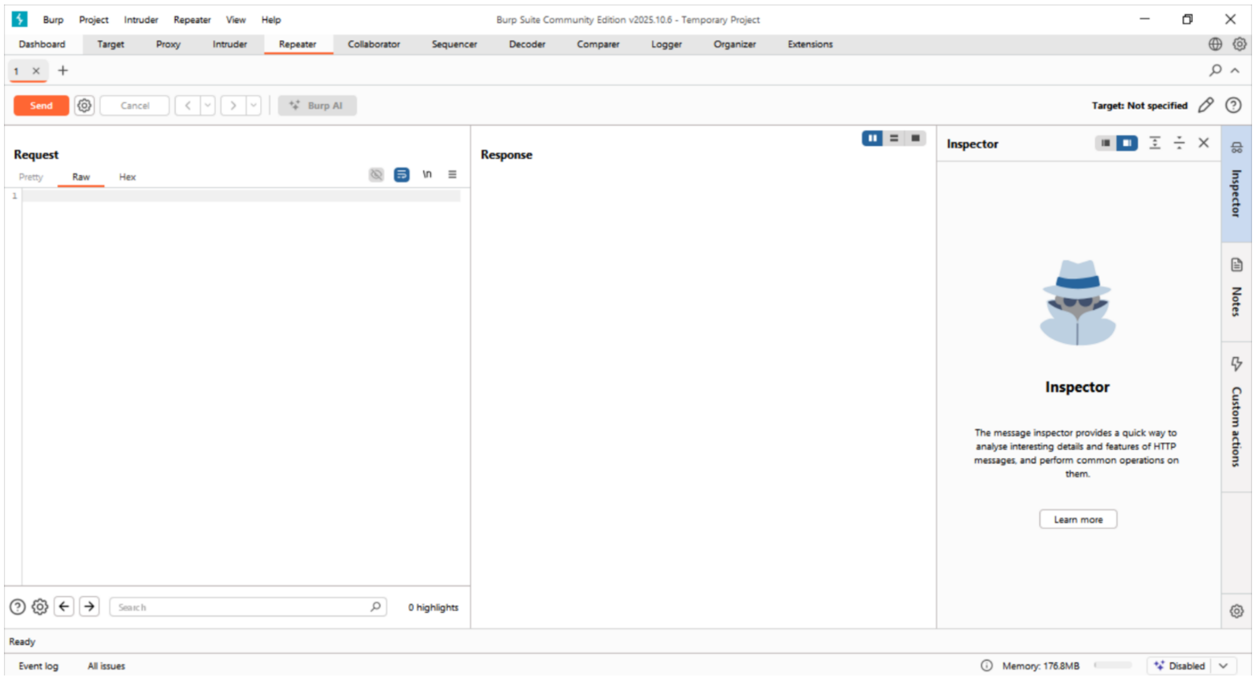
- Confirming SQL injection vulnerabilities
- Testing XSS payloads
- Checking authentication bypass attempts
- Modifying session cookies
- Exploring hidden parameters
- Identifying error message variations
- Testing undocumented endpoints

Benefits

- Provides full manual control
- Excellent for verifying vulnerabilities
- No automation, reducing risk during testing

- Easy to use and interpret
- Suitable for detailed, step-by-step analysis

Repeater plays a critical role in validating findings produced by automated tools such as the Scanner and Intruder.


Burp Suite Community Edition v2025.10.6 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Send
Cancel
Burp AI
Target: Not specified
Request
Pretty Raw Hex
Response
Inspector
The message inspector provides a quick way to analyse interesting details and features of HTTP messages, and perform common operations on them.
Learn more
Notes
Custom actions
Search
0 highlights
Ready
Event log All issues
Memory: 176.8MB
Disabled

6.BURP DECODER (DETAILED EXPLANATION)

Burp Decoder is used to convert data between various encoding formats. Since web applications frequently encode data for security or formatting purposes, Decoder assists in translating this data into readable or testable forms.

Common Encodings Handled by Decoder

- URL encoding
- Base64
- Hexadecimal
- HTML encoding

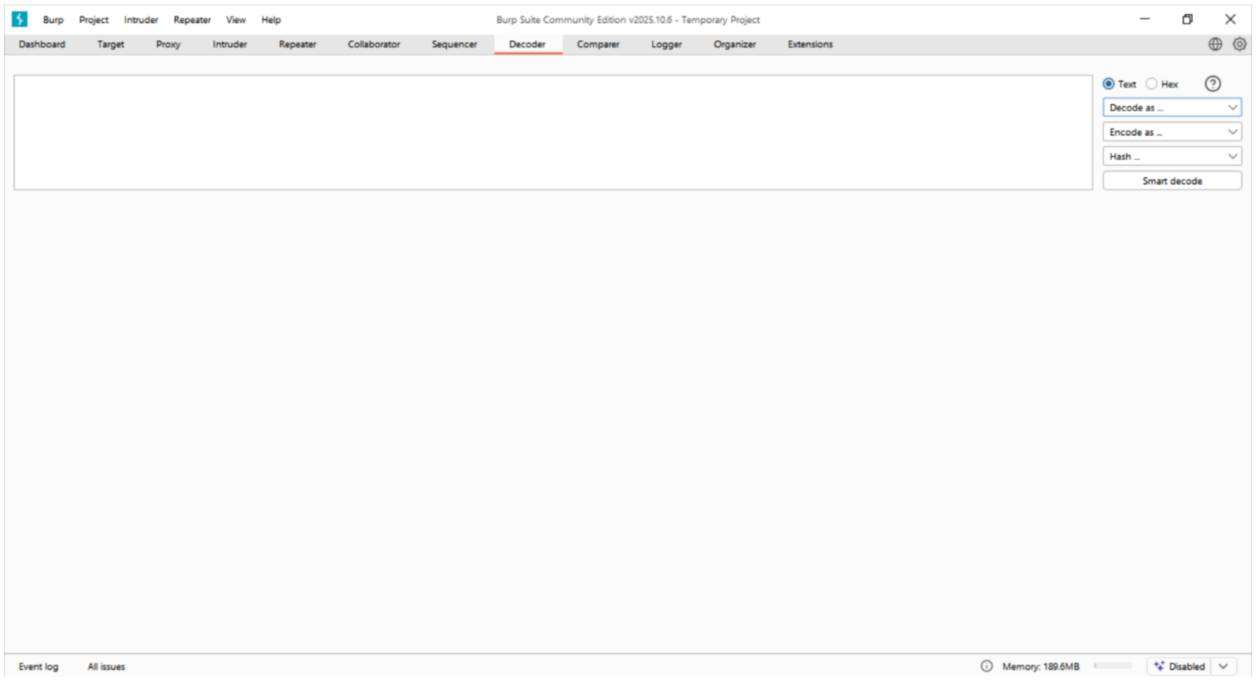
How Decoder Works

- The tester copies encoded data from a request, response, or script.
- The data is pasted into Decoder.
- The tester selects either “Decode” or “Encode.”
- Decoder displays the converted output immediately.

Decoder is valuable for interpreting encoded cookies, session tokens, or hidden parameters. For example:

Encoded cookie:
YWRtaW46MTIzNA==
Base64 decoded value:
admin:1234

This ability to understand encoded values is essential during manual security analysis.


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Text Hex
Decode as ...
Encode as ...
Hash ...
Smart decode
Event log All issues
Memory: 189.6MB
Disabled

7.BURP COMPARER (DETAILED EXPLANATION)

Burp Comparer is a useful tool for comparing two items—such as HTTP requests, responses, or data strings—side-by-side. Even small differences in server responses may indicate important security issues, and Comparer helps testers identify these variations efficiently.

Purpose of Comparer

Comparer is designed to detect differences between:

- Two responses
- Two requests
- Encoded vs. decoded values
- Error messages
- Any similar-looking data that may contain subtle changes

How Comparer Works

- Two items are sent to Comparer from Proxy, Repeater, or Intruder.
- Comparer displays the items side-by-side in separate windows.
- Differences are highlighted automatically.
- The tester reviews these differences to identify unusual behavior or vulnerabilities.

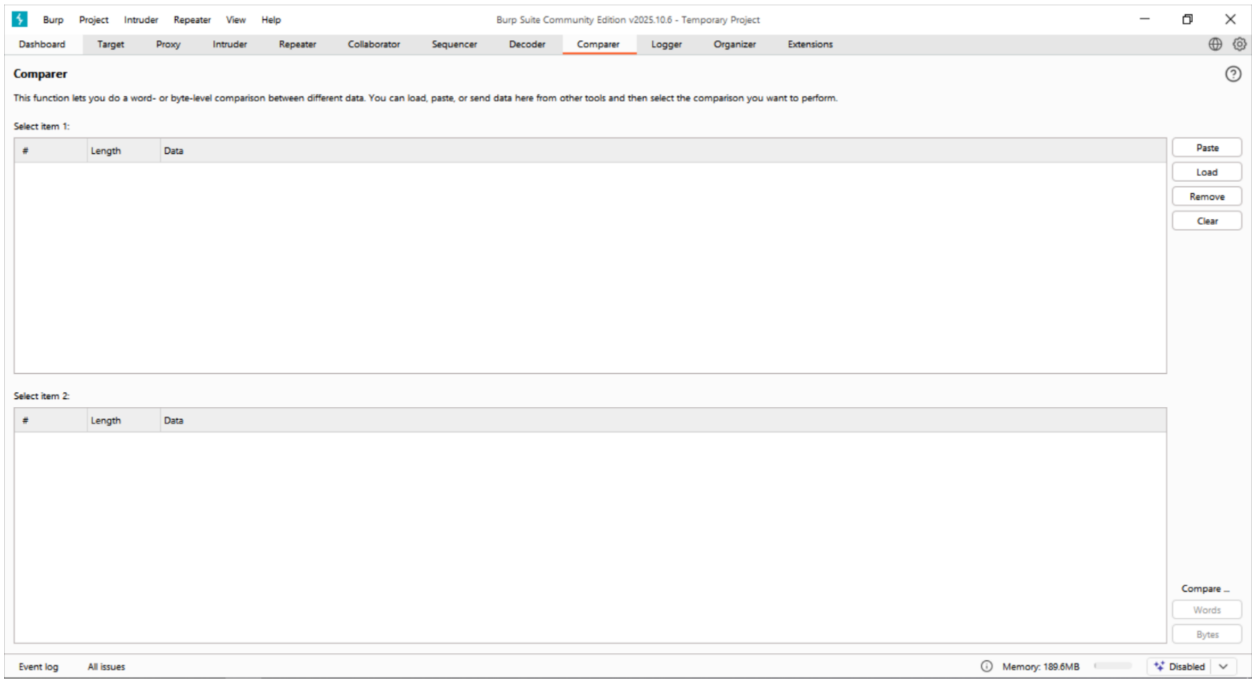
Why Comparer Is Useful

- Detecting differences between valid and invalid login responses
- Identifying changes in error messages
- Finding hidden or modified fields
- Comparing results during fuzzing or brute-force testing
- Analyzing server behavior caused by different inputs

Types of Comparison

- **Byte-level comparison**: Compares data at the raw byte level
- **Word-level comparison**: Compares human-readable content

Comparer streamlines analysis and improves accuracy during manual testing.



This function lets you do a word- or byte-level comparison between different data. You can load, paste, or send data here from other tools and then select the comparison you want to perform.

8.BURP SUITE EXTENDER (DETAILED EXPLANATION)

Burp Suite Extender enhances Burp's functionality by allowing testers to install extensions and integrate custom tools. This makes Burp Suite highly flexible and customizable according to testing needs.

Purpose of Burp Extender

The main purpose of Extender is to expand Burp Suite's capabilities. If a required feature is not available by default, testers can add it through the BApp Store or create their own extensions using the Burp Extender API.

The BApp Store

Burp Suite includes an internal marketplace called the BApp Store, which contains a wide range of community-developed and official extensions. Examples include:

- SQLiPy (SQL injection testing)
- Autorize (access control testing)
- Flow (traffic analysis)
- FakeIP (IP restriction testing)
- Additional scanners, payload generators, automation tools

These extensions improve coverage, accuracy, and efficiency.

How Extensions Work

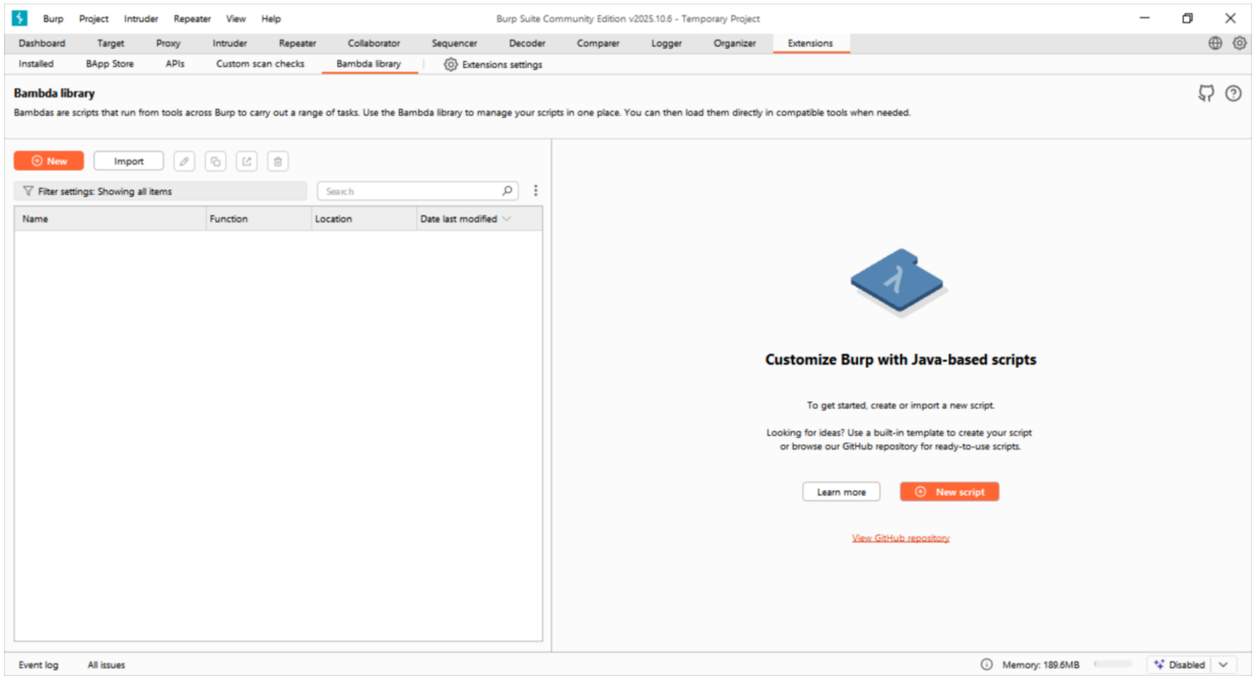
- The tester opens the Extender tab.
- Navigates to the BApp Store section.
- Selects an extension.
- Burp Suite installs and integrates it automatically.
- The new tool or feature becomes immediately usable.

Why Extender Is Useful

- Enhances Burp Suite's default capabilities
- Automates repetitive tasks
- Adds specialized scanners and payloads
- Improves traffic analysis
- Supports advanced testing techniques

Importance in Cybersecurity

Cyber threats evolve continuously. Extensions allow Burp Suite to adapt quickly by adding new testing techniques without needing a full software update. This ensures that testers stay current with emerging vulnerabilities and attack methods.



Bambda library

Bambdas are scripts that run from tools across Burp to carry out a range of tasks. Use the Bambda library to manage your scripts in one place. You can then load them directly in compatible tools when needed.

Name	Function	Location	Date last modified

Customize Burp with Java-based scripts

To get started, create or import a new script.

Looking for ideas? Use a built-in template to create your script or browse our GitHub repository for ready-to-use scripts.

9.BURP COLLABORATOR (DETAILED EXPLANATION)

Burp Collaborator is an out-of-band (OOB) interaction detection service built into Burp Suite. It is specifically designed to identify vulnerabilities where the effects of an attack do not appear directly in HTTP responses. Such vulnerabilities are known as *blind vulnerabilities*. Burp Collaborator works by generating unique payloads that, when executed by a vulnerable system, trigger external interactions (DNS, HTTP, SMTP) with the Collaborator server, thereby confirming the presence of the vulnerability.

How Burp Collaborator Works

- **Payload Generation**
  Collaborator generates unique DNS, HTTP, HTTPS, or SMTP payloads for injection into the application.
- **Payload Delivery**
  The tester places these payloads into potential attack vectors such as parameters, headers, cookies, or file uploads.
- **Callback Detection**
  If the target is vulnerable, it initiates an external interaction with the Collaborator server using the payload.
- **Alert Generation**
  Burp Suite monitors the Collaborator server and notifies the tester when a callback is received, confirming the vulnerability.

Types of Vulnerabilities Detected

Burp Collaborator is highly effective for identifying vulnerabilities that rely on out-of-band behavior. Examples include:

- **Blind Server-Side Request Forgery (SSRF)**
  When the server makes unintended external requests.
- **Blind Cross-Site Scripting (XSS)**
  When malicious scripts execute asynchronously on a victim's browser.
- **Out-of-Band XML External Entity Injection (XXE)**
  When external entity resolution results in DNS or HTTP callbacks.
- **Blind Command Injection**
  When system-level commands execute but do not return visible output.
- **Time-Based SQL Injection Validation**
  Used as secondary confirmation when time delays alone are insufficient.

Collaborator Client Integration

Available in Professional and Enterprise Editions only.

Key Features

- **Integrated Collaborator Client**
  Built directly into the Burp Suite interface for generating and monitoring OOB payloads.
- **Multiple Payload Types Supported**
  DNS, HTTP, HTTPS, and SMTP-based interaction mechanisms.

- **Custom Collaborator Server Support**
  Organizations can deploy private Collaborator instances for confidentiality and internal testing.
- **Configurable Polling Intervals**
  Testers can adjust how frequently Burp Suite checks for incoming interactions.

Practical Workflow Example

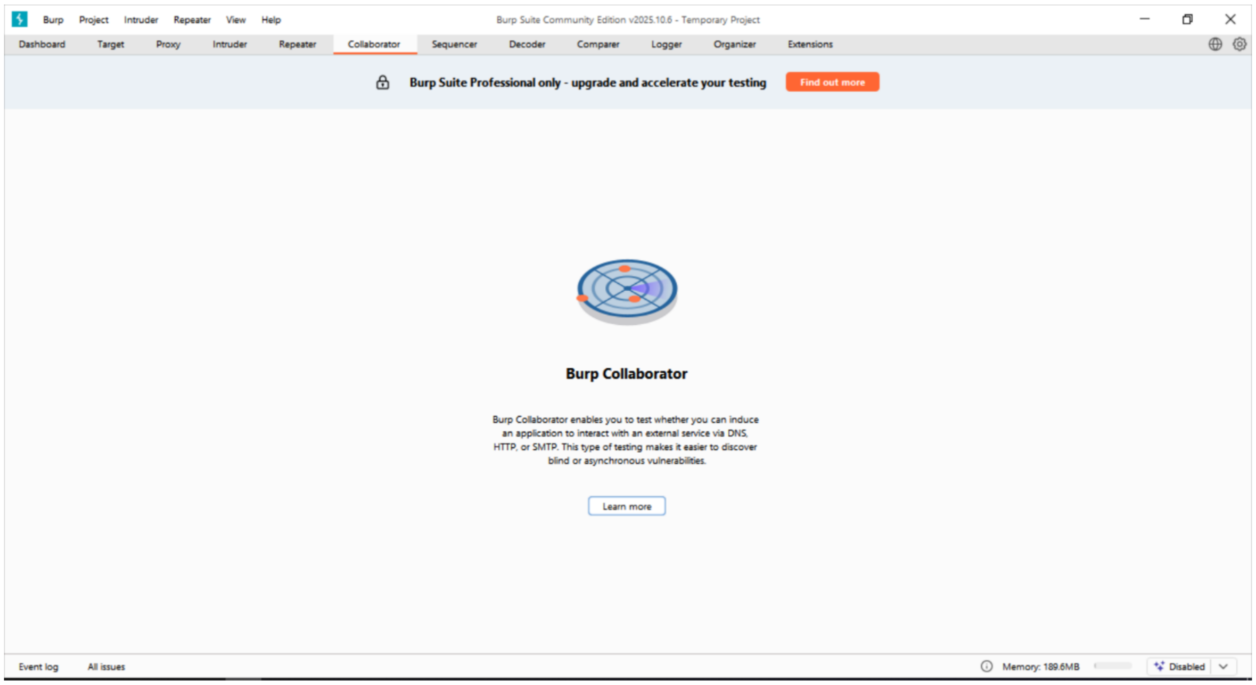
- **Identify a Test Point**
  Locate parameter input fields or data points that may trigger server-side processing.
- **Generate a Collaborator Payload**
  Use the Collaborator Client to create a unique DNS payload such as http://<unique-id>.burpcollaborator.net
- **Inject the Payload**
  Insert the payload into the potential vulnerable parameter or input field.
- **Monitor for Callbacks**
  Open the Collaborator tab and monitor for inbound DNS or HTTP interactions.
- **Confirm the Vulnerability**
  Any callback received from the target server confirms an active OOB vulnerability.

Security Considerations

- **Authorized Testing Only**
  Collaborator can reveal sensitive system interactions; it must be used strictly within authorized environments.
- **Data Privacy Awareness**
  Callback data may include sensitive internal hostnames, credentials, or server information.
- **Network Connectivity Requirements**
  The target environment must allow outbound DNS and HTTP traffic to enable callback detection.

Edition Availability

- **Professional Edition**: Full Collaborator Client functionality is included.
- **Enterprise Edition**: Includes Collaborator integration for automated scanning.
- **Community Edition**: Does *not* include Collaborator. Users cannot generate or monitor Collaborator payloads.


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Find out more
Learn more
Event log All issues
Memory: 189.6MB
Disabled

Burp Suite Professional only - upgrade and accelerate your testing

Burp Collaborator

Burp Collaborator enables you to test whether you can induce an application to interact with an external service via DNS, HTTP, or SMTP. This type of testing makes it easier to discover blind or asynchronous vulnerabilities.

PRACTICAL LAB / TABS PRACTICAL

PRACTICAL EXERCISES ENCODER

Exercise 1: Plain text to Base64 Encode

Objective: Encode text to Base64

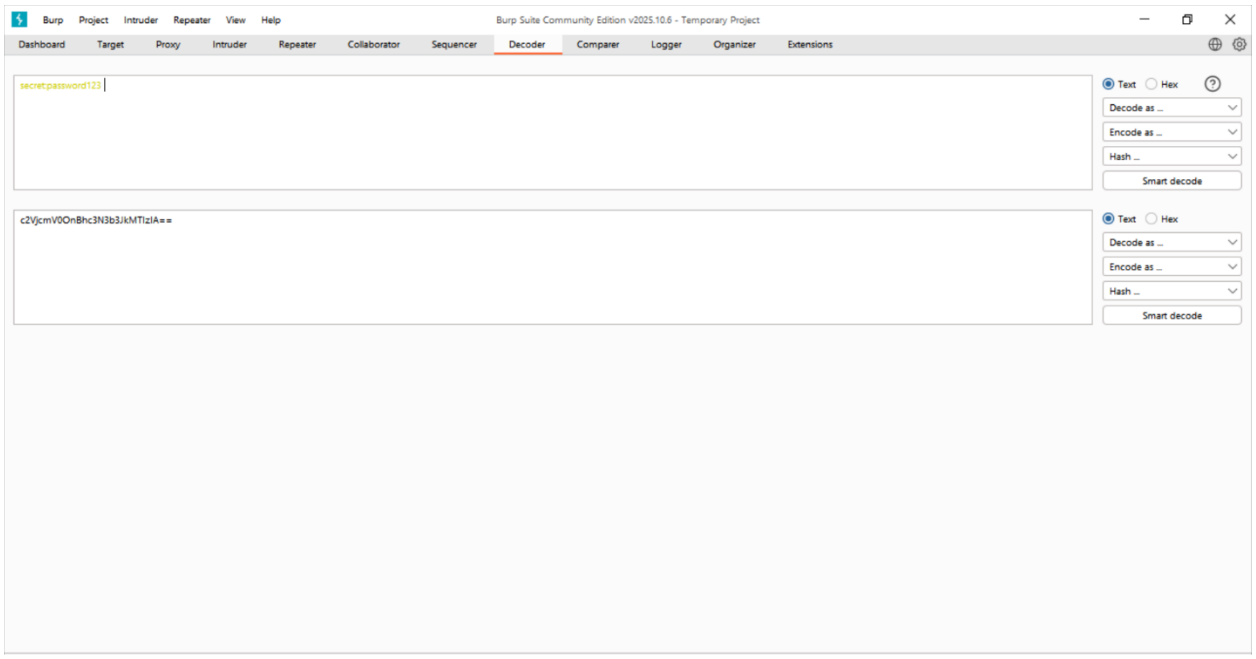
Steps for Encoding:

- Open Burp Suite → Decoder tab
- Input: secret:password123
- Select **"Encode as → Base64"**
- Copy the encoded output

Expected Flow:

Input: secret:password123

Encode → c2VjcmV0OnBhc3N3b3JkMTIzIA==



secret:password123

c2VjcmV0OnBhc3N3b3JkMTIzIA==

Exercise 2: URL Encode

Objective: Encode special characters

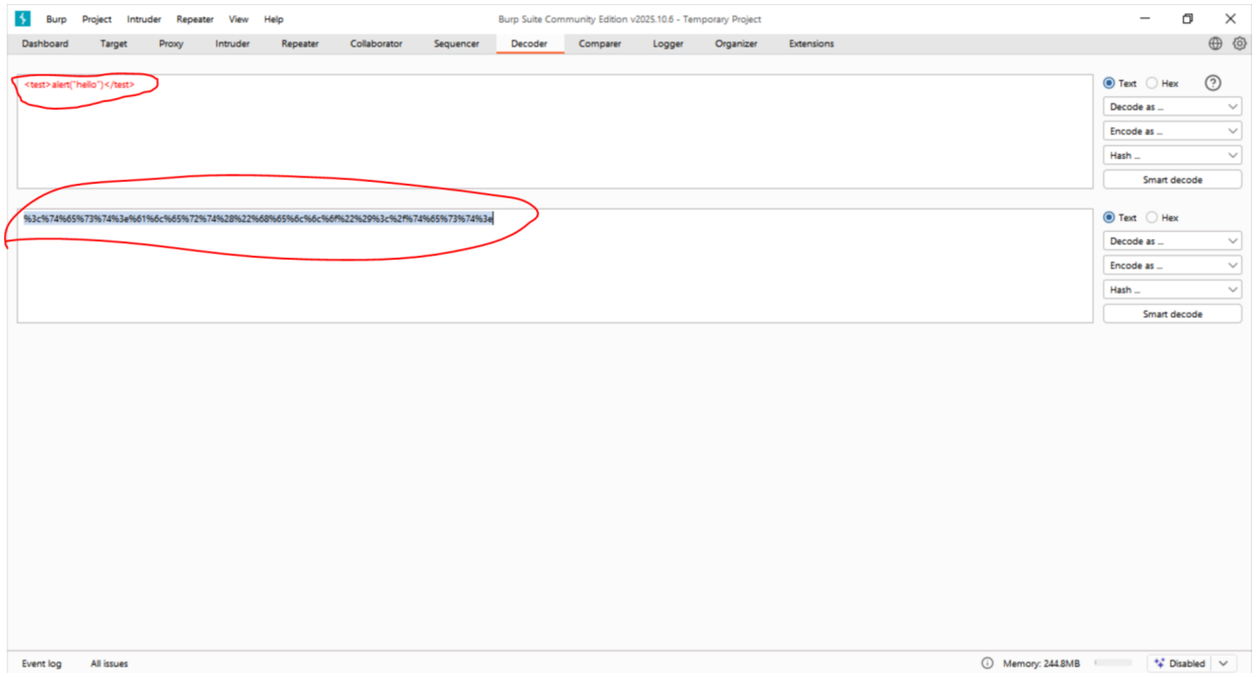
Steps for Encoding:

- Clear input
- Input: <test>alert("hello")</test>
- Select **"Encode as → URL"**
- Copy the encoded output

Expected Flow:

Input: <test>alert("hello")</test>

Encode →
%3c%74%65%73%74%3e%61%6c%65%72%74%28%22%68%65%6c%6c%6f%22%29%3c%2f%74%65%73%74%3e



```
<test>alert("hello")</test>
```

```
%3c%74%65%73%74%3e%61%6c%65%72%74%28%22%68%65%6c%6c%6f%22%29%3c%2f%74%65%73%74%3e
```

Exercise 3: Hex Encode

Objective: Convert text to hexadecimal

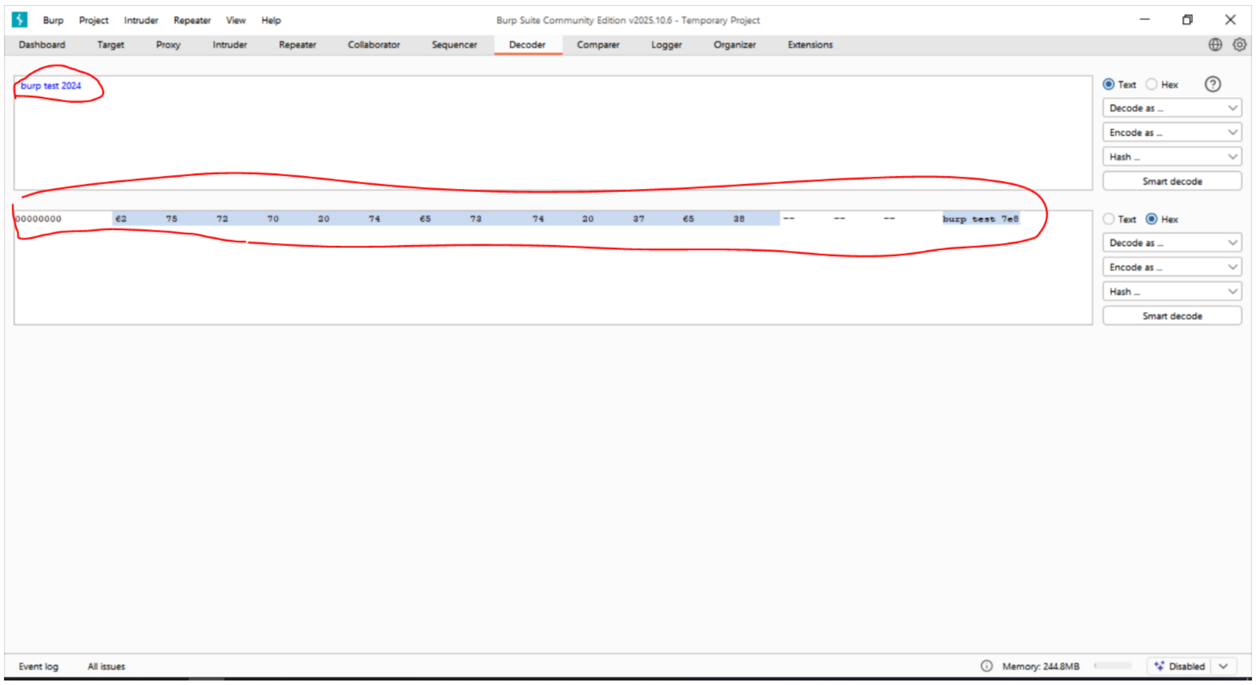
Steps for Encoding:

- Clear input
- Input: burp test 2024
- Select **"Encode as → Hex"**
- Copy the encoded output

Expected Flow:

Input: burp test 2024

Encode → 62 75 72 70 20 74 65 73 74 20 37 65 38


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Text Hex
Decode as ...
Encode as ...
Hash ...
Smart decode
Event log All issues
Memory: 244.8MB
Disabled

```
burp test 2024
```

```
00000000   62  75  72  70  20  74  65  73  74  20  37  65  38  --  --  --   burp test 7e8
```

PRACTICAL EXERCISES DECODER

Exercise 1: Base64 Decoding

Objective: Decode Base64 encoded strings to reveal hidden information

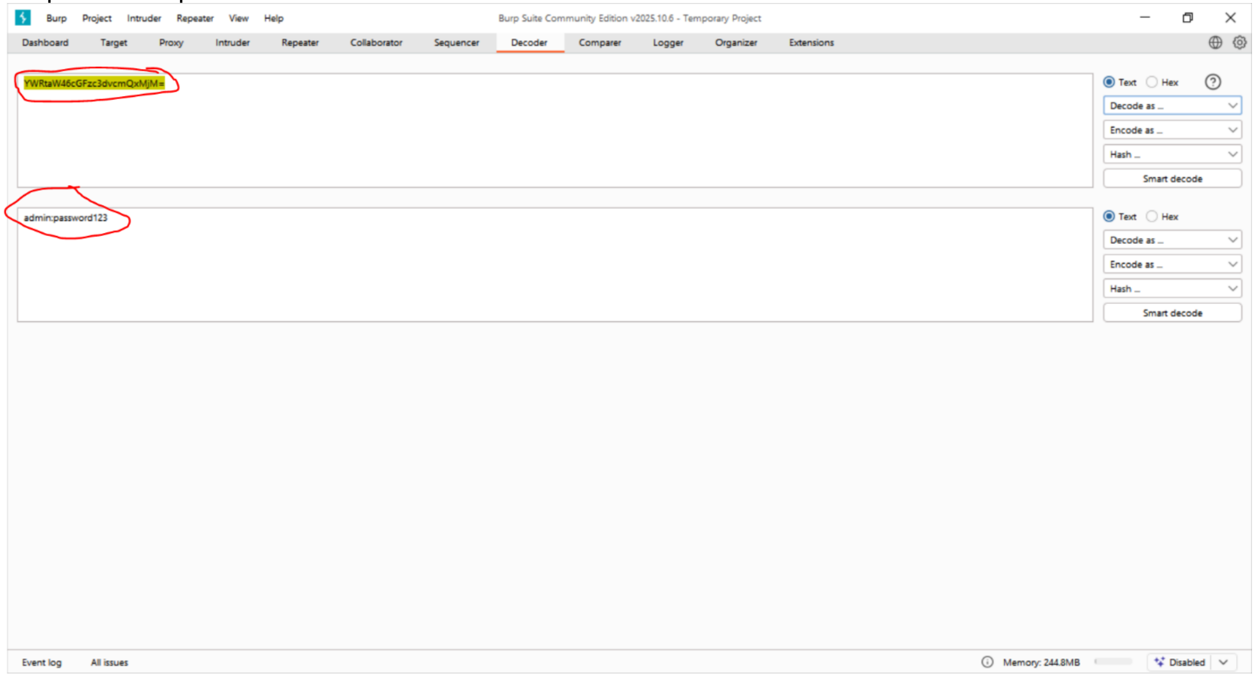
Steps:

- Open Burp Suite → Decoder tab
- Input encoded string in left panel
- Select "Decode as → Base64"
- Observe decoded output in right panel

Test Data 1:

Input: YWRtaW46cGFzc3dvcmQxMjM=

Output: admin:password123


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Text Hex
Decode as ...
Encode as ...
Hash ...
Smart decode
Event log All issues
Memory: 244.8MB
Disabled

YWRtaW46cGFzc3dvcmQxMjM=

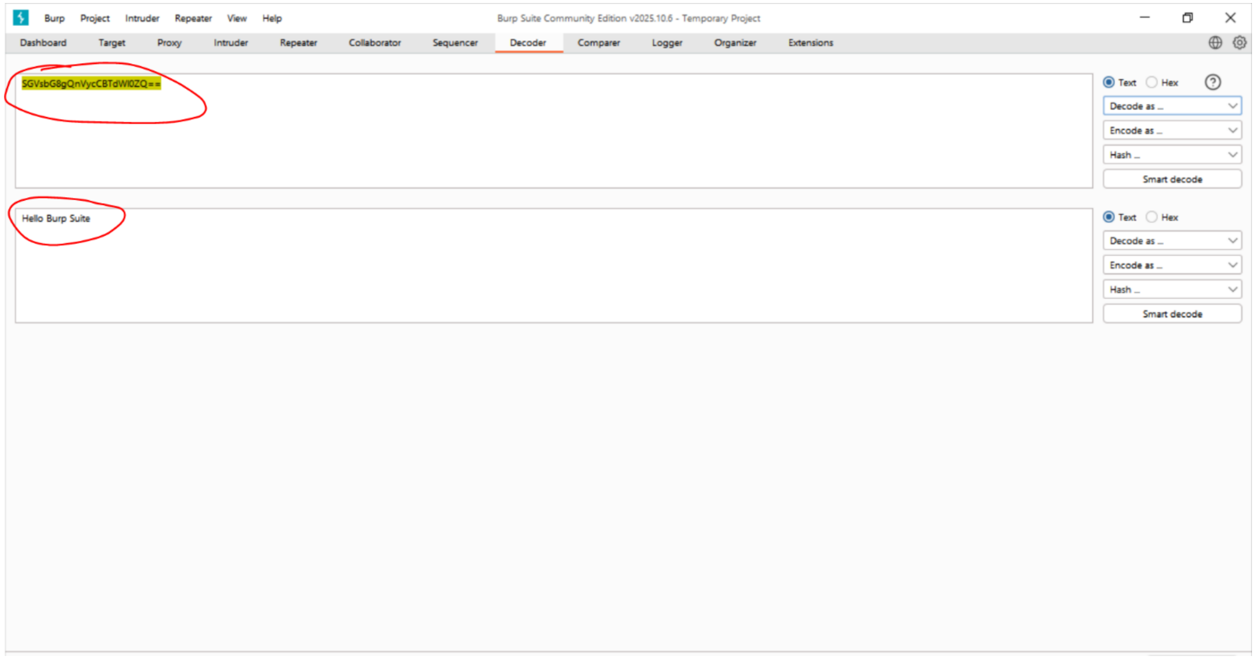
admin:password123

Test Data 2:

bash

Input: SGVsbG8gQnVycCBTdWl0ZQ==

Output: Hello Burp Suite


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Text Hex
Decode as ...
Encode as ...
Hash ...
Smart decode

SGVsbG8gQnVycCBTdWl0ZQ==

Hello Burp Suite

Exercise 2: URL Encoding/Decoding

Objective: Understand URL encoded parameters and special characters

Steps:

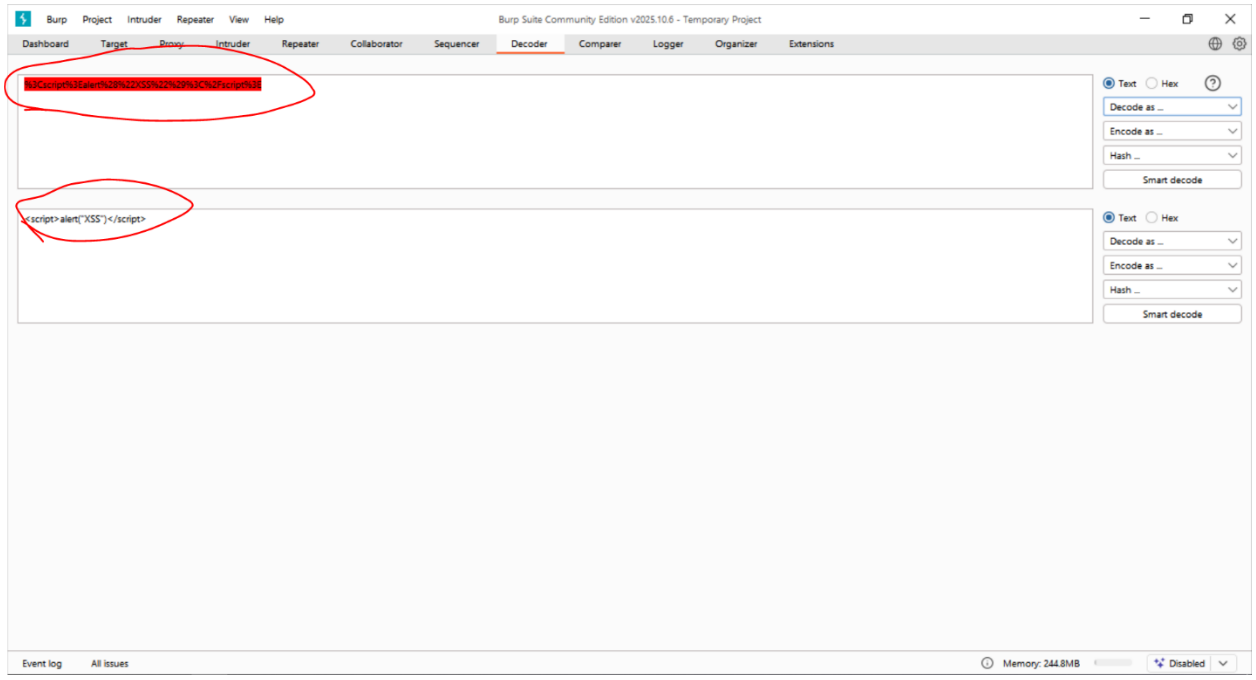
- Clear previous input
- Paste URL encoded string
- Select "Decode as → URL"
- Analyze decoded output

Test Data 1 (Decoding):

bash

Input: %3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E

Output: <script>alert("XSS")</script>



Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Text Hex
Decode as ...
Encode as ...
Hash ...
Smart decode
Event log All issues
Memory: 244.8MB
Disabled

%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E

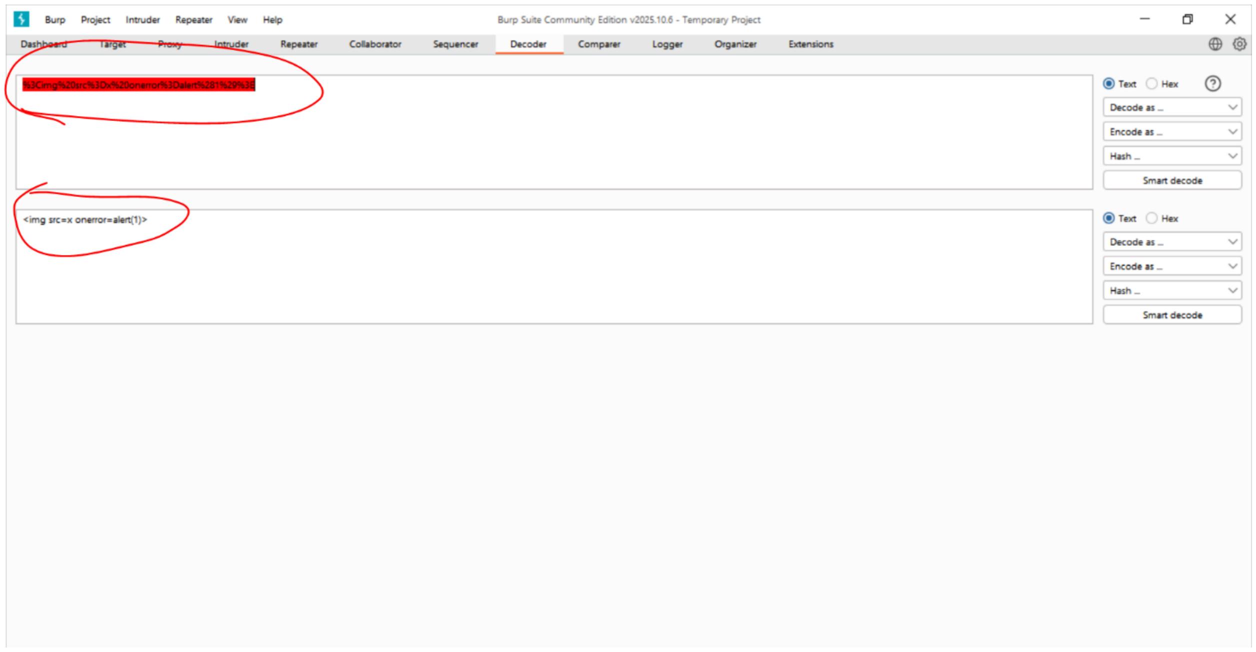
<script>alert("XSS")</script>

Test Data 2 (Encoding):

bash

Input: <img src=x onerror=alert(1)>

Output: %3Cimg%20src%3Dx%20onerror%3Dalert%281%29%3E


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Text Hex
Decode as ...
Encode as ...
Hash ...
Smart decode

```
%3Cimg%20src%3Dx%20onerror%3Dalert%281%29%3E
```

```
<img src=x onerror=alert(1)>
```

Conclusion

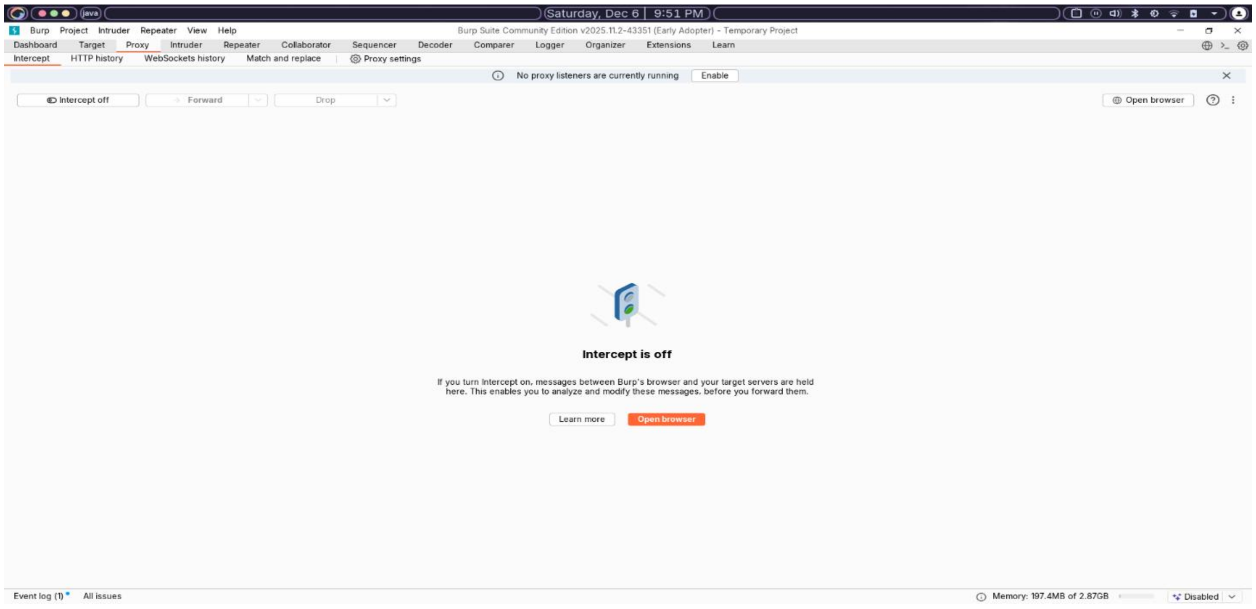
The Burp Decoder practical exercise successfully demonstrated the importance of data encoding/decoding in cybersecurity. Through hands-on practice with Base64, URL encoding, hexadecimal, and hash functions, key skills were developed for analyzing and manipulating encoded data in security testing scenarios.

INTERCEPTION PRACTICLE

Steps Performed

Opening Burp Suite & Checking Intercept

I clicked **Intercept** → **Intercept On** to allow Burp to capture browser requests.



No proxy listeners are currently running

Intercept is off

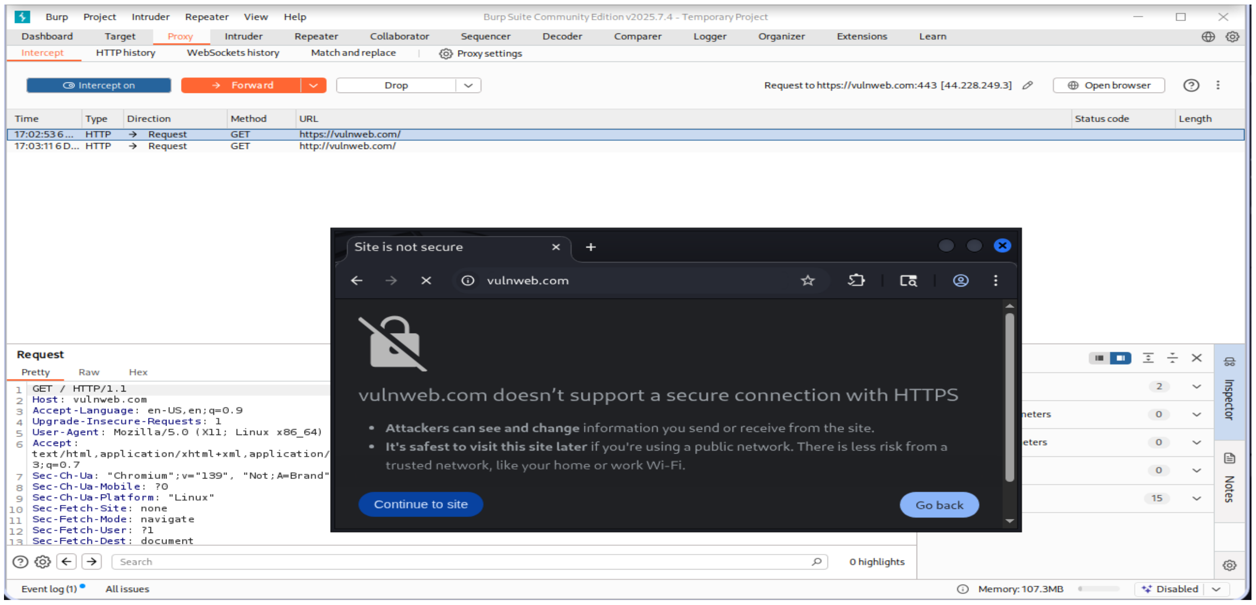
If you turn Intercept on, messages between Burp's browser and your target servers are held here. This enables you to analyze and modify these messages, before you forward them.

Visiting a Website

I opened vulnweb.com. The browser showed a warning:

- "Site is not secure – doesn't support secure connection with HTTPS".

This happened because the Burp CA certificate was not installed.



Request to https://vulnweb.com:443 [44.228.249.3]

```
Time          Type  Direction  Method  URL                     Status code  Length
17:02:53 6... HTTP  → Request  GET     https://vulnweb.com/
17:03:11 6 D... HTTP → Request GET     http://vulnweb.com/
```

Request

```
GET / HTTP/1.1
Host: vulnweb.com
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
Accept:
text/html,application/xhtml+xml,application/
3;q=0.7
Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
```

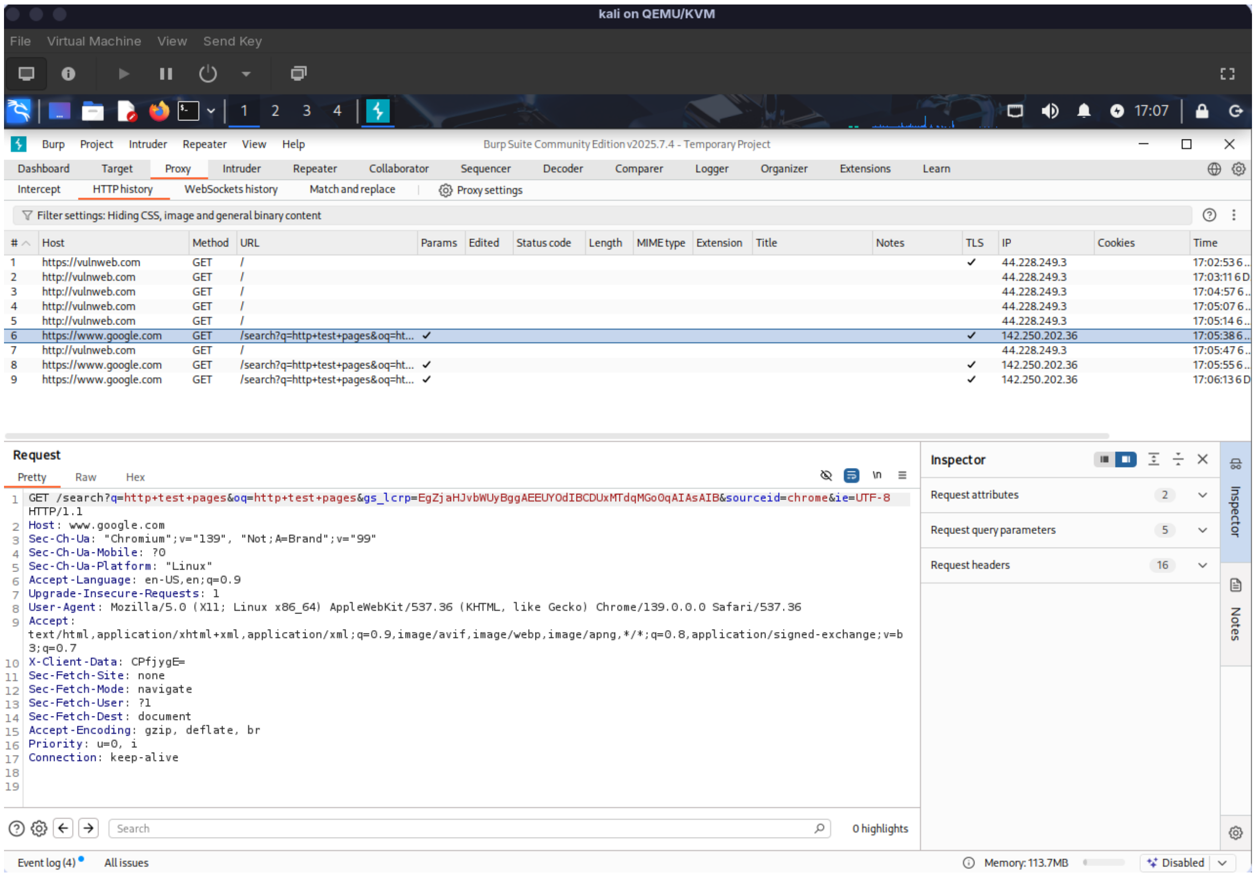
vulnweb.com doesn't support a secure connection with HTTPS

- **Attackers can see and change** information you send or receive from the site.
- **It's safest to visit this site later** if you're using a public network. There is less risk from a trusted network, like your home or work Wi-Fi.

Requests Captured in HTTP History

I checked the **Proxy** → **HTTP history** tab. Burp showed:

- Requests to vulnweb.com
- Requests to Google search
- GET / queries with parameters



Burp Suite Community Edition v2025.7.4 - Temporary Project

Filter settings: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time
1	https://vulnweb.com	GET	/									✓	44.228.249.3		17:02:53 6…
2	http://vulnweb.com	GET	/										44.228.249.3		17:03:11 6 D
3	http://vulnweb.com	GET	/										44.228.249.3		17:04:57 6…
4	http://vulnweb.com	GET	/										44.228.249.3		17:05:07 6…
5	http://vulnweb.com	GET	/										44.228.249.3		17:05:14 6…
6	https://www.google.com	GET	/search?q=http+test+pages&oq=ht…	✓								✓	142.250.202.36		17:05:38 6…
7	http://vulnweb.com	GET	/										44.228.249.3		17:05:47 6…
8	https://www.google.com	GET	/search?q=http+test+pages&oq=ht…	✓								✓	142.250.202.36		17:05:55 6…
9	https://www.google.com	GET	/search?q=http+test+pages&oq=ht…	✓								✓	142.250.202.36		17:06:13 6 D

Request

```
GET /search?q=http+test+pages&oq=http+test+pages&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCDUxMTdqMGo0qAIAsAIB&sourceid=chrome&ie=UTF-8
HTTP/1.1
Host: www.google.com
Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b
3;q=0.7
X-Client-Data: CPfjygE=
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Connection: keep-alive
```

Inspector

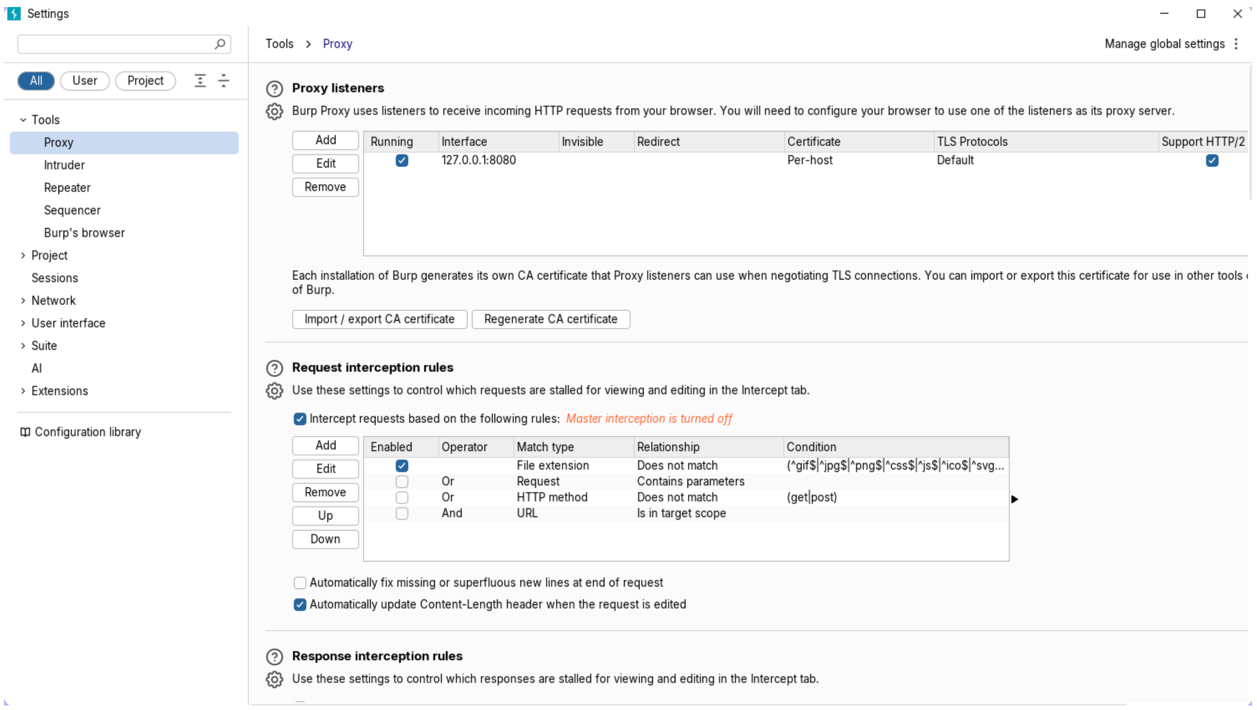
Request attributes	2
Request query parameters	5
Request headers	16

kali on QEMU/KVM
File Virtual Machine View Send Key
17:07
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Pretty Raw Hex
Search
0 highlights
Event log (4) All issues
Memory: 113.7MB
Disabled
Inspector
Notes

Checking Proxy Listener Settings

I opened **Settings** → **Tools** → **Proxy**. I confirmed:

- Proxy listener running at 127.0.0.1:8080
- Certificate: Per-host
- TLS protocols: Default



Tools > Proxy

Proxy listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Running	Interface	Invisible	Redirect	Certificate	TLS Protocols	Support HTTP/2
✓	127.0.0.1:8080			Per-host	Default	✓

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in other tools of Burp.

Request interception rules

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

✓ Intercept requests based on the following rules: *Master interception is turned off*

Enabled	Operator	Match type	Relationship	Condition
✓		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$\|^ico$|^svg...
☐	Or	Request	Contains parameters	
☐	Or	HTTP method	Does not match	(get|post)
☐	And	URL	Is in target scope	

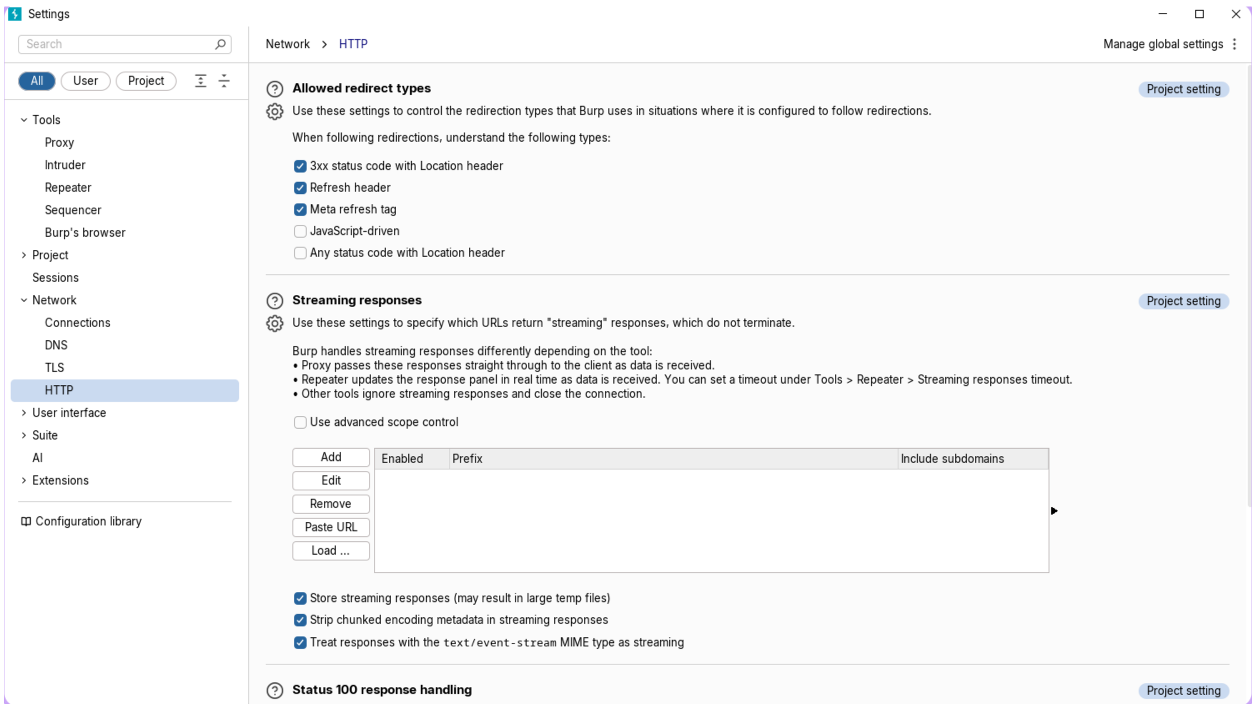
☐ Automatically fix missing or superfluous new lines at end of request
✓ Automatically update Content-Length header when the request is edited

Response interception rules

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

Checking Network → HTTP Settings

I also checked redirect settings and streaming responses.



Network > HTTP

Allowed redirect types

Use these settings to control the redirection types that Burp uses in situations where it is configured to follow redirections.

When following redirections, understand the following types:

✓ 3xx status code with Location header
✓ Refresh header
✓ Meta refresh tag
☐ JavaScript-driven
☐ Any status code with Location header

Streaming responses

Use these settings to specify which URLs return "streaming" responses, which do not terminate.

Burp handles streaming responses differently depending on the tool:
• Proxy passes these responses straight through to the client as data is received.
• Repeater updates the response panel in real time as data is received. You can set a timeout under Tools > Repeater > Streaming responses timeout.
• Other tools ignore streaming responses and close the connection.

☐ Use advanced scope control

Enabled	Prefix	Include subdomains

✓ Store streaming responses (may result in large temp files)
✓ Strip chunked encoding metadata in streaming responses
✓ Treat responses with the `text/event-stream` MIME type as streaming

Status 100 response handling

Conclusion

This lab helped me learn:

- How to configure Burp Proxy
- How to intercept web traffic
- How HTTP requests look inside Burp
- Why HTTPS interception requires CA certificate installation

This completes the simple version of the Burp Suite Proxy Lab Report.

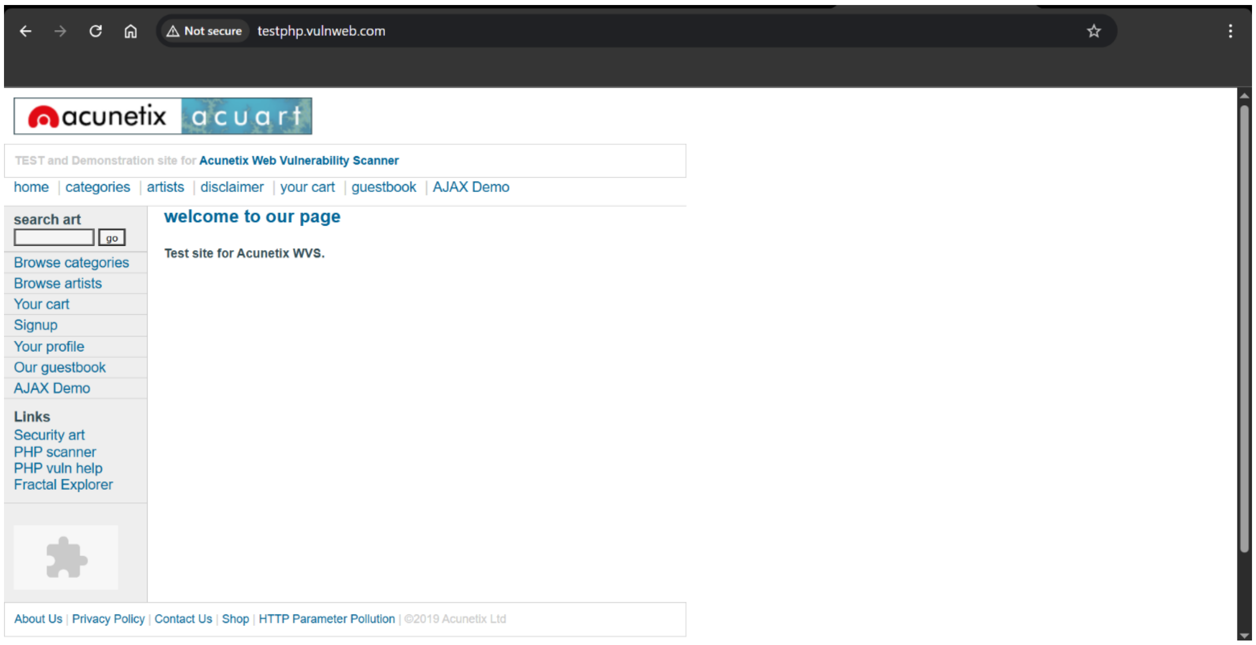
REPEATER LOGIN TESTING REPORT

1. Target Website Overview

The test was performed on the website:

- http://testphp.vulnweb.com/login.php

This page contains a login form with **username** and **password** fields.



TEST and Demonstration site for Acunetix Web Vulnerability Scanner

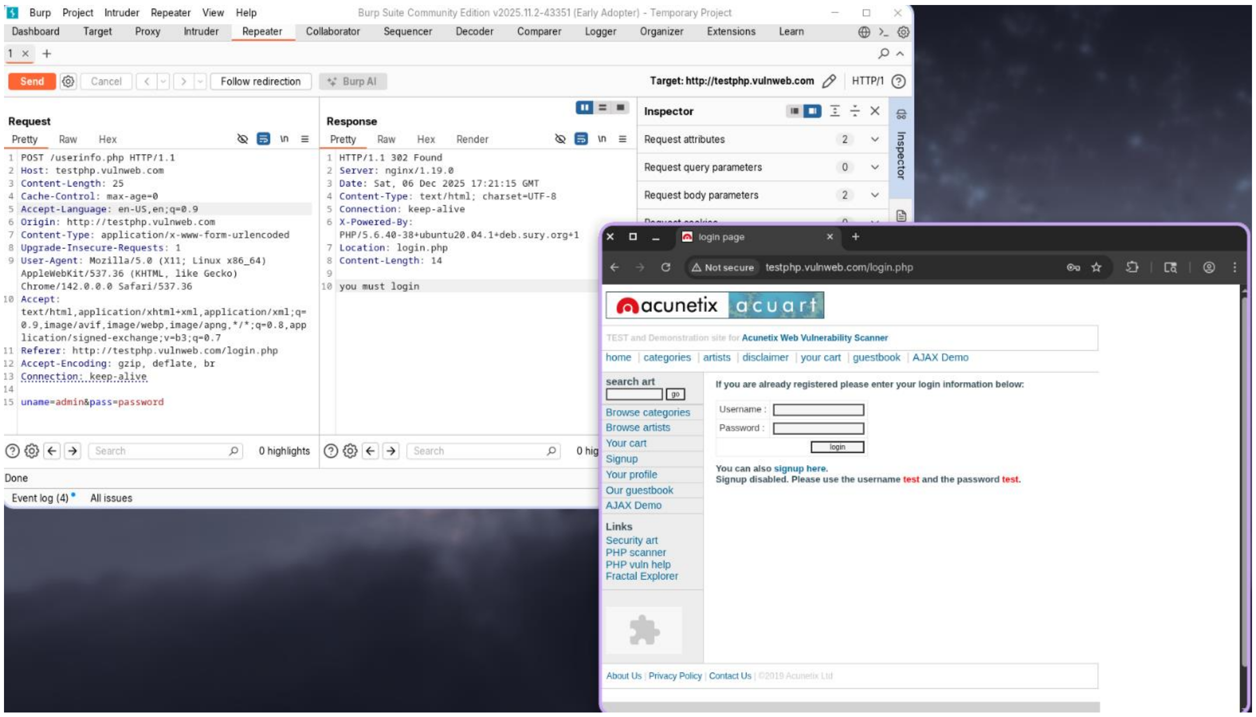
welcome to our page

Test site for Acunetix WVS.

2. Capturing the Login Request

Using Burp Proxy, the login request was intercepted when credentials were entered into the login page.

- Method: **POST**
- Endpoint: /userinfo.php
- Content-Type: application/x-www-form-urlencoded
- Body parameters:
  - Uname:admin
  - Pass:password


```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 25
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/142.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

uname=admin&pass=password
```

```
HTTP/1.1 302 Found
Server: nginx/1.19.0
Date: Sat, 06 Dec 2025 17:21:15 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By:
PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Location: login.php
Content-Length: 14

you must login
```

Target: http://testphp.vulnweb.com

If you are already registered please enter your login information below:

Username :
Password :

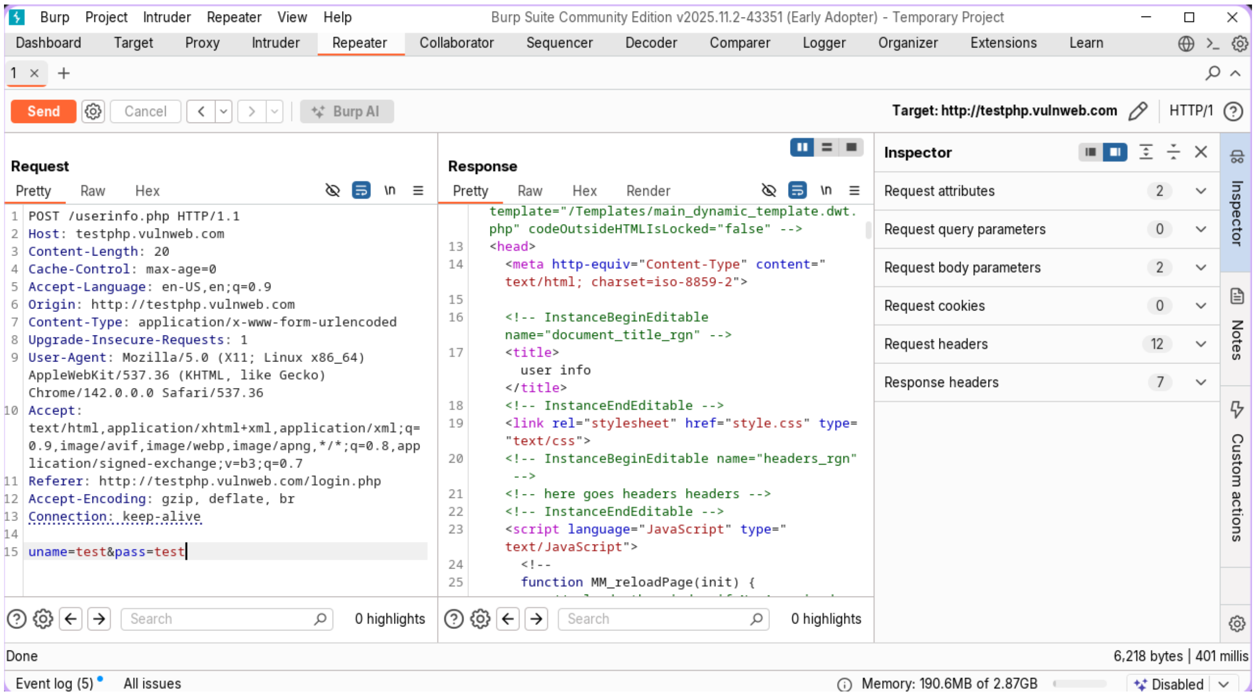
You can also signup here.
Signup disabled. Please use the username test and the password test.

3. Sending Request to Repeater

Once the request was intercepted, it was forwarded to **Repeater** for manual testing.

Steps followed:

- Right-click the request
- Click **Send to Repeater**
- Open the **Repeater** tab


Burp Suite Community Edition v2025.11.2-43351 (Early Adopter) - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Send Cancel Burp AI
Target: http://testphp.vulnweb.com HTTP/1
Request Pretty Raw Hex
Response Pretty Raw Hex Render
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 2
Request cookies 0
Request headers 12
Response headers 7
0 highlights
Done
6,218 bytes | 401 millis
Event log (5) All issues
Memory: 190.6MB of 2.87GB
Disabled

```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 20
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/142.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=
0.9,image/avif,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3;q=0.7
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

uname=test&pass=test
```

```
template="/Templates/main_dynamic_template.dwt.
php" codeOutsideHTMLIsLocked="false" -->
<head>
  <meta http-equiv="Content-Type" content="
  text/html; charset=iso-8859-2">

  <!-- InstanceBeginEditable
  name="document_title_rgn" -->
  <title>
    user info
  </title>
  <!-- InstanceEndEditable -->
  <link rel="stylesheet" href="style.css" type=
  "text/css">
  <!-- InstanceBeginEditable name="headers_rgn"
   -->
  <!-- here goes headers headers -->
  <!-- InstanceEndEditable -->
  <script language="JavaScript" type="
  text/JavaScript">
    <!--
    function MM_reloadPage(init) {
```

4. Testing Incorrect Credentials

First, wrong credentials were tested:

uname=admin&pass=password

Server Response:
you must login

This shows the login failed.

5. Testing Correct Credentials

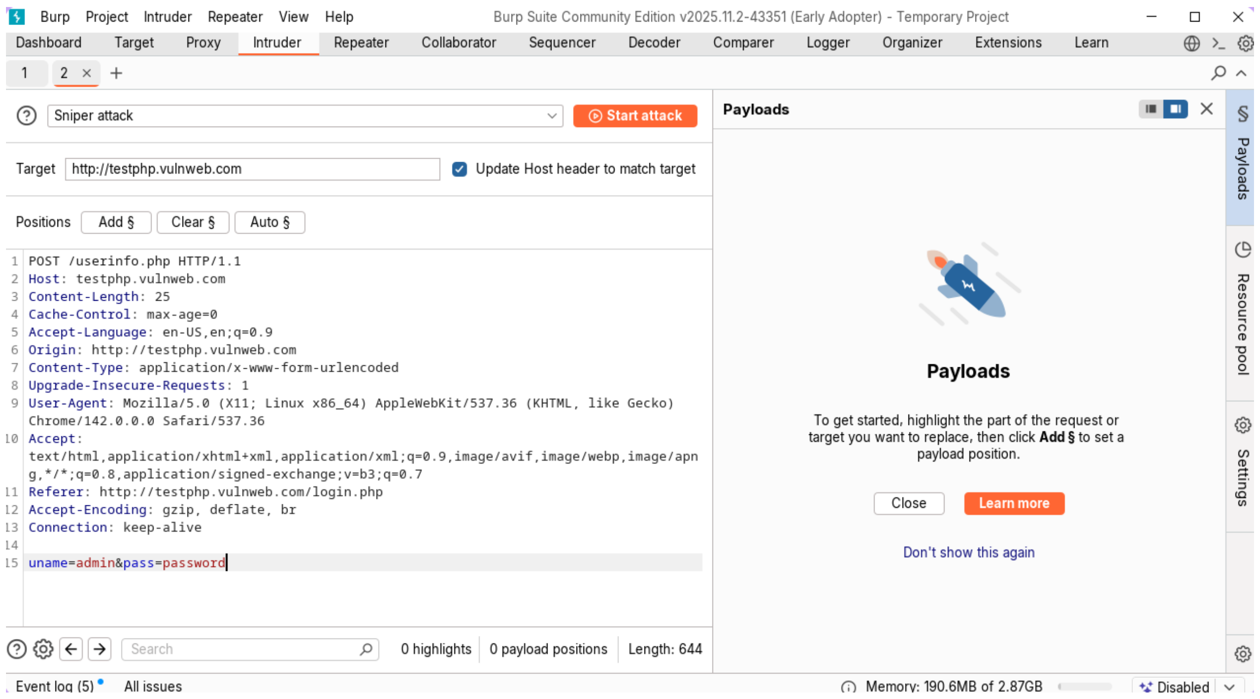
Next, valid test credentials were used:

uname=test&pass=test

The server responded with a full HTML page showing **user info**, meaning login was successful.

INTRUDER PRACTICAL

This section documents the workflow followed when conducting an Intruder attack in Burp Suite Community Edition. The three supplied screenshots illustrate the process from selecting a request in the Proxy HTTP history, through configuring Intruder positions and payloads, to executing the attack and reviewing the results. The following analysis outlines the purpose and function of each step, as well as its significance in a security-testing context.


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.11.2-43351 (Early Adopter) - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Sniper attack
Start attack
Target http://testphp.vulnweb.com
Update Host header to match target
Positions Add § Clear § Auto §
Payloads
To get started, highlight the part of the request or target you want to replace, then click Add § to set a payload position.
Close
Learn more
Don't show this again
Resource pool
Settings
0 highlights
0 payload positions
Length: 644
Event log (5)
All issues
Memory: 190.6MB of 2.87GB
Disabled

```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 25
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/142.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apn
g,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

uname=admin&pass=password
```

1. Selecting the Request from HTTP History and Sending It to Intruder

How We Reached This Step

The first screenshot shows a raw HTTP POST request captured by Burp Suite's Proxy tool as the user interacted with the target web application (testphp.vulnweb.com). The tester selected this request from the HTTP history tab and chose the option "Send to Intruder", which automatically opened the Intruder interface and populated it with this request.

What Is Being Done

Burp Intruder replicates this request, allowing specific parts of the request to be parameterized for automated manipulation. This step captures all request details—headers, content length, cookies (if present), and payload fields.

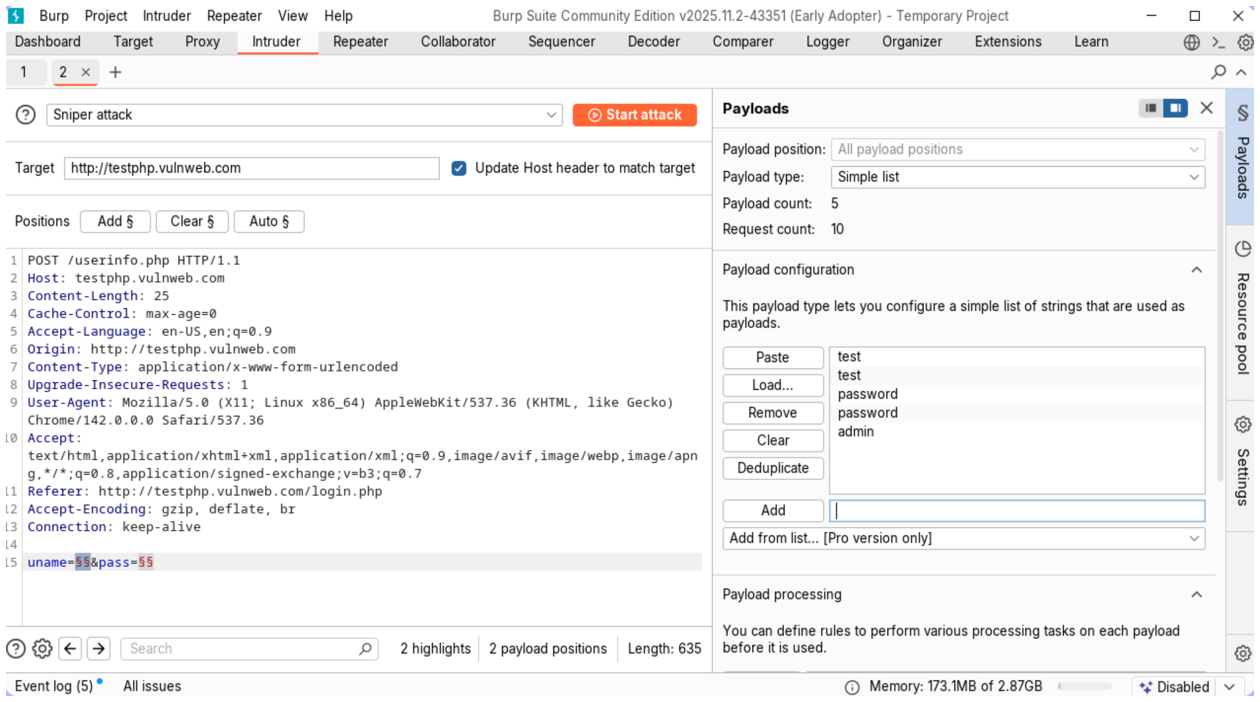
Why This Action Is Performed

Sending a request from Proxy to Intruder enables controlled, automated testing of user-supplied inputs. In this case, the login form parameters (uname and pass) are suitable for:

- brute-force testing

- credential enumeration
- parameter fuzzing
- behavioral analysis of authentication controls

This step forms the foundation of any Intruder-based assessment because Intruder requires a concrete request template to repeatedly manipulate and replay.


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.11.2-43351 (Early Adopter) - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Start attack
Positions Add § Clear § Auto §
Paste Load... Remove Clear Deduplicate Add
Payloads Resource pool Settings
Event log (5) All issues
Memory: 173.1MB of 2.87GB
Disabled

Sniper attack

Target: http://testphp.vulnweb.com ✓ Update Host header to match target

```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Length: 25
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/142.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apn
g,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

uname=§§&pass=§§
```

2 highlights | 2 payload positions | Length: 635

Payloads

Payload position: All payload positions
Payload type: Simple list
Payload count: 5
Request count: 10

Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

```
test
test
password
password
admin
```

Add from list... [Pro version only]

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Defining Intruder Positions and Configuring Payloads

How We Reached This Step

In the Intruder interface, the tester selected the “Positions” tab shown in the second screenshot. Here the tester manually highlighted the values for the fields pass and marked them as payload positions by clicking “Add §”, resulting in:
uname=test&pass=§

Additionally, the right-pane shows the Payloads tab where several sample payloads were configured.

What Is Being Done

<u>Two key configurations occur here:</u>

A. Selecting Payload Positions

The § markers indicate the input parameters Burp Intruder will replace with payload values. In Sniper attack mode, Burp tests one position at a time while holding the other constant. This is appropriate for single-parameter fuzzing or for assessing differences in server behavior.

B. Configuring Payloads

The tester used a Simple List payload type and added the following values:

- test
- password
- admin

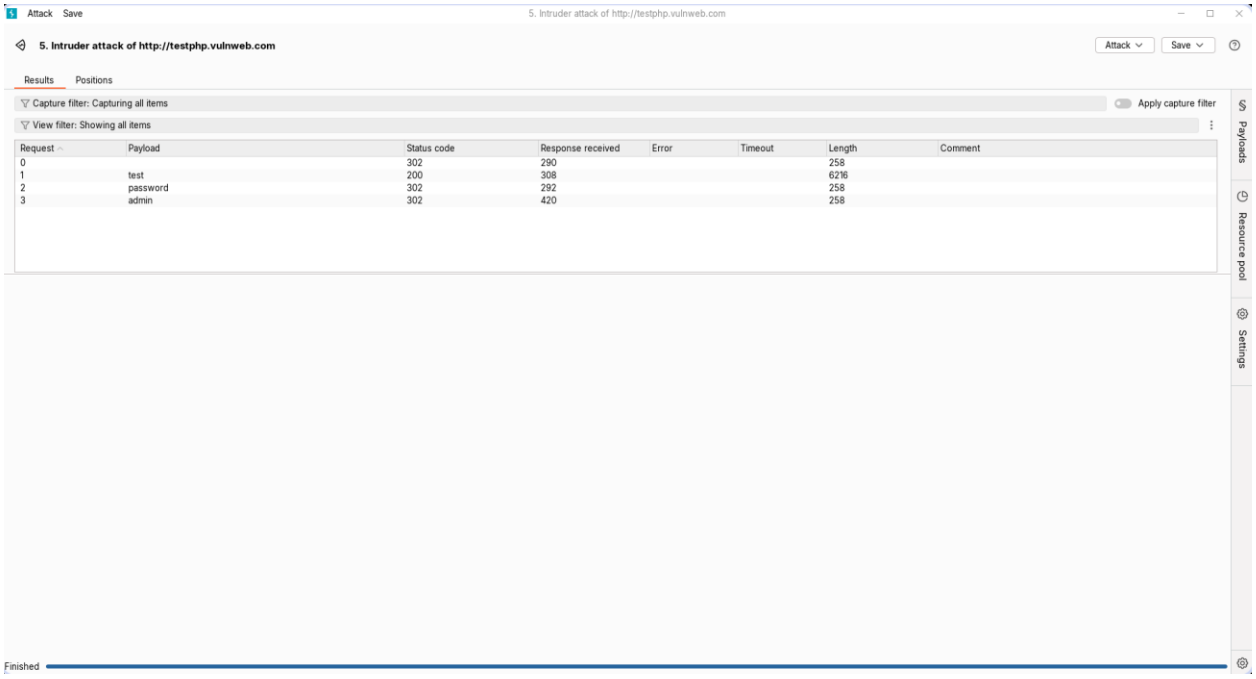
Each value will be substituted into the target position during the automated attack.

Why These Actions Are Performed

This step defines what values should be tested and where those values will be injected. Position configuration is critical because:

- It allows systematic enumeration of input values.
- It ensures the request structure remains intact except for the variable under test.
- The payload list lets the tester model realistic credentials or fuzzing strings.
- Proper configuration ensures repeatable and measurable attack execution.

In a credential-testing scenario, this step enables Burp Intruder to test multiple username/password combinations for authentication weaknesses or valid-account discovery.



5. Intruder attack of http://testphp.vulnweb.com

Capture filter: Capturing all items

View filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		302	290			258	
1	test	200	308			6216	
2	password	302	292			258	
3	admin	302	420			258	

Executing the Intruder Attack and Reviewing the Results

How We Reached This Step

After configuring positions and payloads, the tester clicked “Start attack”, which triggered the automated request sequence. The third screenshot displays the Intruder Attack Results window, listing each automated request and its corresponding server response.

What Is Being Done

For each payload value, Intruder:

- Injects the payload into the marked position(s).
- Sends the modified request to the target.
- Records the following information:
- Status code
- Response length
- Response time
- Any errors or anomalies

The results table shows each attempt (e.g., requests using “test,” “password,” “admin”) and the server’s responses (primarily HTTP 302 redirect codes).

Why These Actions Are Performed

The attack results are used to:

- Identify valid or invalid credentials by analyzing patterns in status codes or response sizes.
- Detect behavioral inconsistencies that might suggest authentication bypass.
- Evaluate rate-limiting or brute-force protections.
- Highlight potential flaws in the login mechanism.
- For example, a differing response length or status code often indicates a successful login or unique server behavior tied to a specific payload.

Conclusion

The section collectively illustrates a complete Burp Suite Intruder workflow:

- Capturing a relevant authentication request in Proxy and sending it to Intruder.
- Defining fuzzing or brute-force parameters by marking Intruder positions and supplying payload lists.
- Executing an automated attack and analyzing server responses to determine authentication behavior or detect vulnerabilities.

This workflow is fundamental in web-application security assessments where understanding server responses to crafted inputs is essential for identifying authentication weaknesses, misconfigurations, or potential exploitation vectors.