browsertotal™

Free Service by SERAPHIC

BrowserTotal™ Browser Security Assessment for Enterprises

The AI-Driven Platform for
Auditing Enterprise Browser Security

USER GUIDE

JUNE 2025

Table of Contents

Introduction

The Seraphic **BrowserTotal**™ Platform is a browser-native security assessment environment purpose-built to demonstrate all browser risks, and allow users to assess risk of browser extensions and URLs with zero effort.

BrowserTotal™ is built to:

- **Assess Browser Security Posture**: The platform provides detailed insights into browser configurations, vulnerabilities, and potential exposures, helping organizations identify security gaps and misconfigurations.
- **Demonstrate Emerging Threats and Attacks**: **BrowserTotal**™ simulates real-world browser-based attacks and emerging threats, making it easier for security teams to visualize how cybercriminals exploit browsers and browser extensions.
- **Analyze URLs and Extensions**: The tool examines the safety and behavior of URLs and browser extensions in real time, identifying malicious activity, risky permissions, and suspicious web destinations.

By offering hands-on, practical assessments and threat demonstrations, **BrowserTotal**™ equips security teams and decision-makers with actionable intelligence to improve their defenses. The platform is purpose-built to raise awareness, validate security controls, and empower organizations to proactively address browser-related risks in the evolving threat landscape.

The Platform delivers a suite of modular, self-contained capabilities that simulate real-world attack scenarios, including phishing, data leakage, risky extensions, and unauthorized access across both managed and unmanaged devices. Each module runs autonomously within the browser environment, enabling real-time threat simulations, localized analysis and insights, as well as actionable best-practice guidance. The result is a seamless and consistent user experience that works across diverse devices and deployment models.

With **BrowserTotal**™, users gain hands-on exposure to key risks and defenses, offering both education and validation of browser-native protection strategies with practical insights in a safe, controlled environment.

Browser**Total** Feature Summary Table

Test Module	Description	Key Benefits / Outcomes
Browser Posture	Automatically evaluates the current security configuration of the browser	Provides a baseline risk score and identifies gaps in browser security posture
Extensions	Analyzes browser extensions and other platform plugins for potential security risks and vulnerabilities	Offers visibility into any potential security or privacy risk associated with an extension.
Analysis	Evaluates any given URL for potential risk associated with the site as well as its privacy policy	Demonstrates the risk that the site is posing to a user.
Emerging Threats	Simulates advanced malware campaigns like SocGholish and Clear Fake	Validates browser's effectiveness against sophisticated and evolving web-based threats
Identity Attacks	Emulates phishing techniques that target token theft, session hijacking and other impersonation scenarios	Highlights protections for user identity and authentication integrity
Client-Side Attacks	Demonstrates attacks such as reflected XSS and clickjacking	Showcases in-browser defenses against manipulation and injection-based threats, allows for educating users on such risks
Downloads	Simulates malicious file downloads via Blob URLs, Data URLs or steganography	Reinforces download inspection and prevention within the browser
Data Loss Prevention	Simulates sensitive data exposure or exfiltration via forms, clipboards and more	Demonstrates policy enforcement to protect sensitive data in real-time within the browser
Content Filtering	Blocks access to defined categories of restricted or non-compliant websites	Provides flexible, policy-driven access controls based on user roles or device posture
Reconnaissance	Simulates attacker reconnaissance techniques like fingerprinting and header scans	Demonstrates how to detect and stop early-stage recon tactics used in complex attacks
Sandbox	Offers a safe environment to interact with and analyze suspicious URLs	Enables secure exploration of risky and other unknown content without exposing the local browser or system

browsertotal.com

Browser Posture

Description

This browser simulation environment automatically evaluates the browser's posture. It runs over 120 tests and detects key signals like browser type and version, extension activity, OS-level indicators, and security hardening status. This assessment allows users to identify security gaps in their browser and outlines what steps are needed to close them.

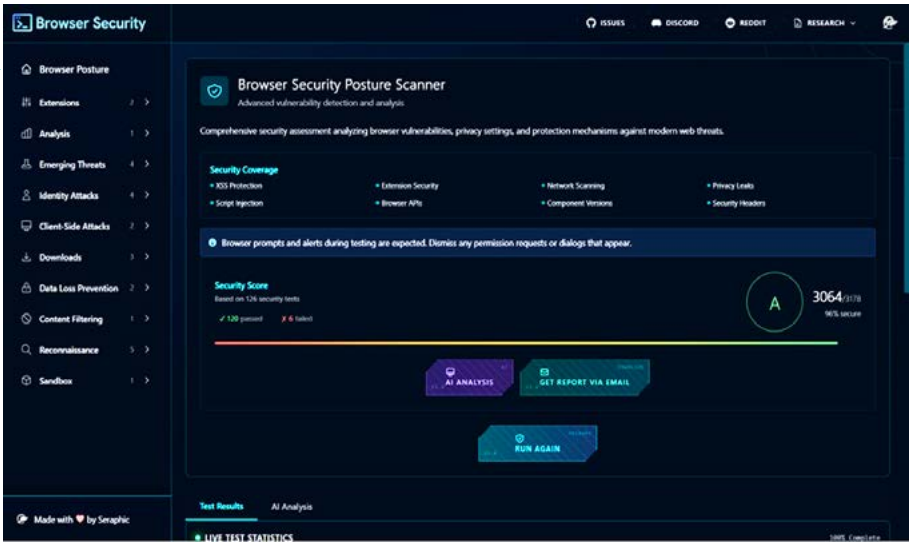


Figure 1: *Browser Security Posture Scanner displays a detailed assessment of browser configurations, providing a baseline risk score while highlighting security gaps and risks*

Instructions

1. Launch the Simulation Lab and select “Browser Posture”.
2. Click “Evaluate Your Browser Posture”.
3. Click “Start Security Scan”
4. A pop-up window will appear alerting you to the nature of the test and provides you with pre-scan recommendations. Click “Proceed with Scan” to begin the scan.

browsertotal.com

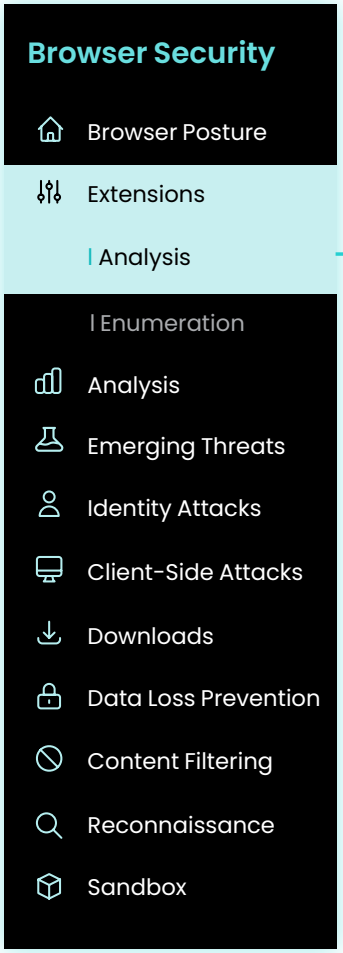
5. Observe the tests and results response in each scenario – see failure and success. **Note**: It is highly recommended to enforce restricting policies on the browser to fully evaluate its posture (for example, apply content filtering, update your browser version, etc.)
6. See summary of scan – how many tests failed, which tests failed, and what is the overall scoring of the test.
7. Review the logs to understand what failed and what should be done to fix.

Highlights

- No agent installation required—works directly in the browser.
- Instant posture assessment of all risks and gaps in the browsers.
- Can detect misconfigured or high-risk environments (e.g., Incognito mode, DevTools open, tampered extensions).
- Highlights the gaps and helps resolve them (and ensures they are indeed solved).

Can the results be altered?

Since the framework runs inside the browser, we took care to consider that malware or harmful extensions might detect or interfere with it. If a user's browser is already compromised—for example, by a malicious extension or modified browser code—a skilled attacker could potentially change how the testing script runs or what results it produces. To protect against this, our framework includes self-checks (such as comparing the script to a known hash, if available) and uses several methods to spot any kind of external tampering attempt.



Extensions Analysis

Description

This simulation showcases Seraphic's ability to analyze and categorize installed browser extensions in real time (as well as JetBrains add-ons, VS Code extensions and NPM JS Packages). By leveraging a combination of store metadata, static and dynamic analysis, and AI-driven risk scoring, Seraphic helps identify extensions that may pose a privacy, or security risk—even before they are actively used.

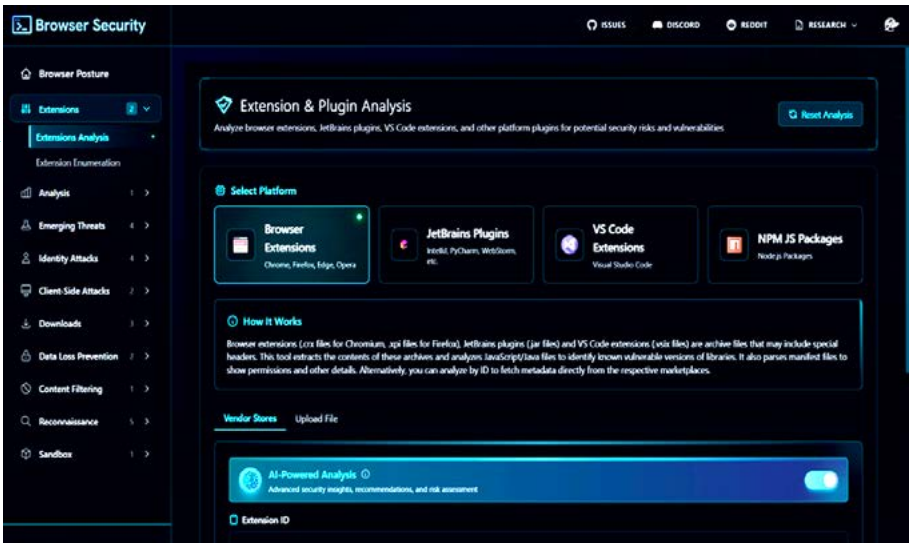
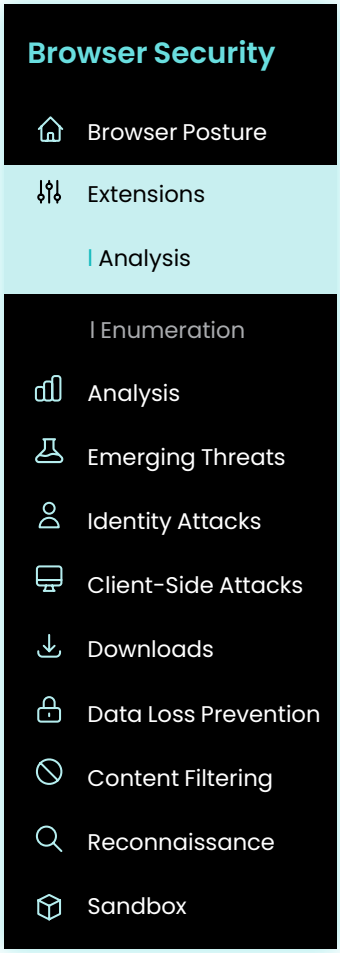


Figure 2: *Extensions Module identifies installed extension and flags potentially risky or unauthorized ones, offering visibility into extension-based risks across managed and BYOD devices.*

Instructions

1. Launch the Simulation Lab and navigate to **Extensions, Extensions Analysis.**
2. Select the desired Platform then choose the relevant Vendor Store
3. Search for the Extension – searching by name or extension ID/URL
4. Click Analyze Extension
5. Seraphic automatically analyzes each extension using parameters such as:



browsertotal.com

a. Extension store reputation (rating, number of installs)

b. Publisher authenticity

c. Requested permissions

d. Behavioral analysis

e. AI-based categorization

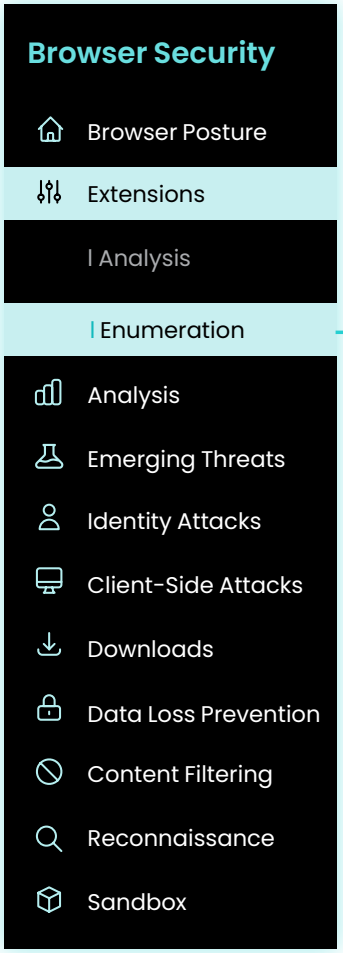
6. Observe how the system assigns a risk score and categorizes each extension (e.g., Security, Productivity, Privacy Risk, AI tool, etc.).
7. Use the control panel to simulate installation of additional extensions and review how Seraphic responds—e.g., block, alert, or allow.
8. **Note**: In some cases, AI analysis results might be different from the results found by static/dynamic analysis, this is acceptable as it adds additional context into the analysis that may have impact on the results.

Highlights

- Real-time extension discovery and analysis—no user input required.
- AI classification engine that continuously improves based on global and local data.
- Supports multiple browsers including Chrome, Edge, Safari, and Firefox.
- Can analyze sideloaded or enterprise-distributed extensions that don't appear in the store.

Risk of malicious extensions

Browser extensions often use APIs to inject scripts into pages, attaching event listeners or keeping copies of DOM elements. For example, an ad-blocker might track removed ads to prevent them from reappearing. Similarly, a malicious extension—or a script injected via a compromised CDN or man-in-the-middle attack—could wrap DOM APIs or access sensitive form fields to steal data. This behavior is usually invisible to the page or external monitors. With new malicious extensions discovered almost daily, such risks are an increasing concern.



browsertotal.com

Extensions Enumeration

Description

This module attempts to detect all browser extensions installed or active in the end-user's browser session, regardless of whether the extension is visible to the user. The module is designed to support security teams in identifying risky or unauthorized extensions that could compromise data integrity, privacy, or compliance. The detection mechanism uses JavaScript-based fingerprinting and runtime analysis techniques to bypass traditional visibility limitations, including extensions running in incognito or with hidden permissions.

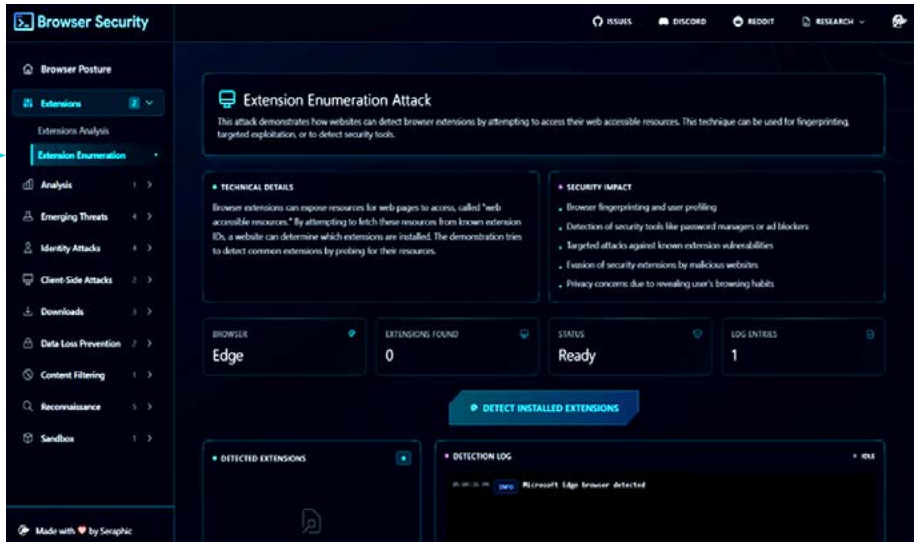


***Figure 3**: Extension Module, Extension Enumeration detects all browser extensions installed or active, regardless of whether the extension is visible to the user, to identify risky or unauthorized extensions.*

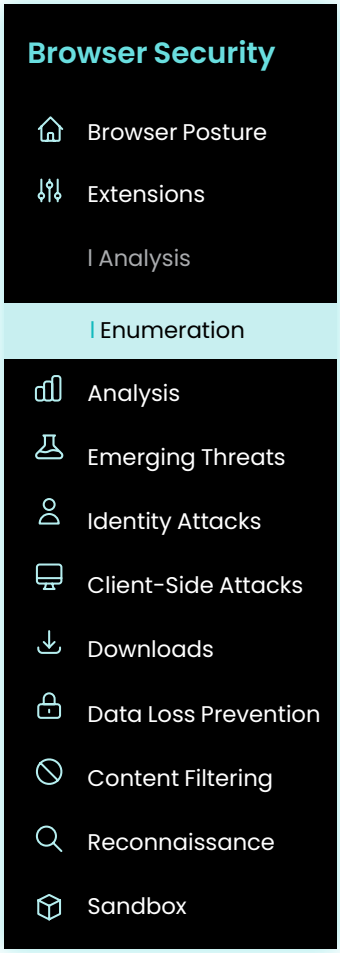
Instructions

1. **Activation:**

   a. Click "Detect Installed Extensions" button.

   b. It runs silently during the session and does not require user interaction.

   c. If you have Seraphic, ensure the Extension Enumeration module is enabled to test its ability to protect against extension enumeration.



browsertotal.com

2. Output:

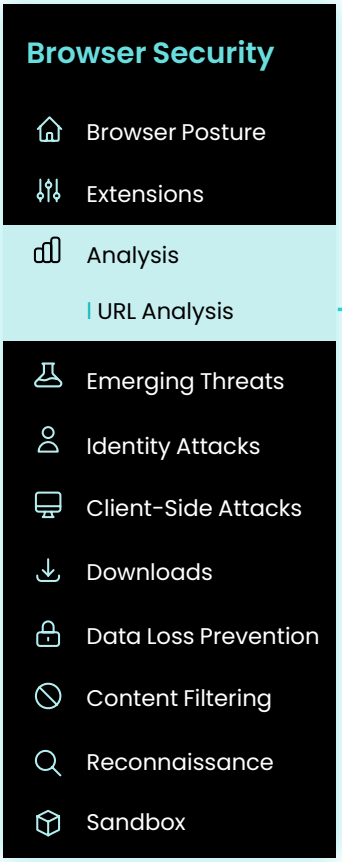
a. If found, results are presented on the page.

Highlights

- **Privacy-Aware Scanning**: No PII is collected; only extension metadata.
- **Zero-Install Detection**: Operates without installing new extensions or requiring elevated browser permissions.

Risk of extension enumeration

Extension enumeration relies on detecting unique markers left by browser extensions, like DOM elements or resource paths. While sometimes used for harmless checks, attackers can exploit it to identify security tools or user behavior to adapt their tactics. As new malicious extensions appear almost daily, this technique poses a growing threat in browser-based environments.



URL Analysis

Description

The URL Analysis module inspects URLs specified by the user. This analysis is executed entirely within the browser. This module helps detect risky, malicious, or non-compliant destinations, as well as vulnerable resources, different components, modules and technologies used by the website, sandbox to interact with the website, certificates, code, static analysis and AI-based analysis.

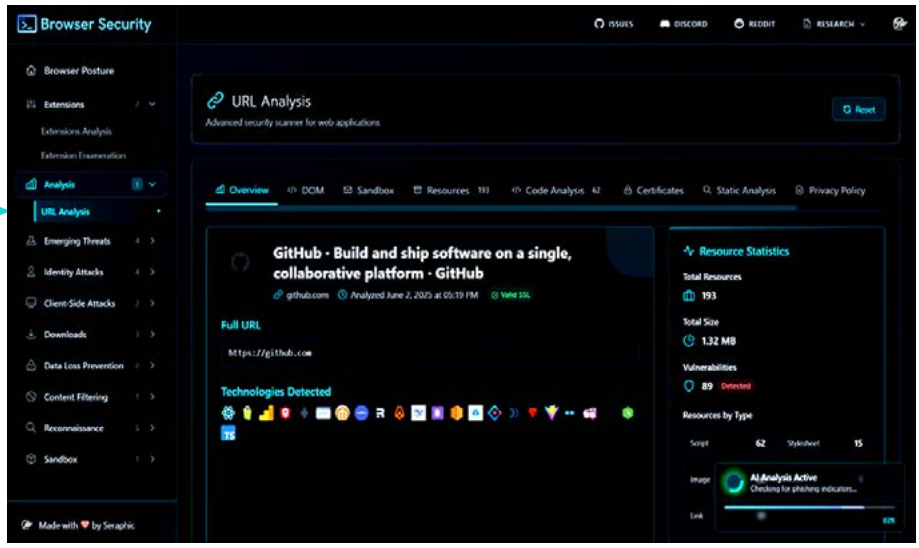
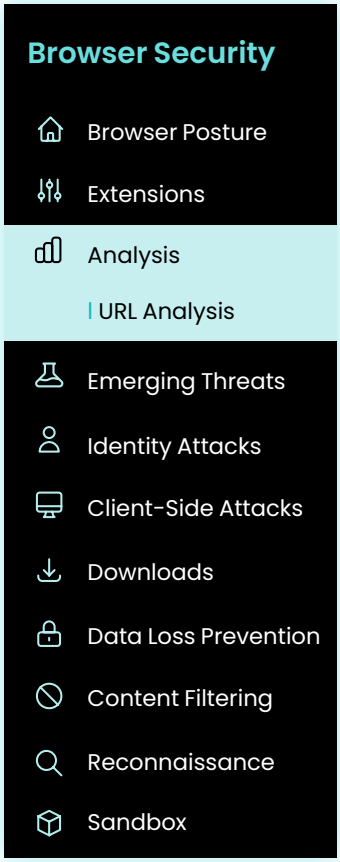


***Figure 4**: Analysis Module, URL Analysis demonstrates how the browser responds to suspicious or malicious URLs, highlighting real-time protection against drive-by threats and redirection attacks.*

Instructions

1. **Activation:**
   a. This module is embedded and runs by default as part of the core browser agent.
   b. Users input the website to analyze within the URL analysis page.
   c. Click "Analyze webpage"



browsertotal.com

2. Output:

a. High-level overview of the websites, Domain, Ips, DNS records,

b. DOM analysis, forms, links and iframes.

c. Sandbox interaction with the website

d. Resources – JavaScript, CSS, Images, Documents

e. Code analysis of JS files on the site

f. SSL/TLS Certificates

g. Static analysis

h. Vulnerabilities

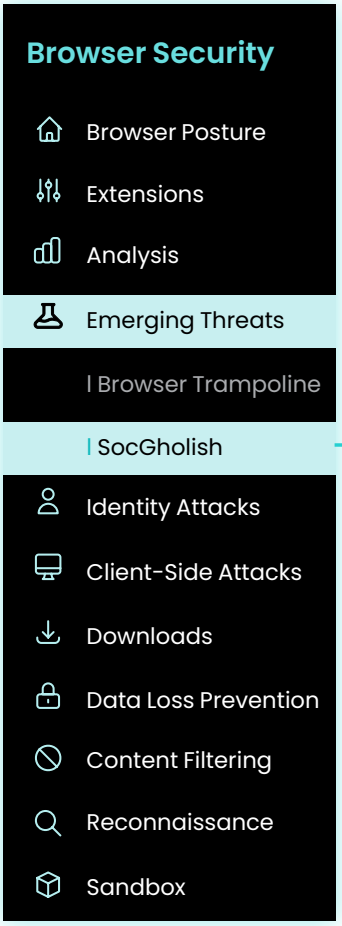
i. AI analysis – high level trust score/website risk, trust indicators, vulnerabilities, recommendations, privacy risk assessment

Highlights

- **Real-Time, Local Execution**: Analysis and enforcement happen entirely within the browser — no roundtrips, no dependencies, including AI-based analysis
- **Comprehensive Visibility**: Explore all website components including vulnerabilities, certificates, scripts, stylesheets, etc.

Bad websites are YOUR problem

Visiting websites with vulnerable resources, outdated modules, misconfigured components, or invalid certificates can expose users to significant risks. Attackers can exploit these weaknesses to inject malicious code, perform drive-by downloads, or launch phishing and man-in-the-middle attacks. Even passive browsing can lead to malware infection or data leakage if the site loads compromised third-party scripts or insecure assets. Such vulnerabilities can also enable user tracking, session hijacking, or unauthorized access to sensitive information, making them a serious threat to both privacy and security.



Emerging Threats

Description

This module simulates how new and emerging threats like Browser Trampoline, SocGholish (FakeUpdates), and Clear Fake as well as the latest Common Vulnerabilities and Exposures (CVEs) are used to target browsers and trick users into downloading malicious software disguised as legitimate browser updates. It demonstrates how to detect potentially risky, malicious, or non-compliant code attempting to be injected into the browsers JavaScript Engine (JSE) for the purpose of contacting a remote server for additional malware downloads.

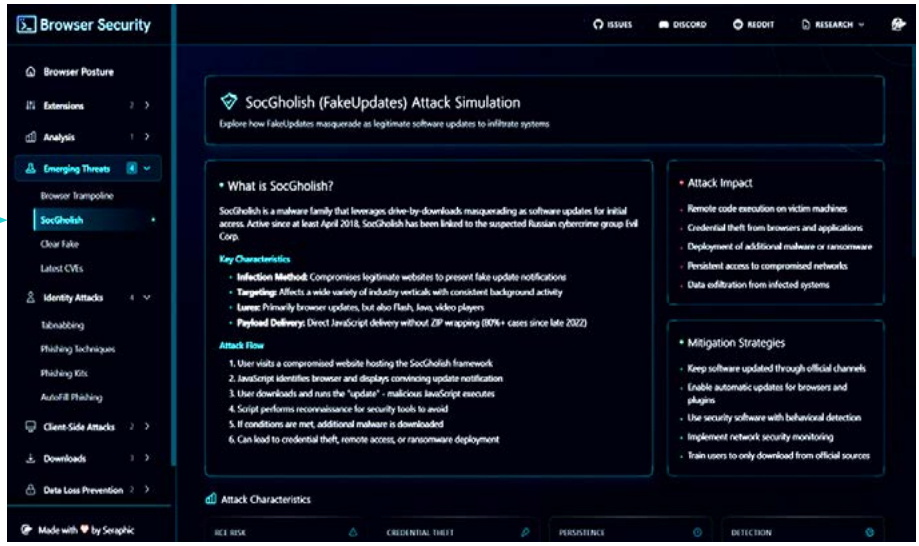


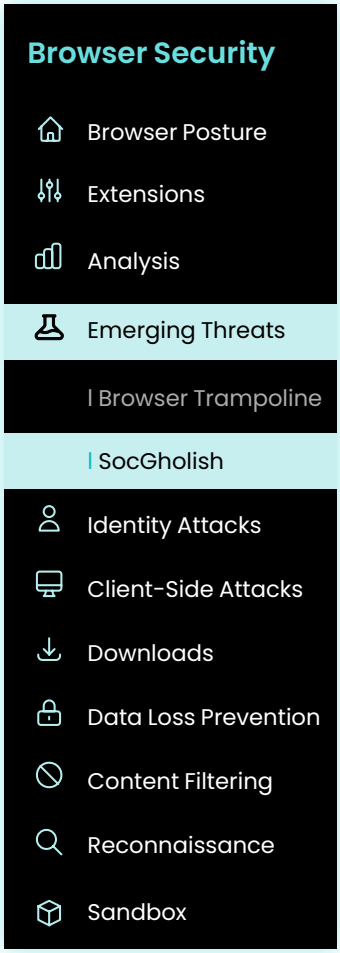
Figure 5: Emerging Threats Module simulates high-impact malware campaigns like SocGholish to test browser resilience against evolving web-based attack methods.

Instructions

1. **Activation:**

   a. This module is embedded and runs as part of the browser simulation environment.

   b. Users select which threat they would like to simulate:



browsertotal.com

i. Browser Trampoline – user selects the desired language then either copies or downloads the Browser Trampoline code, then executes it. User then runs "Execute" from the **BrowserTotal** console.

ii. SocGholish and Clear Fake – user simply runs the attack simulation – follow the steps detailed to see how potentially malicious files are being pushed down to the device through the browser.

iii. CVEs – provides real-time monitoring of current vulnerabilities affecting all major browsers.

Highlights

- **Real-Time, Local Execution**: Attack simulations in a secure, isolated environment with no malware being downloaded to your endpoint.
- **Comprehensive Visibility**: Detailed analysis and explanation of each attack, including how they operate and how to recognize them in the wild.
- **Best Practices Guidance**: Tips on how best to mitigate these and other emerging threats
- **Educate**: Comprehensive list of CVEs to help security teams stay informed and ensure timely patching to protect their users.

Malware infection via legitimate sites

New threats emerge every day, and one particularly dangerous example is the SocGholish malware. Delivered through compromised websites, SocGholish disguises itself as a browser update or a legitimate-looking prompt to trick users into downloading malware. Once installed, it can provide attackers with remote access, steal credentials, or drop additional payloads such as ransomware. What makes SocGholish especially dangerous is its use of trusted but vulnerable websites to deliver its payload, often bypassing traditional security filters. This highlights the growing risk of socially engineered attacks and the importance of securing both user behavior and the websites they visit.

Browser Security
Browser Posture
Extensions
Analysis
Emerging Threats
Identity Attacks
| Tabnabbing
| Phishing Techniques
| Phishing Kits
| AutoFill Phishing
Client-Side Attacks
Downloads
Data Loss Prevention
Content Filtering
Reconnaissance
Sandbox
browsertotal.com

Identity Attacks

Description

This module walks users through real-world examples of identity-based attacks – including Tabnabbing, Phishing Kits, AutoFill Phishing, and other deceptive techniques designed to steal credentials or impersonate trusted users. The simulation demonstrates how attackers exploit familiar user behaviors, manipulate the browser interface, and hijack trust to capture sensitive information.

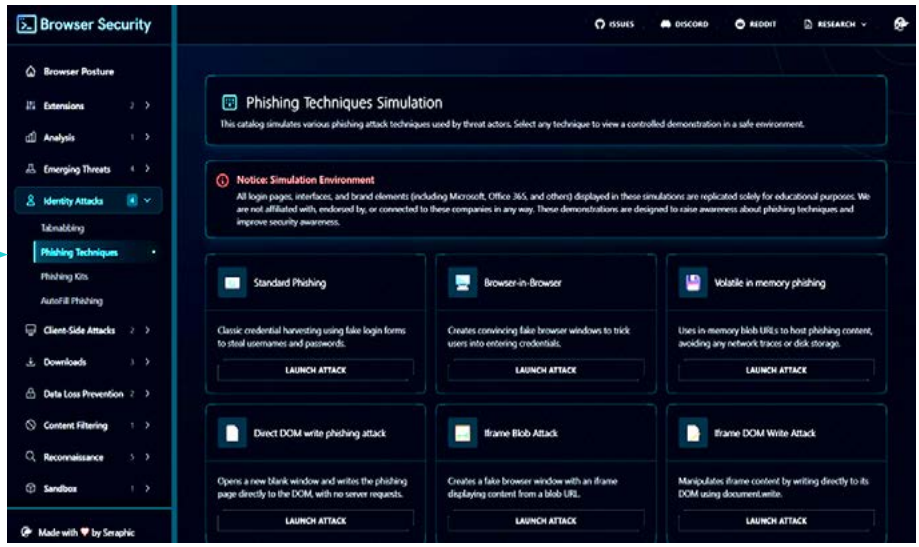
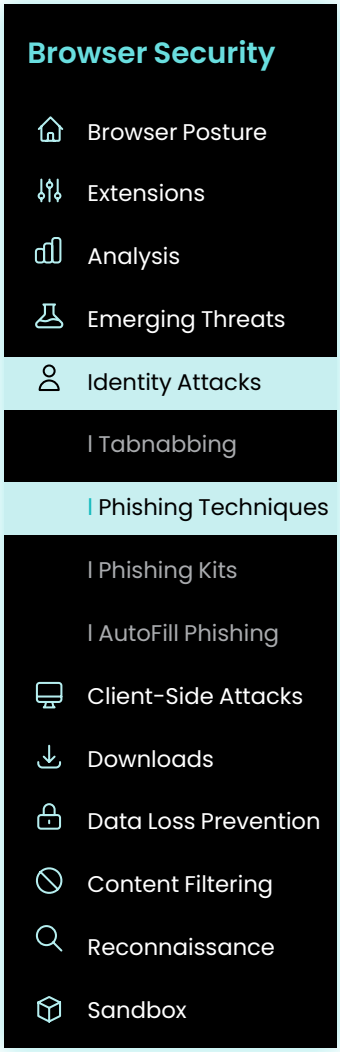


***Figure 6**: Identity Attacks Module illustrates common scenarios such as token theft, session hijacking, and impersonation to show how to protect user identities within the browser.*

Instructions

1. **Activation:**
   a. This module is embedded and runs as part of the browser simulation environment.
   b. Users select which Identity Threat they would like to simulate:
      i. Tabnabbing – user clicks the "Start Tabnabbing Simulation", follow screen instructions.



browsertotal.com

ii. Phishing Techniques – user selects the desired simulated attack (standard phishing, browser-in-browser, Volatile in Memory Phishing, Direct DOM write Phishing, Iframe Blob Attack, or Iframe DOM Write Attack), and clicks "Launch Attack"

iii. Phishing Kits – user selects which kit to run in a sandbox environment and clicks "Initiate Sandbox"

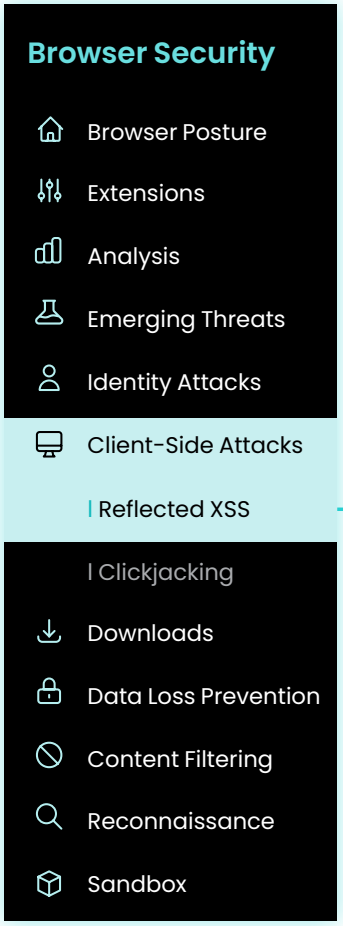
iv. AutoFill Phishing – user simply clicks "Launch Demo" to run this identity attack simulation. Use browser autofill to check if it exposes sensitive/personal data that was not provided. Note: It could be that autofill doesn't have sensitive data such that it will not be presented in the demo.

Highlights

- **Real-Time, Local Execution**: Attack simulations in a secure, isolated environment with no malware being downloaded to your endpoint.
- **Comprehensive Visibility**: Detailed analysis and explanation of each attack, including how they operate and how to recognize them in the wild.
- **Best Practices Guidance**: Tips on how best to mitigate these and other emerging threats, including security tips on best ways to avoid falling victim to these attacks.

Phishing by AI

Emerging phishing kits are becoming increasingly sophisticated, especially with the integration of AI, making phishing campaigns more convincing and harder to detect. These AI-powered tools can generate realistic login pages in seconds while dynamically adapting content based on the target's profile, increasing the chances of success. This evolution drastically lowers the barrier for attackers and amplifies the scale and effectiveness of phishing attacks, posing serious threats to both individuals and organizations in terms of data theft, credential compromise, and financial loss.



Client-Side Attacks

Description

This section demonstrates how client-side browser attacks like Reflected Cross-Site Scripting (XSS) and Clickjacking can silently compromise end user sessions, hijack inputs, or trick users into unintended actions – all without breaching the underlying web app. Through interactive simulations, users learn how attackers inject malicious scripts, manipulate the browser DOM, or invisibly overlay UI elements to steal data or hijack workflows.

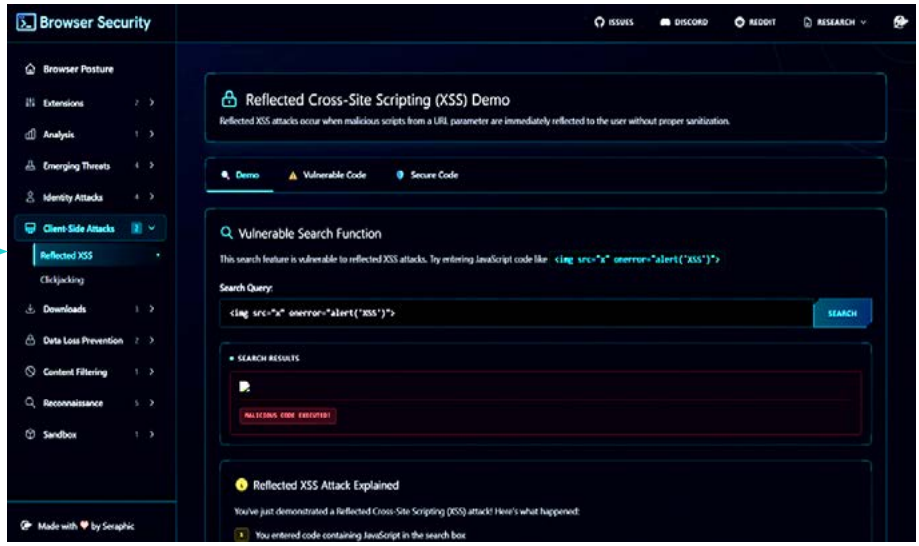
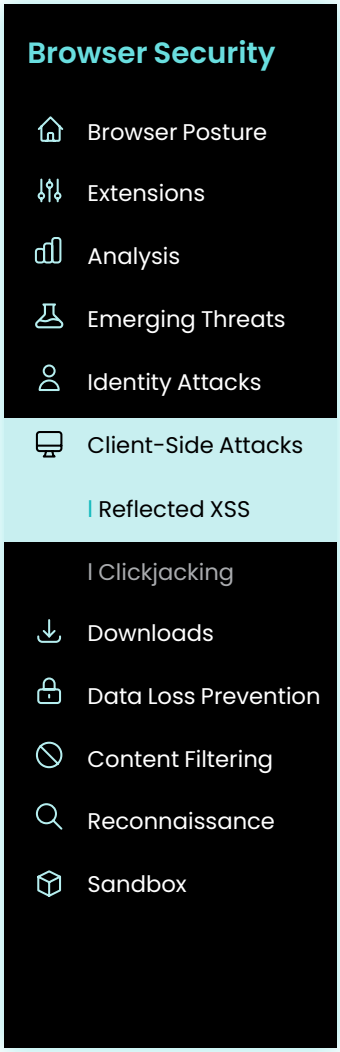


Figure 7: Client-Side Attack Module demonstrates protection against reflected XSS, clickjacking and other in-browser manipulation techniques that can compromise data or sessions.

Instructions

1. **Activation:**
   a. This module is embedded and runs as part of the browser simulation environment.
   b. Users select which threat they would like to simulate:
      i. Reflected XSS – this scenario simulates a vulnerable search function by allowing users to enter JavaScript code directly into a search query. Enter JavaScript code like "<img src="x" onerror="alert('XSS')">" to demonstrate the vulnerability.



browsertotal.com

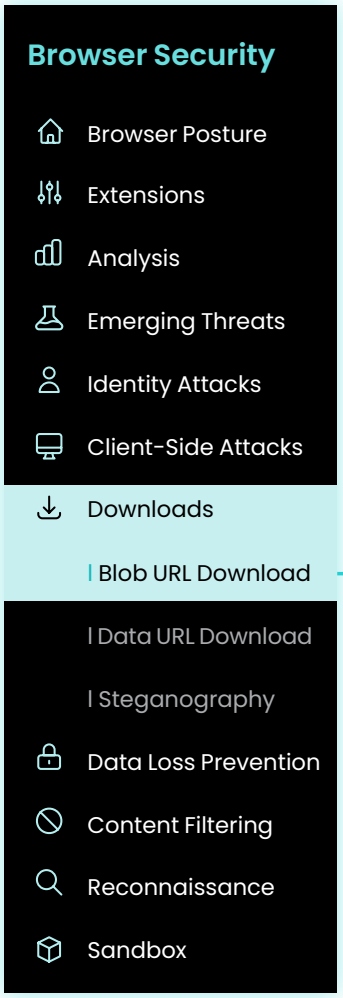
ii. Clickjacking – this scenario invites users to enter a URL and click "Test for Clickjacking Vulnerability" to see if a site is vulnerable to clickjacking. If you can see the page being loaded and fully presented, it is most likely that it is vulnerable to clickjacking attacks.

Highlights

- **Real-Time, Local Execution**: Attack simulations in a secure, isolated environment with no malware being downloaded to your endpoint.
- **Comprehensive Visibility**: Detailed analysis and explanation of each attack, including examples of vulnerable versus secure code, explaining the importance of sanitizing user input before rendering it to the DOM.
- **Best Practices Guidance**: Tips on how best to recognize and mitigate these and other emerging threats, including security tips on best ways to avoid falling victim to these attacks.

XSS and Clickjacking steal your identity

Client-side attacks like Cross-Site Scripting (XSS) and clickjacking pose serious security risks by exploiting how users interact with web pages. XSS allows attackers to inject malicious scripts into trusted websites, enabling them to steal session tokens, capture keystrokes, or redirect users to harmful sites. Clickjacking tricks users into clicking hidden or deceptive elements, potentially leading to unauthorized actions like changing settings or sharing sensitive information. These attacks are particularly dangerous because they occur within the user's browser, often without any visible signs, making them hard to detect and capable of compromising both user data and application integrity.



Downloads

Description

This module explores how seemingly benign downloads can be weaponized to deliver malware, steal data, and bypass traditional defenses all through the browser. In addition, it will also highlight the sophistication of modern browser-based delivery techniques that easily avoid detection by traditional security methods, such as CASB, SWG and SASE/SSE solutions as well as network scanning tools, highlighting why in-browser defenses are critical for stopping today's stealthiest attacks.

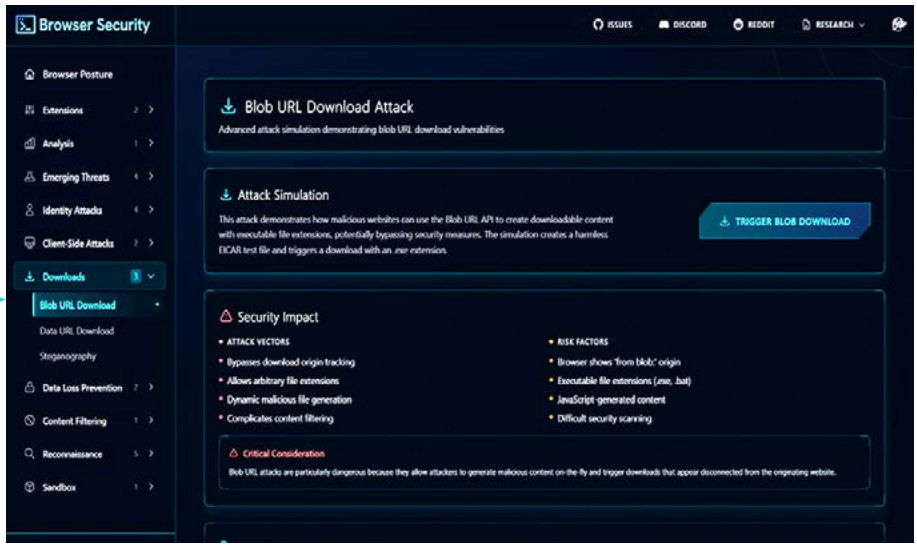
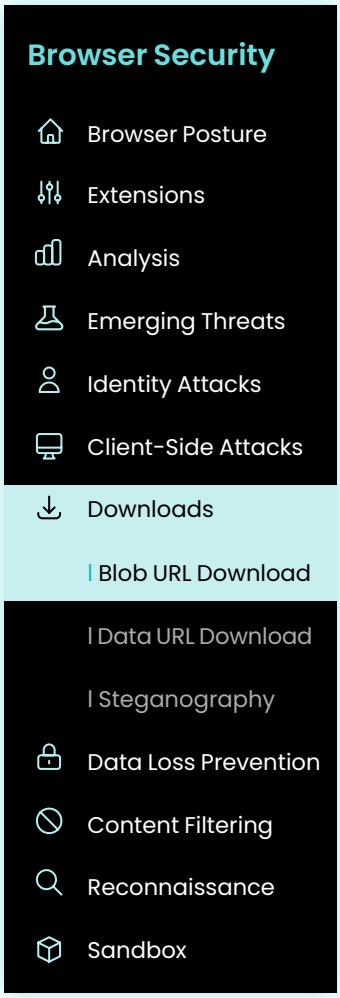


***Figure 8**: Download Module simulates file-based threats delivered through unconventional vectors like Blob URLs and steganography, showing how to block malicious downloads in real time.*

Instructions

1. Activation:
   a. This module is embedded and runs as part of the browser simulation environment.
   b. Users select which threat they would like to simulate:
      i. Blob URL Download Attack – user initiated by clicking the "Trigger Blob Download" to simulate how malicious websites



browsertotal.com

can create malicious downloadable content with executable file extensions

ii. Data URL Download Attack – user initiated by clicking "Trigger Data URL Download" to simulate how malicious websites can use Data URLs to embed file content directly in URLs and trigger downloads with executable file extensions

iii. Steganography Extraction Attack – this scenario invites users to "Extract Hidden Payload" from an embedded image that is designed to be malicious in nature

Highlights

- **Real-Time, Local Execution**: Attack simulations with sample files such that no malware being downloaded to your endpoint.
- **Comprehensive Visibility**: Detailed analysis, telemetry and explanation of how each attack works and how they can be used to bypass security measures, including specific security impacts of each threat

Zero touch malware download

HTML smuggling is a stealthy technique used on websites to bypass security defenses like firewalls and endpoint protection. It hides malicious code within seemingly harmless HTML or JavaScript, which is then reassembled and executed directly in the user's browser. Since the payload is built on the client side, it avoids detection during transmission, allowing attackers to deliver malware such as trojans or ransomware through compromised or malicious web pages.

Browser Security
Browser Posture
Extensions
Analysis
Emerging Threats
Identity Attacks
Client-Side Attacks
Downloads
Data Loss Prevention
Sensitive Data Display
Data Exfiltration
Content Filtering
Reconnaissance
Sandbox
browsertotal.com

Data Loss Prevention

Description

This module focuses on how sensitive data – whether unintentionally exposed or actively exfiltrated – can leak through the browser without triggering traditional DLP tools. Users will walk through where critical information (like PII, credentials, financial data, etc.) is exposed in the browser and where attackers attempt to extract data via clipboard abuse, screen scraping or malicious scripts, which often go undetected by traditional defenses.

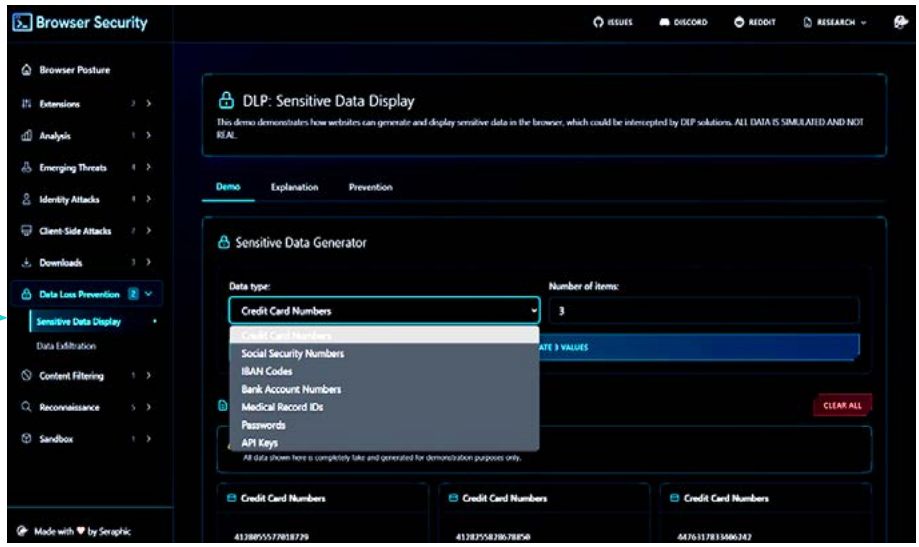
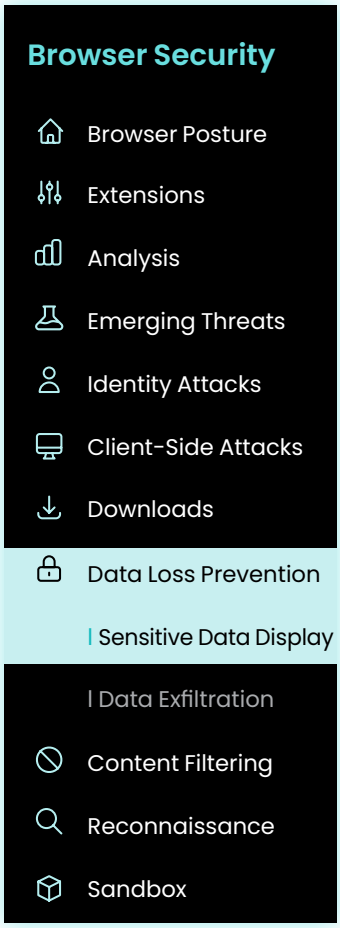


***Figure 9**: Data Loss Prevention Module highlights the ability to detect and prevent the exposure or exfiltration of sensitive data through web forms, clipboards, or developer tools.*

Instructions

1. **Activation:**
   a. This module is embedded and runs as part of the browser simulation environment.
   b. Users select which scenario they would like to run:
      i. Sensitive Data Display – this scenario demonstrates how websites can generate and display sensitive data in the browser



browsertotal.com

ii. Data Exfiltration Demo – this scenario provides an educational simulation of data exfiltration techniques being used in the wild

2. Output:

a. Demo environment displaying simulated data for educational purposes

b. Explanation of DLP and security implications of transmitting sensitive data across web-based applications

c. Samples of personal/sensitive data to test existing controls

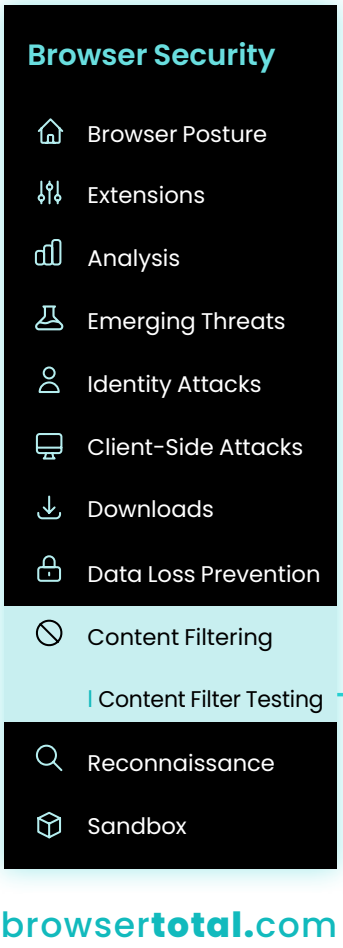
d. DLP Best Practices guidance for Organizations and Developers

Highlights

- **Real-Time, Local Execution**: Data exfiltration simulation in a secure, isolated environment with simulated, or fake, data.
- **Comprehensive Visibility**: Detailed explanations of sensitive data categorization as well preventative measures for both organizations and developers.
- **Best Practices Guidance**: Tips on how best to protect sensitive data.

Browsers can leak your most valuable information

Sophisticated data exfiltration tactics, such as copy-paste monitoring, JavaScript-based uploads, or clipboard manipulation, occur entirely within the browser. These bypass both network DLP and OS-level logging unless monitored from inside the browser



Content Filtering

Description

This module provides users with detailed instructions about how content filters work, various filtering methods, and tests how to block access to restricted or other potentially harmful categories such as gambling, adult content and more. Users will get an opportunity to experience applying various categories and attempting to reach sites – both predefined as well as user initiated – but being denied access based on established policies, including warning messages explaining what triggered the violation.

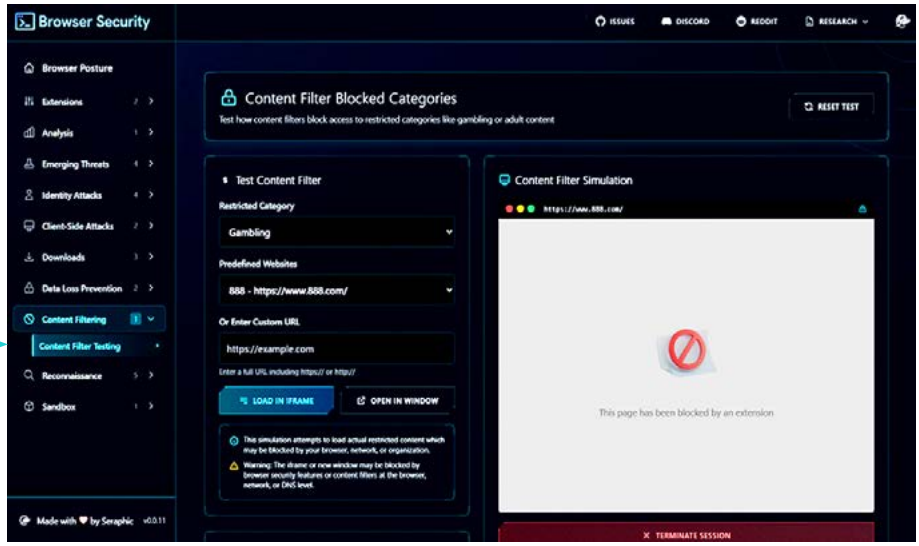


***Figure 10**: Content Filtering Module enforces access restrictions by blocking non-compliant or high-risk website categories, tailored by policy to user identity and device type.*

Instructions

1. **Activation:**
   a. This module is embedded and runs as part of the browser simulation environment.
   b. Users select a "Restricted Category"
   c. Then select a predefined website or users can enter a custom website

d. User selects "Load in Iframe" to test directly in the simulator or "Open in Window" to launch a separate browser instance

2. Output:

a. Demo environment displays a warning message indicating a violation along with explanation of policy violation/trigger

b. New browser window displays with "Seraphic Alert" informing the user of the content filtering trigger and what action has been taken

c. Test Results provide an overview of the incident and/or justification of action taken.

d. DLP Best Practices detailed guidance for Organizations and Developers

Highlights

- **Real-Time, Local Execution**: content filtering simulations in a secure, isolated environment without requiring users to access these destinations.
- **Comprehensive Visibility**: Detailed analysis and explanation of any action taken to block access to restricted sites based on acceptable use, as well as descriptions of the various content filtering methods, implementation examples, common bypass attempts and defense in depth recommendations.

Reconnaissance

Description

This module reveals how attackers silently gather intelligence through the browser to map out their targets – often as a precursor to more advanced attacks. Users will experience simulated demonstrations of key reconnaissance methods including Header Analysis, Browser Fingerprinting, IP Geoprinting, Application Enumerating, and LAN Scanning.

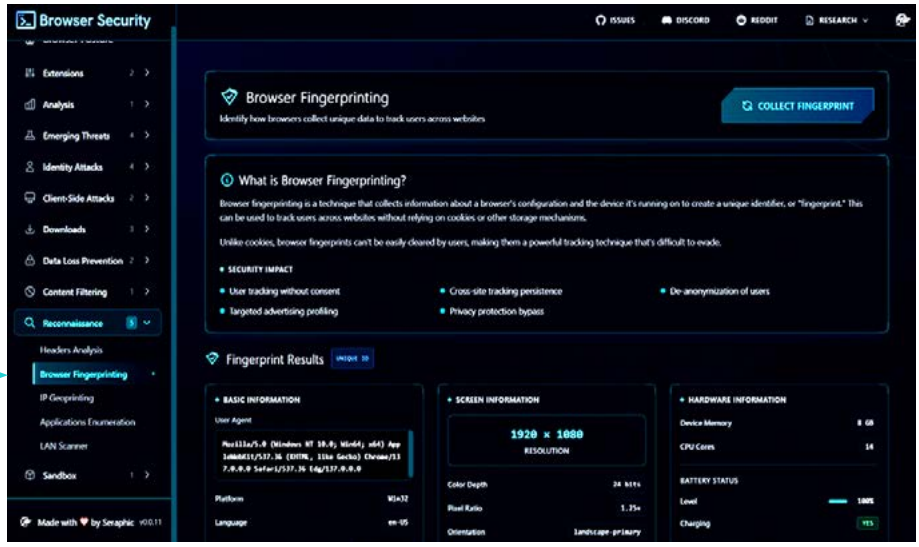
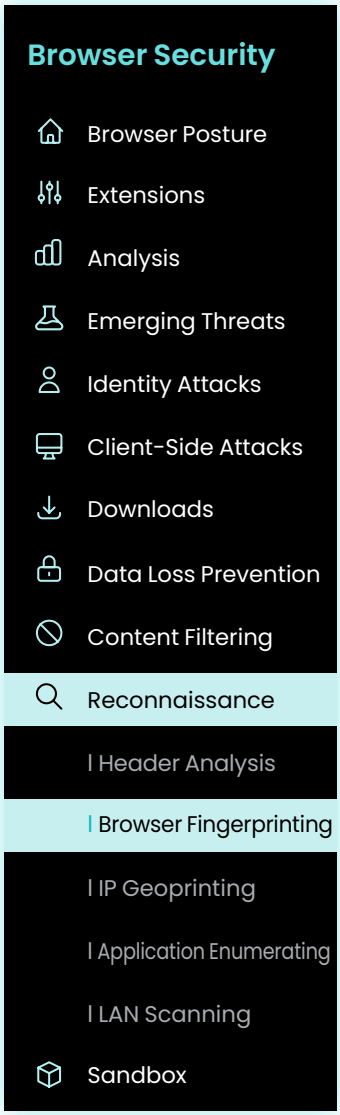


***Figure 11**: Reconnaissance Module simulates attacker recon techniques to demonstrate early detection of pre-attack behavior.*

Instructions

1. **Activation:**
   a. This module is embedded and runs as part of the browser simulation environment.
   b. Users select the type of reconnaissance activity commonly exploited in browsers to learn more:
      i. Header Analysis – this method extracts metadata from HTTP requests to profile the browser's security posture and digital fingerprint. It runs as soon as the user selects this reconnaissance option.



browsertotal.com

ii. Browser Fingerprinting – this reconnaissance method collects information about the browser's configuration and the device to track users across sessions, even without cookies. Users initiate this method by clicking on the "Collect Fingerprint" button.

iii. IP Geoprinting – this pinpoints a user location and network origin using public IP and geolocation data, and is run as soon as the user select this reconnaissance option.

iv. Applications Enumeration – this method detects what web apps or internal systems are accessible from the browser. It is run as soon as the user selects this option.

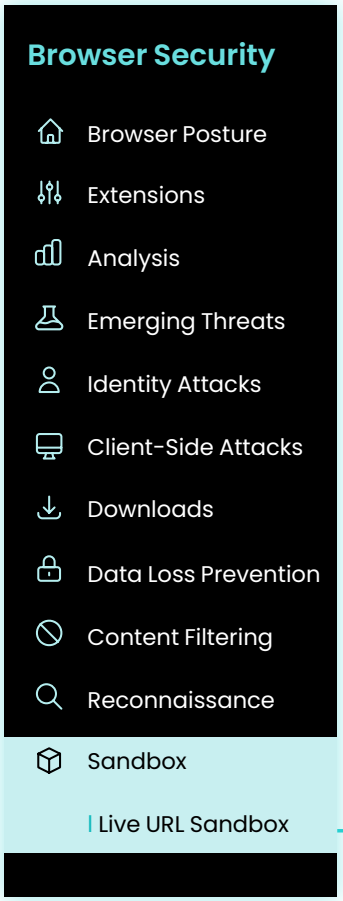
v. LAN Scanner – this method scans the local network for hosts and detects open ports. Users initiate scans by selecting "Start Scan".

Highlights

- Real-Time, Local Execution: Reconnaissance simulations are conducted in a secure, isolated environment without impacting your endpoint.
- Comprehensive Visibility: Detailed analysis and explanation of any action taken to and privacy recommendations based on best practices.

Important information is exposed with the help of your browser

One often overlooked capability of browsers is their ability to make requests to internal networks (e.g., http://localhost or http://192.168.0.1). This can be exploited by malicious websites to target internal infrastructure (so-called browser-based intranet hacking or DNS rebinding attacks) or create phishing pages designed to target specific geolocations. From an enterprise perspective, it is useful to know what internal services are reachable via the browser, as that constitutes an attack surface or potential exfiltration path.



Sandbox

Description

This module of the **BrowserTotal** platform introduces the Sandbox – a safe, isolated environment allowing users to interact with potentially malicious websites without putting their device, network, or data at risk. Users can leverage this browser-native sandbox to simulate real-world interactions while fully containing and neutralizing threats such as drive-by downloads, phishing, or script-based exploits in a zero-risk environment.

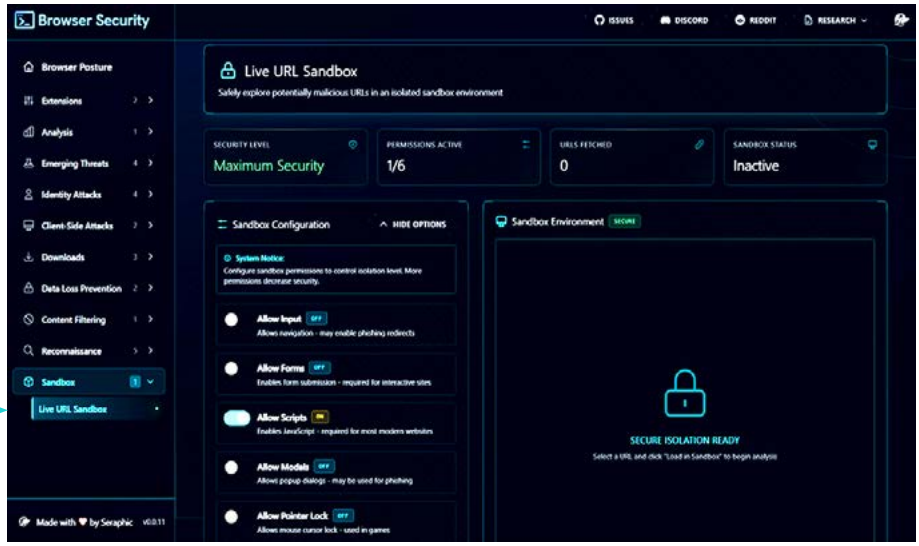
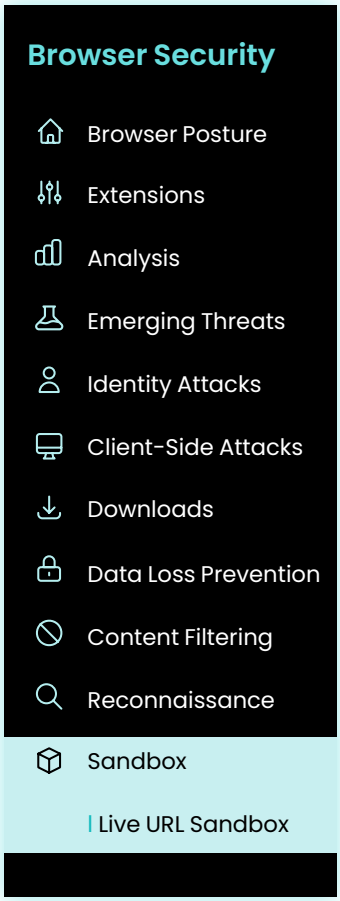


***Figure 12**: Sandbox Module provides a safe, isolated space to interact with potentially malicious as well as unknown URLs without impacting the local environment.*

Instructions

1. **Activation:**
   a. This module is embedded and runs as part of the browser simulation environment.
   b. User select the following:
      i. Configure their sandbox settings to control isolation levels and increase or decrease security levels



browsertotal.com

ii. URL Source Selection – users choose either OpenPhish or PyFunceble to collect potentially malicious URLs to test against and click on the "Fetch URL List"

iii. URL Selection – users have the option to "Use URL List" of malicious sites from OpenPhish or PyFunceble, or they can enter in their own custom URL

1. Use the dropdown list to select a URL the click "Load in Sandbox"
2. Click "Customer URL" and enter a custom URL then click "Load in Sandbox"

2. Output:

a. Activity log captures all activity in the Sandbox simulator

b. Displays security level based on sandbox configuration parameters

c. Sandbox Environment displays warning messages about potential threats

Highlights

- **Educate**: Detailed description of URL sandboxing techniques utilized to allow safe examination of potentially malicious content.
- **Best Practices Guidance**: Tips to utilize when investigating suspicious URLS to keep users, devices and data secure.

Glossary

AI-Powered Simulation

A machine-learning-enhanced environment within **BrowserTotal** that mimics real-world attack behaviors to educate users on modern browser threats.

Autonomous Module

A self-contained feature or simulation within **BrowserTotal** that runs independently in the browser without requiring external connectivity or backend processing.

Browser-Native Security

Security capabilities that operate directly within the user's existing browser environment, without needing browser replacements, extensions, or virtualization layers.

Clickjacking

A malicious technique that tricks users into clicking on something different from what they perceive, often used to hijack sessions or execute unwanted actions.

Controlled Environment

A secure, isolated testing environment where **BrowserTotal** users can explore and simulate browser-based attacks without risking actual systems or data.

Data Leakage Prevention (DLP)

A feature that simulates the detection and prevention of sensitive or confidential information leaving the browser environment.

Enterprise Browser Security

A category of cybersecurity solutions focused on securing browser-based work in enterprise environments, addressing threats like phishing, data exfiltration, and session hijacking.

Extension Risk Detection

A capability within **BrowserTotal** that identifies and simulates the risks associated with unvetted or malicious browser extensions.

Malicious Download Simulation

A **BrowserTotal** module that demonstrates how browser-based downloads—such as those using Blob URLs, Data URLs, or steganographic techniques—can be used to deliver threats.

Phishing Simulation

A test scenario within **BrowserTotal** that mimics deceptive attempts to trick users into revealing credentials or sensitive data, helping demonstrate in-browser phishing protections.

Reconnaissance Activity

The process of gathering information about a target environment, simulated in **BrowserTotal** through techniques such as browser fingerprinting, header analysis, and local network scanning.

Sandbox

A protected browser-based environment in **BrowserTotal** where users can explore potentially malicious URLs or attack flows without risk to their system or data.

Session Hijacking

A simulated attack in which an adversary takes over a valid user session—used in **BrowserTotal** to demonstrate how session integrity can be monitored and preserved.

Token Theft

The unauthorized acquisition of authentication tokens, often used in modern browser-based attacks. **BrowserTotal** simulates this to show how enterprise-grade protections can prevent token compromise.

Unmanaged Device

A device that is not directly controlled by an organization's IT team. **BrowserTotal** supports simulations on unmanaged endpoints to demonstrate policy enforcement and data protection in bring-your-own-device (BYOD) scenarios.

Zero-Day Exploit

A type of threat that targets a previously unknown vulnerability. While simulated in a safe manner within **BrowserTotal**, these threats demonstrate the importance of proactive browser defenses.