Web Application Penetration Testing Report

**25-July-2025**

**Level 1 Report**

# Copyright

The copyright in this work is vested in MCS Pvt. Ltd., and the document is issued in confidence for the purpose for which it is supplied. It must not be reproduced in whole or in part or used for tendering or manufacturing purposes except under agreement or with the consent in writing of MCS Pvt. Ltd. and then only on condition that this notice is included in any such reproduction. No information as to the contents or subject matter of this document or any part thereof arising directly or indirectly there from shall be given orally or in writing or communicated in any manner whatsoever to any third party being an individual firm or company or any employee thereof without the prior consent in writing of MCS Pvt. Ltd.

**© MCS Pvt. Ltd.**

# Contact Information

**Mr. Gaurav Kumar**

CISA, CIPM, FIP, CIPP/E, ISO 27001

LA, DevOps, MSc, PG Data Science. ( Lead Auditor)

+91 9528637406

[info@xyz.com](mailto:info@xyz.com)

**Corporate Office**

(Address)

UP

HARYANA, India

**Registered Office**

Address

# About Us

UTTAR PRADESH, INDIA

MCS Private Limited is an ISO Certified (ISO 27001:2013) company. MCS aims to provide Auditing; Consulting, Information Risk Management and Managed services tailored as per the market requirements and enhance the satisfaction and confidence index of its existing and potential customers.

MCS has been a CERT-In empaneled IT Security Auditor. It is an acknowledgement of MCS’s technical expertise in conducting Information Security Audits. As a CERT-In empaneled auditor MCS is qualified to conduct security audits of websites, networks & applications. On successfully completing the audit as per CERT-In Guidelines, our team can issue the CERT-In Certification as required by compliance requirements.

From the development of a Security Policy, Security Awareness Training, through to the delivery of complete end-to-end solutions that encompass Perimeter Security, Secure Content Management, Identity and Access Management, Vulnerability Assessment, Risk, Policy and Compliance Management, MCS Private Limited helps organizations understand, monitor and mitigate the risks in their IT infrastructure.

MCS Private Limited has information security experts, compliance professionals and process consultants from different industry verticals with multiple certifications each with experience in handling consulting assignments, audits and training programs.

MCS Consultants has established a reputation for providing practical solutions that are both businesses driven and cost-effective. This has enabled the company to secure the IT Infrastructure of leading Stock Brokers, Government undertakings, Banks, Insurance Companies and Financial Institutions, BPO, KPO, PKI Industry, Data Center, Software Company, Automobile, Healthcare & Life Sciences, Hospitality, Travel, Transportation, Consumer & Retail, Technology, Media & Telecommunication, School, College, Smart City. Security Leadership our experienced team of ethical hackers have identified security vulnerabilities, we have put together a highly qualified team of security researchers with credentials like CISA, CISSP, OSCP, CSOE, CCNP, MCSE+, ISO 27001 LA, ITIL, COBIT, CIPP/E, CIPM, FIS, DISA, CPA, Cyber Law, Data Science, Artificial Intelligence & Machine Learning, DevSecOps, AWS Architect, Azure Architect.

# Table of Contents

* 1. 1.1 Copyright 2
  2. [Contact Information 2](#_bookmark0)

1. [About Us 3](#_bookmark1)
2. [Table of Contents 4](#_bookmark2)
3. Assessment Information 5
   1. Assessment scope summary 5
   2. MCS Pvt. Ltd. Details 5
4. [Engagement Overview 6](#_bookmark3)
5. [Process and Methodology 7](#_bookmark4)
6. [Vulnerability Overview 8](#_bookmark5)
7. [Checklist 9](#_bookmark6)
8. [Test Execution Summary 13](#_bookmark7)
9. [Executive Summary 14](#_bookmark8)
10. Summary vulnerabilities 15
11. [Detailed Findings and Recommendation 16](#_bookmark9)
12. [High-Level Recommendations 26](#_bookmark10)
13. Engagement Overview

**Indian Council of Historical Research** has engaged with **MCS** to conduct a penetration test of their **Web Application**. This report contains all the results of the report as well as all the action items that were included in the penetration test. The purpose of this report is to present the current security level of the external perimeters including gaps, vulnerabilities, and misconfigurations. The findings presented in this report should be fixed to improve the security level of the network systems.

## Service Description

Web application Vulnerability Assessment and Penetration Testing (VAPT) is the process of simulating real-world attacks by using the same techniques as malicious hackers. For a security assessment that goes beyond a simple vulnerability scanner, you need experts in the industry. **MCS** conducts its penetration test by approaching the scope with both a manual and automatic approach.

## Web Application Penetration Test

our application-level penetration testing consists of both unauthenticated and authenticated testing using both automated and manual methods with particular emphasis placed on identifying vulnerabilities associated with the OWASP Top 10 Most Critical Application Vulnerabilities. It is important to note that a penetration test is not just an automated vulnerability scan, and a large portion of web application penetration testing is a manual process with a skilled engineer attempting to identify, exploit, and evaluate the associate risk of security issues.

## Project Objectives

**MCS** consultants conduct all testing manually combined with custom and commercial tools that perform unique attack approaches on the network to make sure we cover the whole system in the test. Our expert knowledge and experience are the value we provide in our services.

# Process and Methodology

MCS has developed a proven Vulnerability Assessment/Penetration Testing Methodology (illustrated below) from best practices including the Open-Source Security Testing Methodology Manual (OSSTMM), the Council for Registered Ethical Security Testers (CREST), the Penetration Testing Execution Standard (PTES), and our 15 plus years of experience. We have also scaled the methodology to account for differing risks and preferred engagement modalities to ensure that we can provide the right testing and assurance at the right cost.

## Reconnaissance



**1**

This process begins with a detailed scanning and enumeration of the network system and infrastructure, and the information related to the system that is exposed on the internet. After this stage, we conduct manual testing of the gathered data to be analyzed further for attack paths.



**2**

## Automated Approach

Once the scope has been fully enumerated, we start to approach the scope manually and with automatic approaches both self- developed and commercial. The goal is to cover most of the network.

## Exploitation And Manual Testing



**3**

At this stage of the assessment, our experts review the gathered attack vectors and try to exploit the vulnerabilities found in a manual safely approach.

## Assessment Report



**4**

Once the engagement has been completed, we start to develop the assessment report which consists of an executive part and a technical part where we present all the findings and recommendations for fix.

## Re-test



**4**

As an additional option to our standard assessment, we offer a full re-test to our clients to check whether the issues has been fixed by the client and if it’s possible to bypass the patches that were implemented. This will then be updated by a new re-test report

# Vulnerability Overview

**MCS** performed a Network Security Assessment for **Indian Council of Historical Research.** on **2024-10-26** to **2024-10-28** This assessment utilized both commercial and proprietary tools for the initial mapping and reconnaissance of the network system as well as custom tools and scripts for unique vulnerabilities. Several attacking tools was used in order to conduct the penetration testing on the network.

## VULNERABILITY RISK DEFINITION AND CRITERIA

The risk rating assigned to each vulnerability are determined by averaging several aspects of the exploit and the environment, including reputation, difficulty and impact.

Critical vulnerabilities pose a serious threat to an organization's security, and should be fixed immediately. They may provide a total compromise of the target environment, or similar critical impacts.



**Critical**

High risk vulnerabilities provide a serious risk to the company environment and should be corrected promptly. These issues can significantly affect the organization's security posture.



**High**

Medium severity vulnerabilities represent a moderate risk to the environment. They may require additional context before remediation but should be remediated after critical and high risks.



**Medium**

Low severity vulnerabilities provide minimal risk to the target environment, and often theoretical in nature. Remediation of low risks is often a lower priority than other security hardening techniques.



**Low**

**Informational**

Informational vulnerabilities have little-or-no impact to the target scope by themselves. They are included however, as they may be a risk when combined with other circumstances or technologies not currently in place. Remediation of informational items is not necessary.

# Checklist

|  |  |  |
| --- | --- | --- |
| **MCS Web Application Security Testing Checklist** | | |
|  | **Objective** | **Test Outcome** |
|  | **Data Validation** |  |
|  | * Test Error page leak * Test for Stored Cross Site * Test for DOM based Cross Site Scripting * Test for Cross Site Flashing * Test for HTML Injection * Test for SQL Injection * Test for LDAP Injection * Test for ORM Injection * Test for XML Injection * Test for XXE Injection * Test for SSI Injection * Test for XPath Injection * Test for XQuery Injection * Test for IMAP/SMTP Injection * Test for Code Injection * Test for Expression Language Injection * Test for Command Injection * Test for Overflow (Stack, Heap and Integer) * Test for incubated vulnerabilities * Test for HTTP Splitting/Smuggling Test for HTTP Verb Tampering * Test for Open Redirection * Test for Local File Inclusion * Test for Remote File Inclusion * Compare client-side and server-side validation rules Test for NoSQL injection * Test for HTTP parameter pollution Test for auto-binding * Test for Mass Assignment * Test for NULL/Invalid Session Cookie | Not Found |

|  |  |  |
| --- | --- | --- |
|  | **Session Management** |  |
|  | * Establish how session management is handled in the application (eg, tokens in cookies, token in URL) Check session tokens for cookie flags (httpOnly and secure) * Check session cookie scope (path and domain) Check session cookie duration (expires and max-age) Check session termination after a maximum lifetime Check session termination after relative timeout Check session termination after logout * Test to see if users can have multiple simultaneous sessions Test session cookies for randomness * Confirm that new session tokens are issued on login, role change and logout * Test for consistent session management across applications with shared session management * Test for session puzzling * Test for CSRF and clickjacking | Not Found |
|  | **Authentication** |  |
|  | * Test for user enumeration Test for authentication bypass * Test for bruteforce protection * Test password quality rules * Test remember me functionality * Test for autocomplete on password forms/input * Test password reset and/or recovery * Test password change process Test CAPTCHA * Test multi factor authentication * Test for logout functionality presence * Test for cache management on HTTP (eg Pragma, Expires, Max-age) Test for default logins * Test for user-accessible authentication history * Test for out-of channel notification of account lockouts and successful password changes * Test for consistent authentication across applications with shared authentication schema / SSO | Not Found |
|  | **Configuration Management** |  |
|  | * Check for commonly used application and administrative URLs Check for old, backup and unreferenced files * Check HTTP methods supported and Cross Site Tracing (XST) * Test file extensions handling * Test for security HTTP headers (e.g. CSP, X-Frame-Options, HSTS) * Test for policies (e.g. Flash, Silverlight, robots) * Test for non-production data in live environment, and vice-versa | Not Found |

|  |  |  |
| --- | --- | --- |
|  | * Check for sensitive data in client-side code (e.g. API keys, credentials) |  |
|  | **Authorization** |  |
|  | * Test for path traversal * Test for bypassing authorization schema * Test for vertical Access control problems (a.k.a. Privilege Escalation) * Test for horizontal Access control problems (between two users at the same privilege level) * Test for missing authorization | Found |
|  | **Business Logic** |  |
|  | * Test for feature misuse * Test for lack of non-repudiation * Test for trust relationships * Test for integrity of data * Test segregation of duties | Not Found |
|  | **Denial of Service** |  |
|  | * Test for anti-automation * Test for account lockout * Test for HTTP protocol DoS * Test for SQL wildcard DoS | Not Found |
|  | **Risky Functionality - File Uploads** |  |
|  | * Test that file size limits, upload frequency and total file counts are defined and are enforced * Test that file contents match the defined file type * Test that all file uploads have Anti-Virus scanning in-place. Test that unsafe filenames are sanitized * Test that uploaded files are not directly accessible within the web root * Test that uploaded files are not served on the same hostname/port * Test that files and other media are integrated with the authentication and authorization schemas | Not Found |
|  | **Risky Functionality - Card Payment** |  |
|  | * Test for known vulnerabilities and configuration issues on Web Server and Web Application * Test for default or guessable password * Test for non-production data in live environment, and vice-versa * Test for Injection vulnerabilities * Test for Buffer Overflows * Test for Insecure Cryptographic Storage | Found |

|  |  |  |
| --- | --- | --- |
|  | * Test for Insufficient Transport Layer Protection * Test for Improper Error Handling * Test for all vulnerabilities with a CVSS v2 score > 4.0 * Test for Authentication and Authorization issues * Test for CSRF |  |
|  | **Cryptography** |  |
|  | * Check if data which should be encrypted is not * Check for wrong algorithms usage depending on context * Check for weak algorithms usage * Check for proper use of salting * Check for randomness functions | Not Found |
|  | **Secure Transmission** |  |
|  | * Check SSL Version, Algorithms, Key length * Check for Digital Certificate Validity (Duration, Signature and CN) * Check credentials only delivered over HTTPS * Check that the login form is delivered over HTTPS * Check session tokens only delivered over HTTPS * Check if HTTP Strict Transport Security (HSTS) in use | Not Found |

# Test Execution Summary

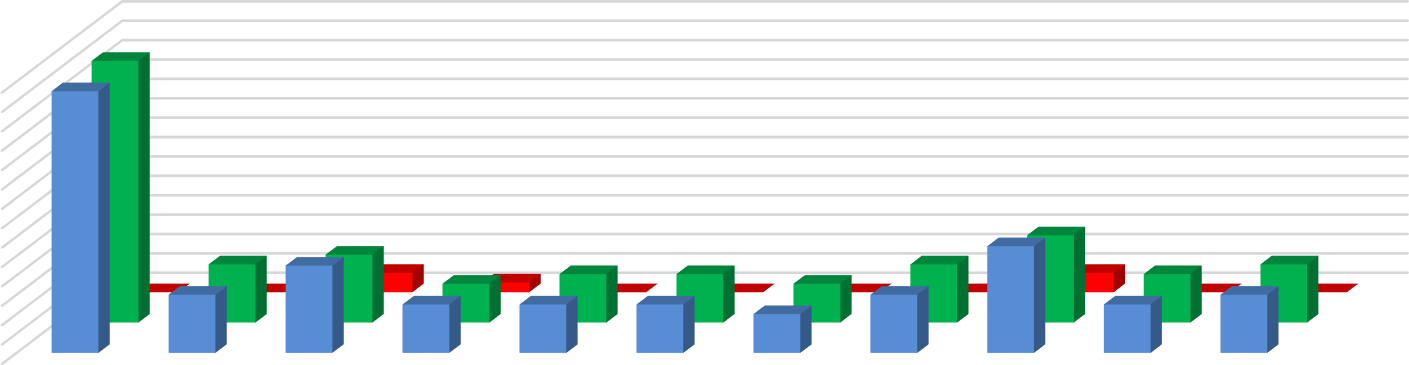


Chart Title

28

26

24

22

20

18

16

14

12

10

8

6

4

2

0

No. Of Failed Cases

No. Of Passed Cases

No. of Test Cases

No. of Test Cases

No. Of Passed Cases

No. Of Failed Cases

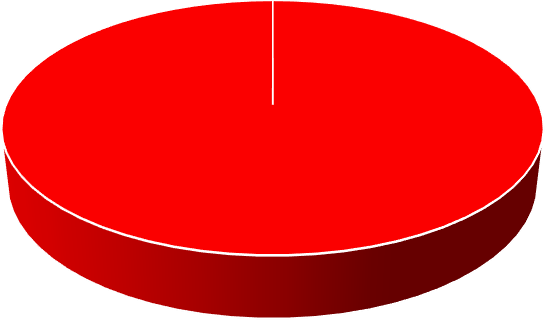
# Executive Summary:

MCS has been engaged by **Indian Council of Historical Research.** to undertake security testing against the [**http://ichr.ac.in/v3**](http://ichr.ac.in/v3)web application. The **Level 1** testing took place over the period from **26/10/2024** to **28/10/2024**. During this period the application was analyzed and assessed using a combination of standard tools and utilities and the knowledge and experience of our technical team. Although at the time of this engagement, the application was not in production, we nonetheless stopped short of undertaking specific tests that would either a) evidently risk the integrity and stability of the systems, or b) actively exploit potential vulnerabilities. Overall, we believe that a reasonable level of security has been attained by the applications that were the target of this test, but due to there being a high and some medium and low risk issues, remedial action needs to be carried out prior to official launch of the product. Testing revealed elements that are well-protected against several well-known vulnerabilities.



Vulnerability by Severity

Critical High Medium Low



Vulnerability Status

Fixed Open

# Summary of Vulnerabilities

Based on the assessment carried out by MCS from **25/07/2025** to **28/07/2025**, the following notable issues have been identified in reference to OWASP standards. This section provides a quick snapshot of the security posture for the Web Application.

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Vulnerability** | **Severity** | **Level 1 –**  **25/07/2025** |
| **1** | **X-Content-Type-Options header is not set** | **Medium** | **Open** |
| **2** | **Absence of Anti CSRF Token** | **Medium** | **Open** |
| **3** | **CSP: Failure to Define Directive with No Fallback** | **Medium** | **Open** |
| **4** | **CSP: Wildcard Directive** | **Medium** | **Open** |
| **5** | **CSP: script-src unsafe-inline** | **Medium** | **Open** |
| **6** | **CSP: style-src unsafe-inline** | **Medium** | **Open** |
| **7** | **Strict-Transport-Security Header Not Set** | **Low** | **Open** |
| **8** | **Timestamp Disclosure - Unix** | **Low** | **Open** |
| **9** | **X-Content-Type-Options Header Missing** | **Low** | **Open** |
| **10** | **Authentication Request Indentified** | **Informational** | **Open** |
| **11** | **Cookie Poisoning** | **Informational** | **Open** |
| **12** | **Information Disclosure - Suspicious Comments** | **Informational** | **Open** |
| **13** | **Session Management Response Identified** | **Information** | **Open** |

# Detailed Findings and Recommendation

h

**12.1 X-Content-Type-Options header is not set**

**Relative Risk: - Medium**

**Impact:** - An attacker could trick the browser into interpreting the response with an incorrect MIME type

Malicious scripts (such as XSS) may be executed if MIME sniffing is enabled

Security controls could be bypassed, leading to a higher risk of exploitation.

**Description: - The web server response is missing the X-Content-Type-Options HTTP security header. This header prevents browsers from MIME type sniffing the response. Without it, browsers may guess the content type of files, which could allow malicious content to be executed**

# 

# 

**Solution: -** Add the X-Content-Type-Options header in the web server or application configuration:

X-Content-Type-Options: nosniff

**Tool Used:- Nikto**

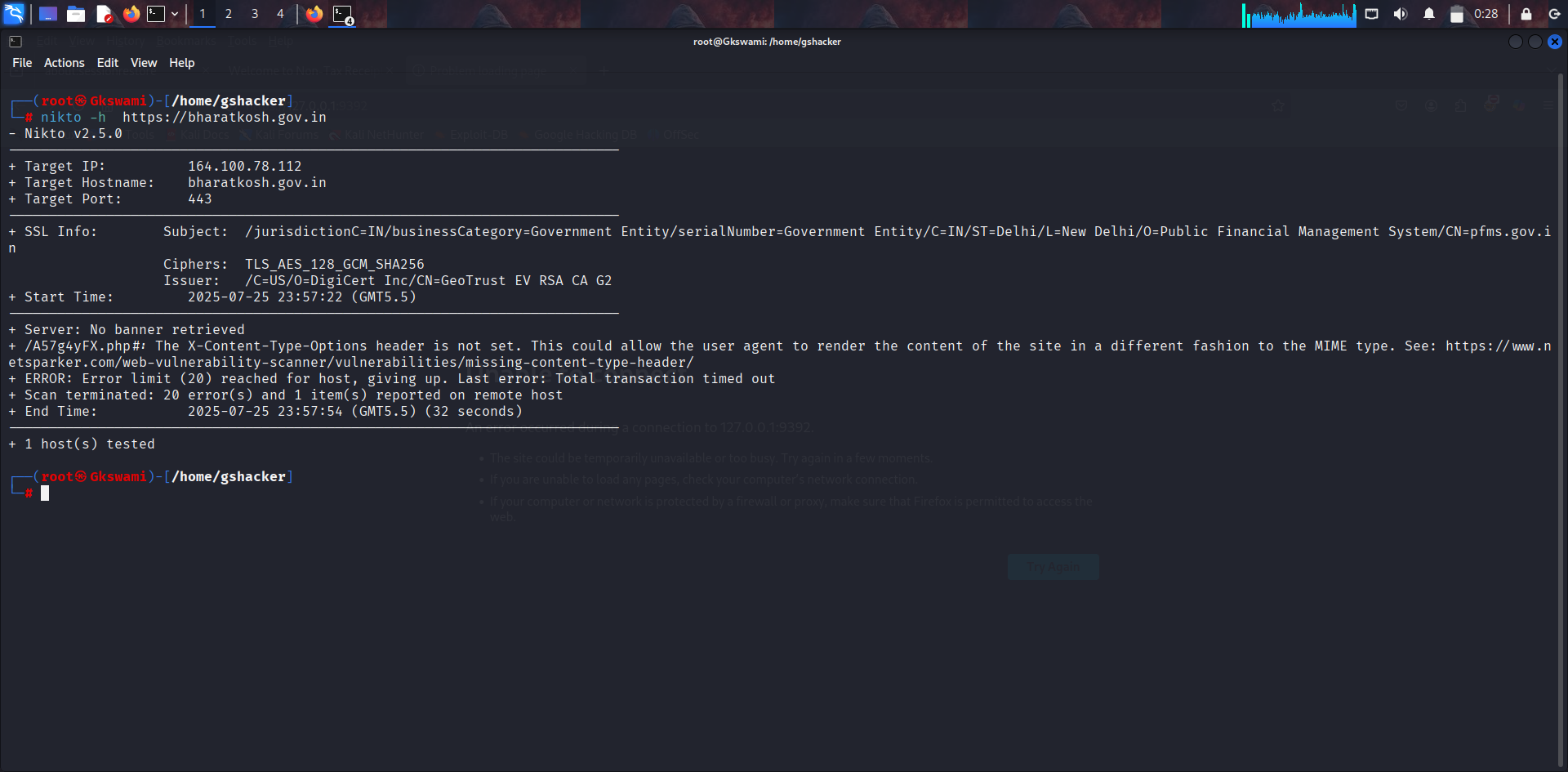
**Affected** <URL:-> <https://bharatkosh.gov.in/>

# 

# 

# Proof of concept 12.1

# 





**Relative Risk: Medium**

**Description:-** No Anti-CSRF tokens were found in a HTML submission form.

A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.

12.2 **Absence of Anti CSRF Token**

**Affected URL:** <https://bharatkosh.gov.in/loginRegistrationProcess/ForgetPassword>

**Tool used** : OWASP ZAP

**Impact**: Without CSRF tokens, the server cannot distinguish between legitimate user requests and forged requests sent by an attacker.

If a victim is authenticated, an attacker can craft a malicious HTML form or request that performs actions with the victim’s

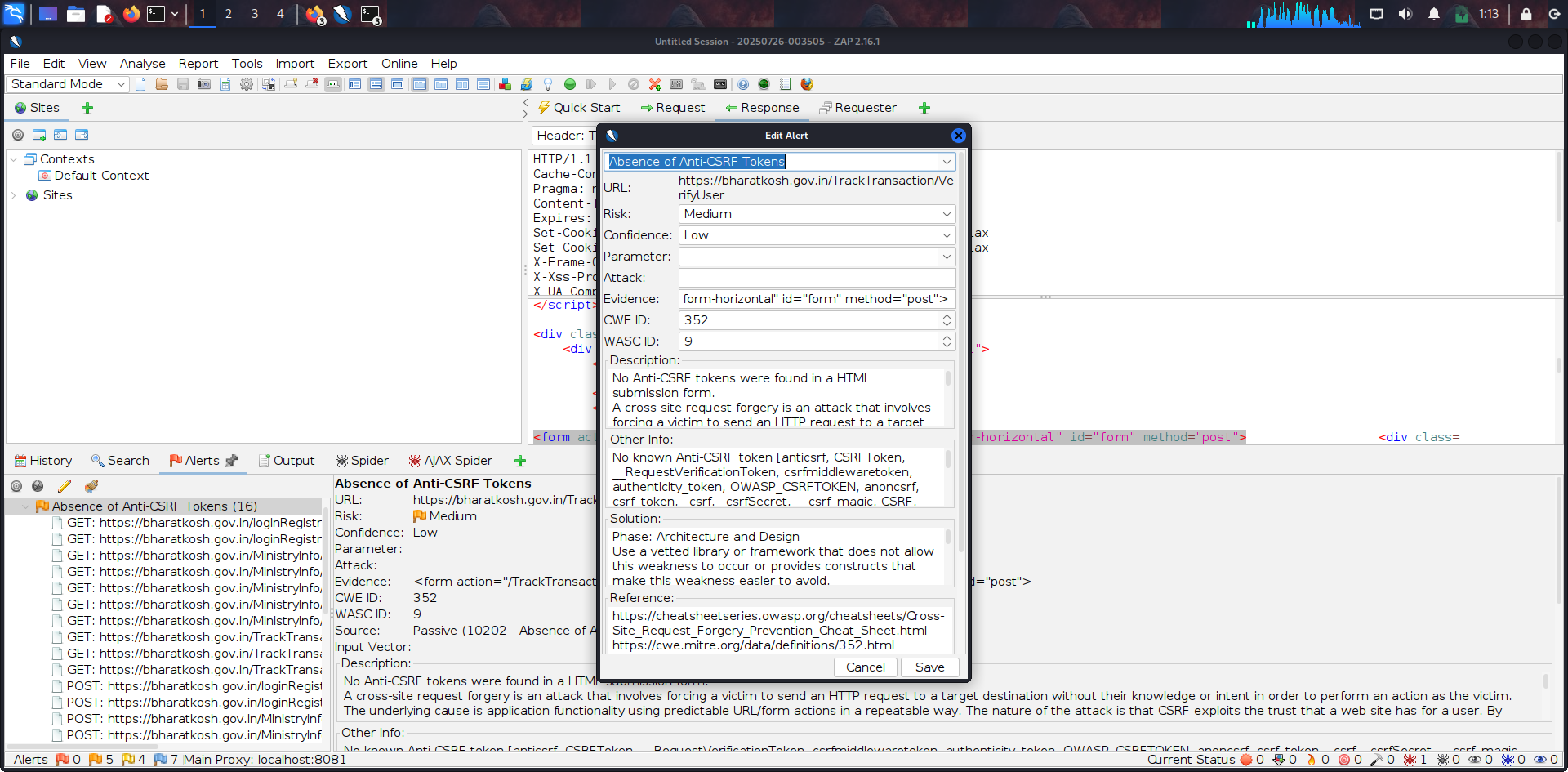
credentials — without their consent or awareness.

**Solution:-** Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness

easier to avoid.

For example, use anti-CSRF packages such as the OWASP CSRF Guard.

**Proof of Concept 12.2**

****

**12.3**  **CSP: Failure to Define Directive with No Fallback**

**Relative Risk:**  Medium

**Description:** The Content Security Policy fails to define one of the directives that has no fallback.

Missing/excluding them is the same as allowing anything.

**Impact:** When a CSP directive like script-src or object-src is not explicitly defined and there's no default-src fallback, the browser may allow the execution or loading of potentially unsafe content. This increases the risk of:

* **Reflected and Stored XSS attacks**
* **Clickjacking**
* **Data exfiltration via malicious scripts**
* **Execution of untrusted third-party content**

Attackers may exploit these weaknesses to:

* Steal session tokens or sensitive information
* Deface the application
* Redirect users to malicious sites

**Recommendation:** Ensure that your web server, application server, load balancer, etc. is properly

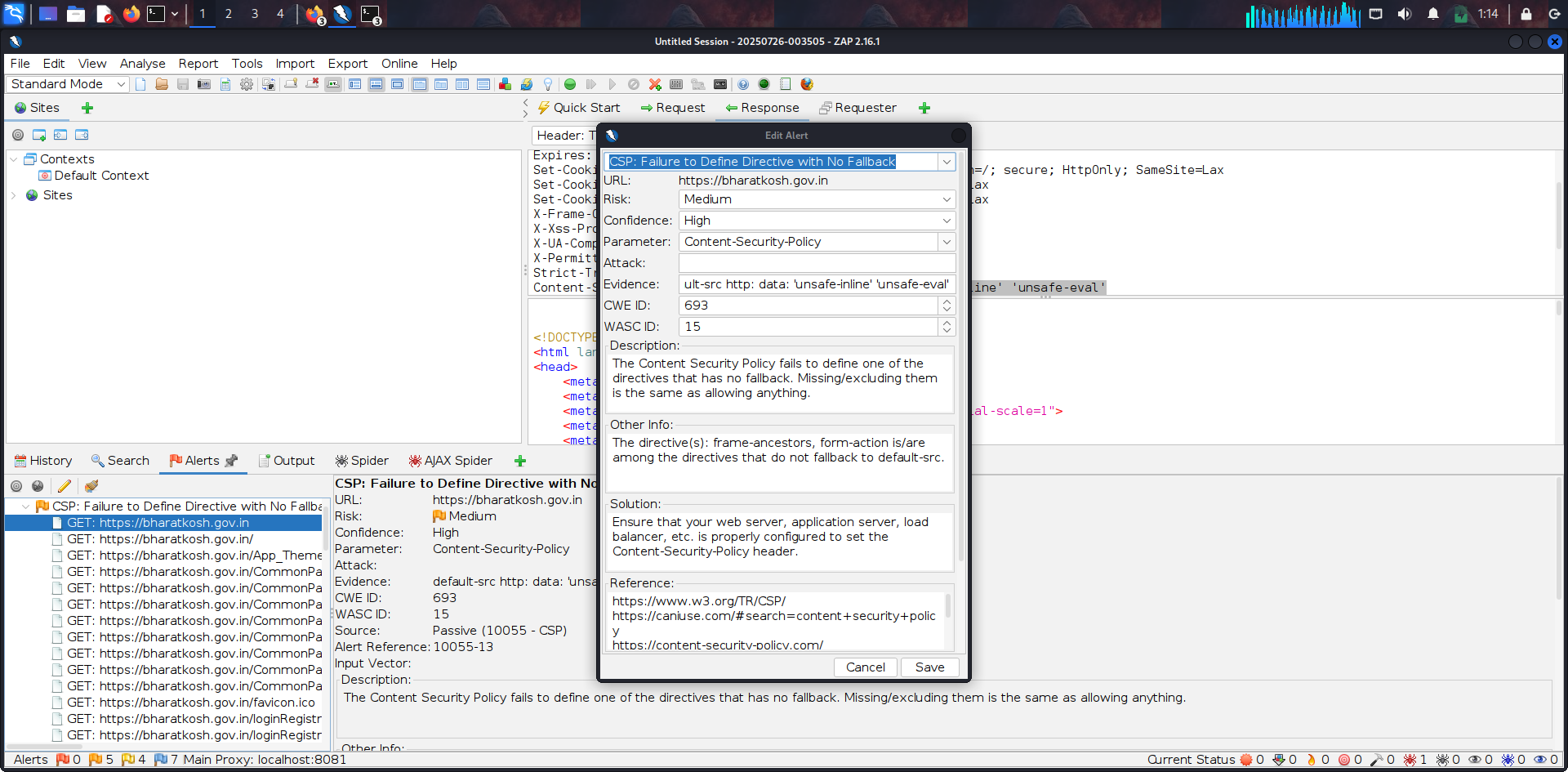
configured to set the Content-Security-Policy header.

**Tool Used: OWASP ZAP**

**Affected** <URL:-> <https://bharatkosh.gov.in/>

### OCUMENT

Proof of Concept 12.3



12.4 **CSP: Wildcard Directive**

**Solution:** Remove or mask these headers in production environment.

**Tool Used:**  OWASP ZAP

**Affected** <URL:-> <https://bharatkosh.gov.in/>

**Impact: -** The use of wildcard (\*) in CSP directives weakens the security policy by allowing content to load

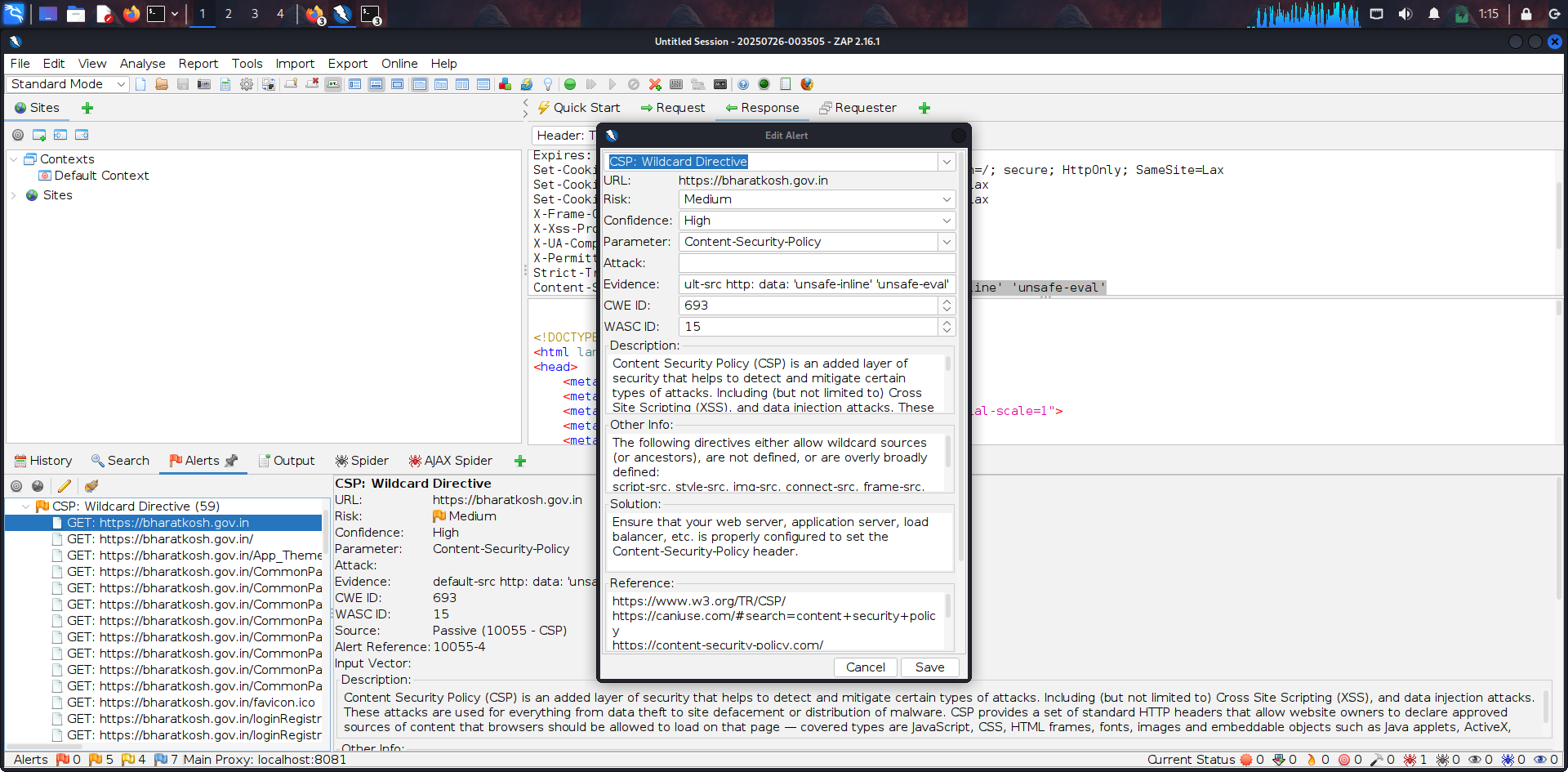
from any source. This can lead to XSS attacks, data theft, or content injection. It defeats the purpose of defining trusted content sources. CSP should restrict resources to specific, trusted domains.

**Description:** Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate

certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

**Relative Risk: Medium**

**Proof of Concept 12.4**

****

**12.5 CSP: script-src unsafe-inline**

**Relative Risk:- Medium**

**Description:** Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate

certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

**Evidence From ZAP Scan:** default-src http: data: 'unsafe-inline' 'unsafe-eval'

**Impact:** Using script-src 'unsafe-inline' in CSP allows inline JavaScript execution, which makes the application vulnerable

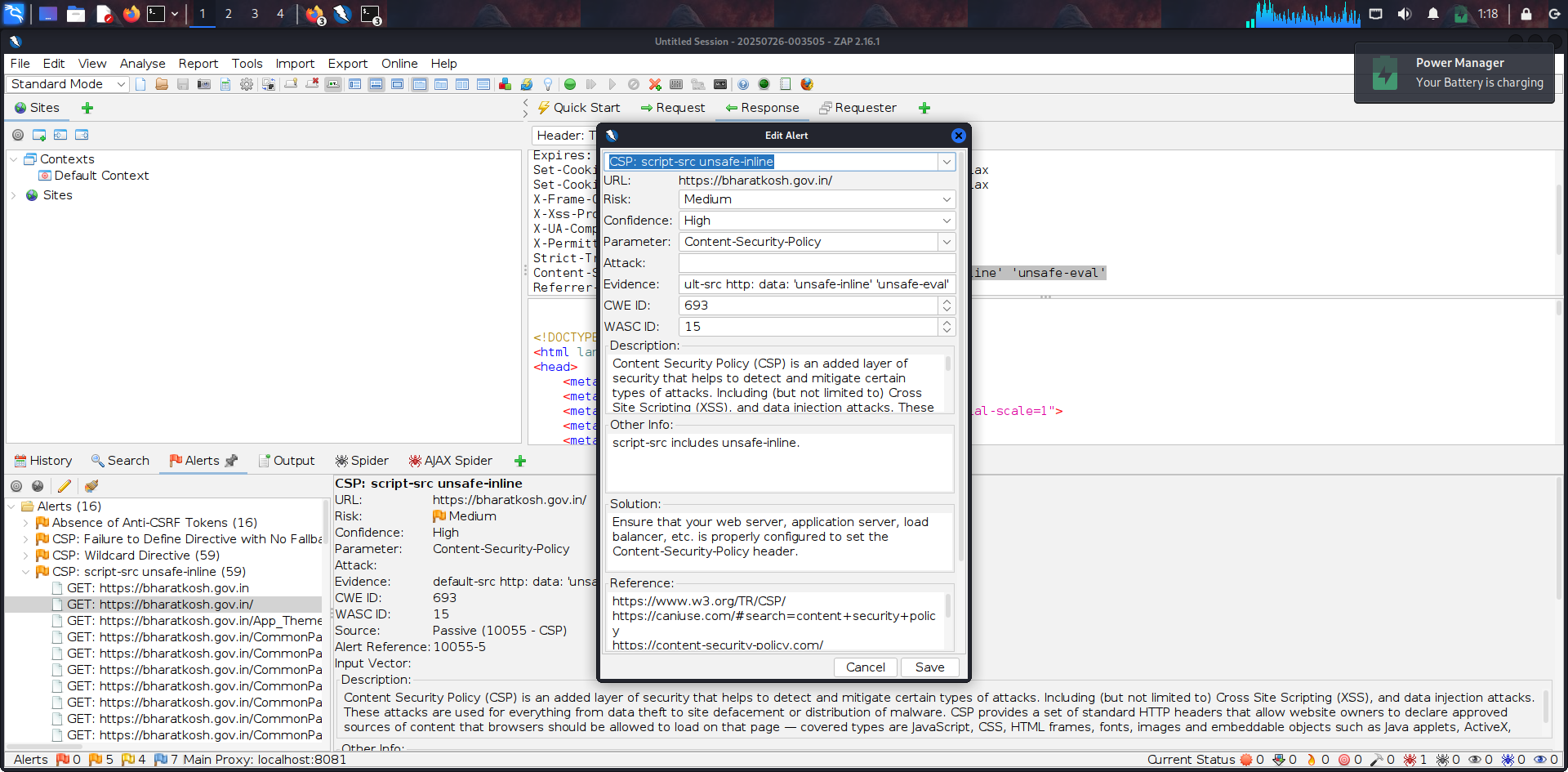
to Cross-Site Scripting (XSS) attacks. It bypasses one of the core protections provided by CSP. This weakens the browser’s ability to prevent malicious script execution.

**Solution:**  Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.

**Affect URL:** <https://bharatkosh.gov.in/App_Themes/Receipt/images/MinFin.jpg>

**Tool used**:- Owasp Zap

**Proof of Concept 12.5**



**12.6 CSP: style-src unsafe-inline**

**Relative Risk:** Medium

**Impact: -** Using style-src 'unsafe-inline' in CSP allows inline CSS to execute, which can enable style-based injection attacks.

Attackers may abuse this to perform UI redress or data exfiltration via malicious styles. It reduces the effectiveness of CSP in preventing content injection

**Description:** Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but

not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

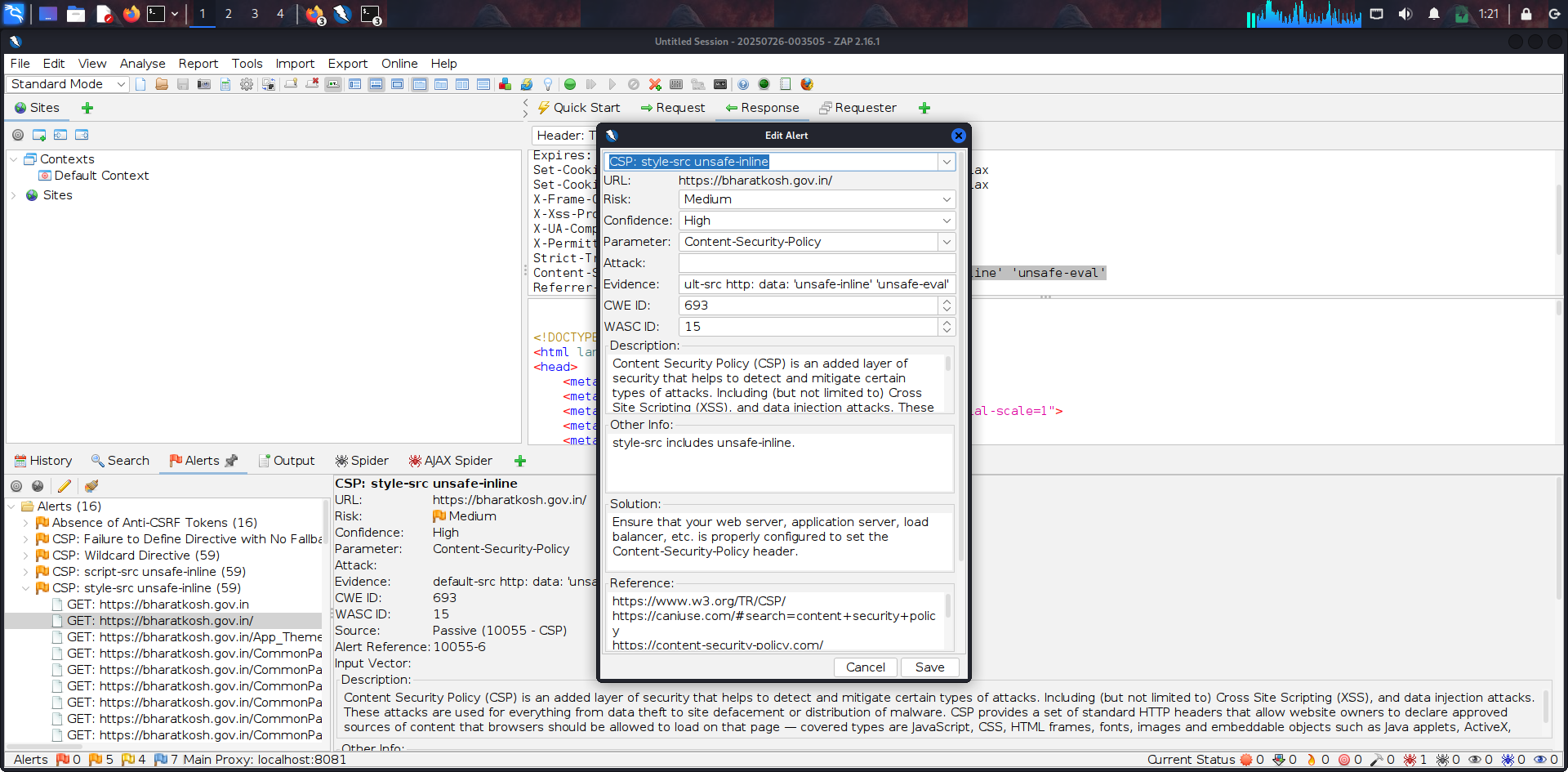
**Tool Used**:- Owasp Zap

**Affect URL:** <https://bharatkosh.gov.in/sitemap.xml>

**Solution:** Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-

Policy header.

##### Proof of Concept 12.6

****

**12.7 Strict-Transport-Security Header Not Set**

###### Relative Risk: Medium

**Affect URL:** <https://bharatkosh.gov.in/App_Themes/Receipt/banners/contact_us.jpg>

**Tool Used :** OWASP ZAP

**Impact:** Absence of the Strict-Transport-Security (HSTS) header allows attackers to perform SSL stripping attacks by forcing users to access the site over

HTTP. This can lead to sensitive data being transmitted in cleartext. Without HSTS, users are not forcibly redirected to secure HTTPS connections.

**Solution:** Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.

**Description:** HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user

agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797



CONFIDENTIAL DOCUMENT

Proof Of Concept 12.7





CONFIDENTIAL DOCUMENT

12.8 **Timestamp Disclosure - Unix**

###### 

**Relative Risk:- Medium**

**Description:** A timestamp was disclosed by the application/web server. - Unix

**Impact:** Unix timestamp disclosure can reveal system or file generation times, which may help attackers estimate

software deployment dates or session validity. This information can aid in targeted attacks like session prediction or version-based exploitation. Though low risk, it can support other vulnerabilities.

**Solution:** Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to

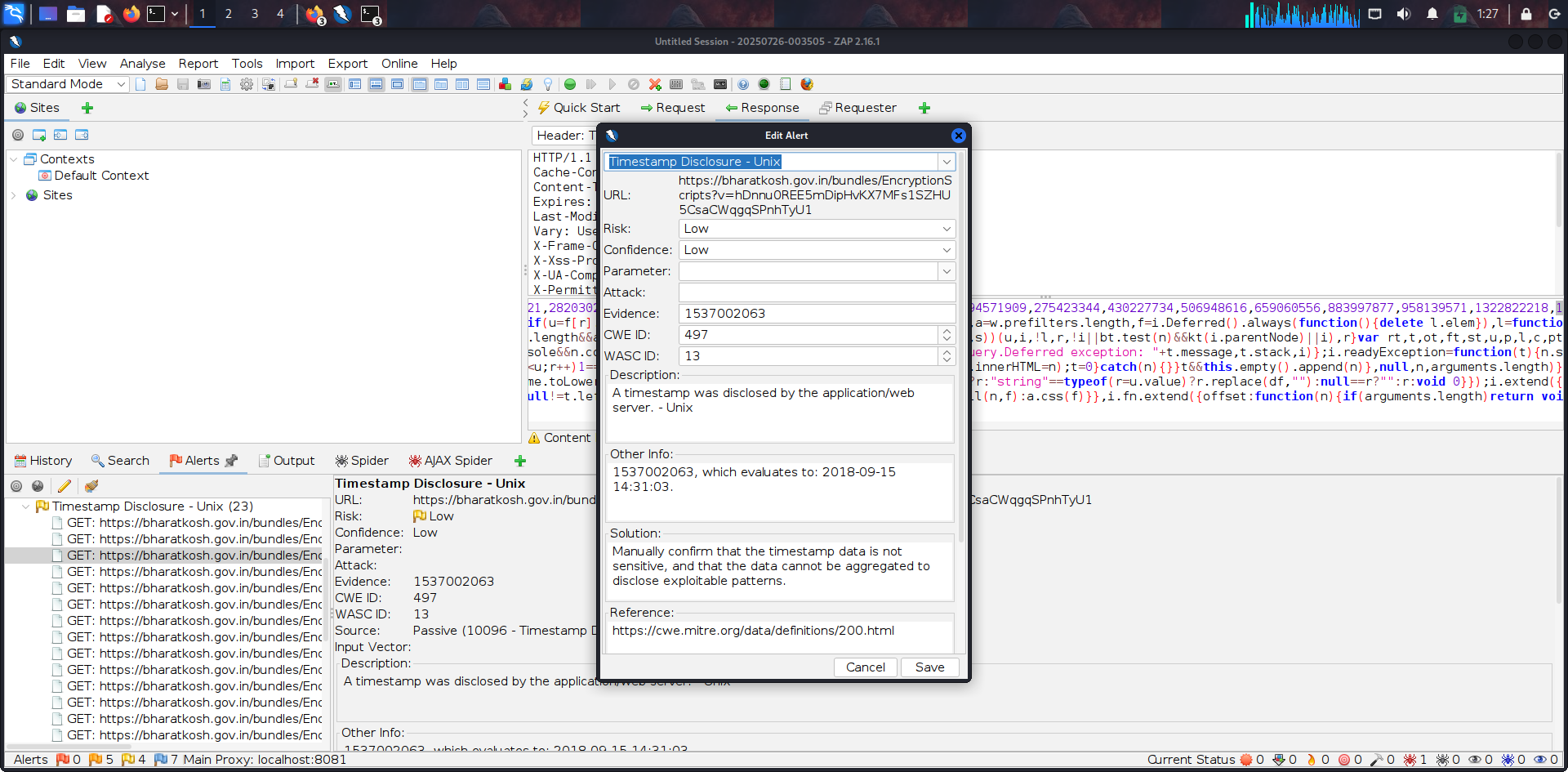
disclose exploitable patterns.

**Tool Used: OWASP ZAP**

**Affect URl:** <https://bharatkosh.gov.in/bundles/EncryptionScripts?v=hDnnu0REE5mDipHvKX7MFs1SZHU5CsaCWqgqSPnhTyU1>

**Evidence: 1508970993**

Proof of Concept 12.8



**12.9 X-Content-Type-Options Header Missing**

**Relative risk: Low**

**Description:** A timestamp was disclosed by the application/web server. - Unix

**Impact:**

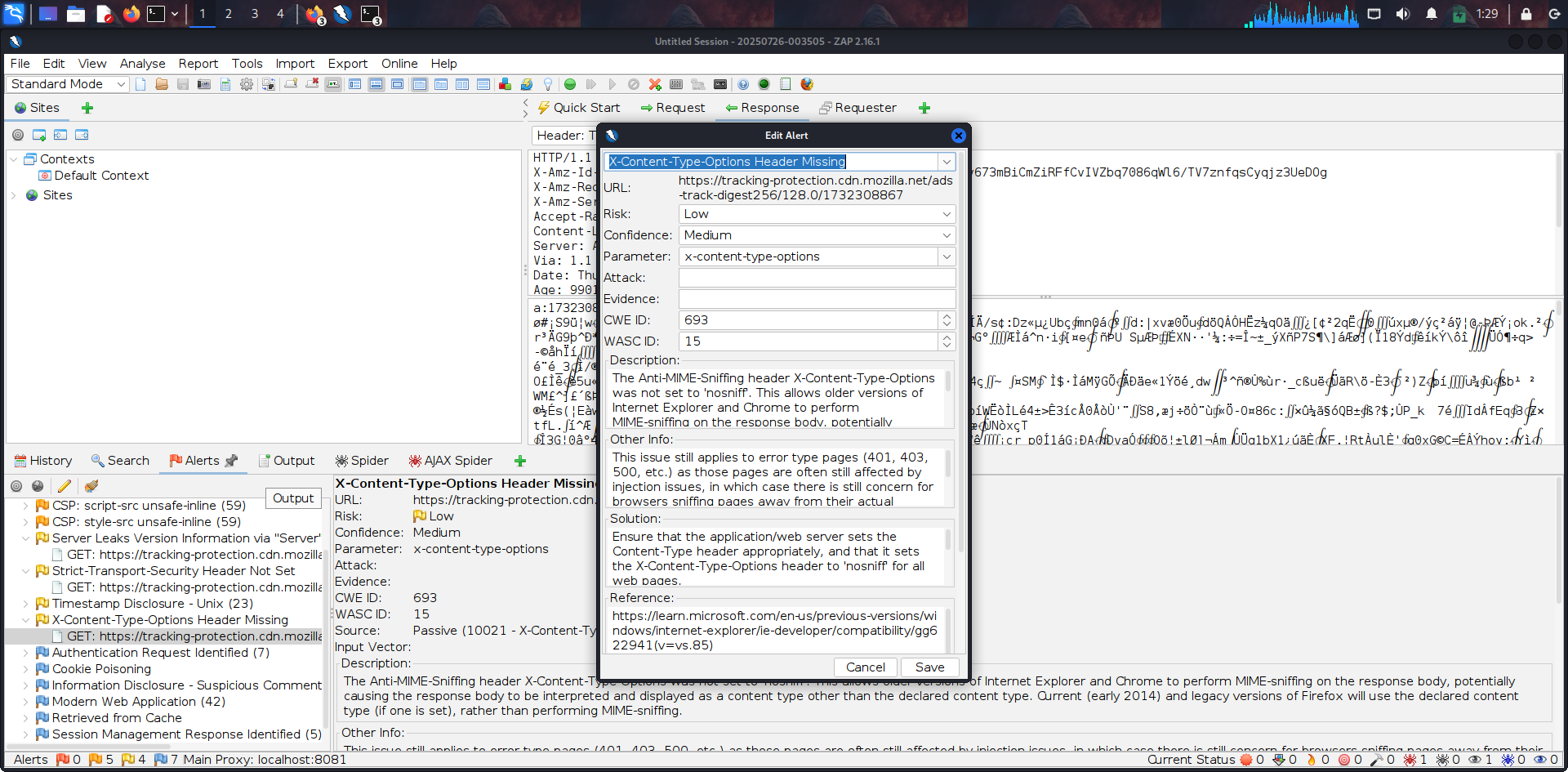
Absence of the X-Content-Type-Options: nosniff header allows browsers to perform MIME type sniffing, which can lead to execution of malicious files as scripts. This increases the risk of drive-by downloads and content-type confusion attacks. The header helps prevent such unintended behavior.

**Solution:** Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose

exploitable patterns.

**Tool Used:** OWASP ZAP

Proof of Concept 12.9



**12.10 Authentication Request Indentified**

**Relative risk:** Informational

**Description:** The given request has been identified as an authentication request. The 'Other Info' field contains a set of key=value lines which identify any relevant fields. If the request is in a context which has an Authentication Method set to "Auto-Detect" then this rule will change the authentication to match the request identified.

**Impact:** Identifying authentication requests can allow attackers to detect login endpoints or authentication mechanisms. This information can aid in brute-force, credential stuffing, or targeted phishing attacks. Even without credentials, exposing such endpoints increases the attack surface.

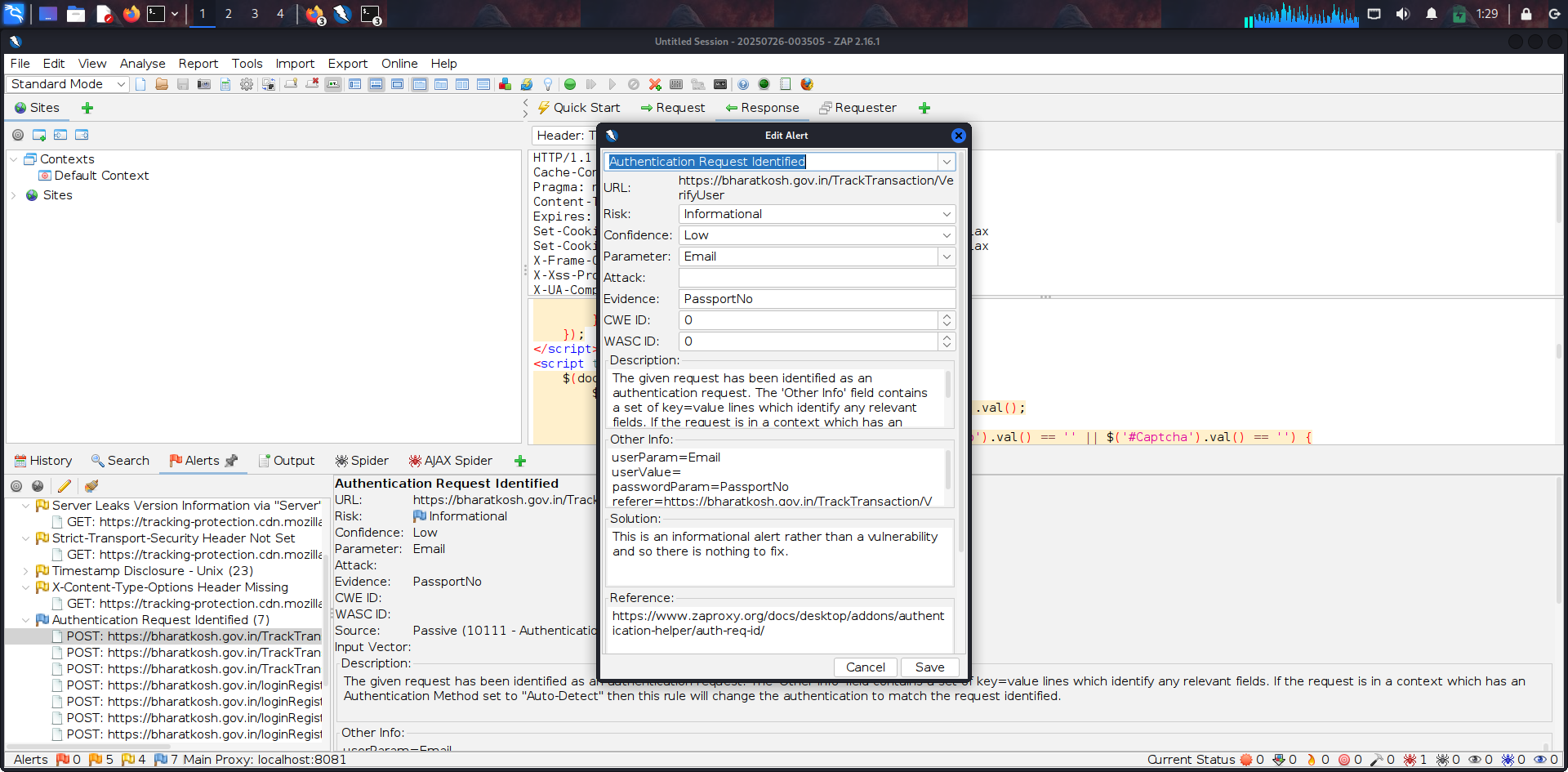
**Solution:** This is an informational alert rather than a vulnerability and so there is nothing to fix.

**Tool Used:** OWASP ZAP

**Affect URL:** <https://bharatkosh.gov.in/TrackTransaction/VerifyUse>

**Evidence:** PassportNo

Proof Of Concept 12.10



**12.11 Cookie Poisoning**

**Relative risk:** Informational

**Description:** This check looks at user-supplied input in query string parameters and POST data to identify where cookie parameters might be controlled. This is called a cookie poisoning attack, and becomes exploitable when an attacker can manipulate the cookie in various ways. In some cases this will not be exploitable, however, allowing URL parameters to set cookie values is generally considered a bug.

**Impact:**

Cookie poisoning involves modifying cookie values to manipulate server-side behavior. Attackers can exploit this to bypass authentication, escalate privileges, or tamper with user sessions. If cookies are not validated properly, it can lead to unauthorized access or data leakage.

.

**Solution:** Do not allow user input to control cookie names and values. If some query string parameters must be set in cookie values, be sure to filter out semicolon's that can serve as name/value pair delimiters.

**Tool Used:** OWASP ZAP

**Affect URL:** <https://bharatkosh.gov.in/NTRPHome/ChangeLanguage?lang=en>

### CONFIDENTIAL DOCUMENT

Proof of Concept 12.11

### 

### 

**12.12** Information Disclosure - Suspicious Comment

**Relative risk:** Information

**Description:** The response appears to contain suspicious comments which may help an attacker.

**Impact:**

Shows your authentication endpoints are properly detectable (good for automated testing)

- Confirms ZAP is correctly identifying your auth flow for potential future tests

.

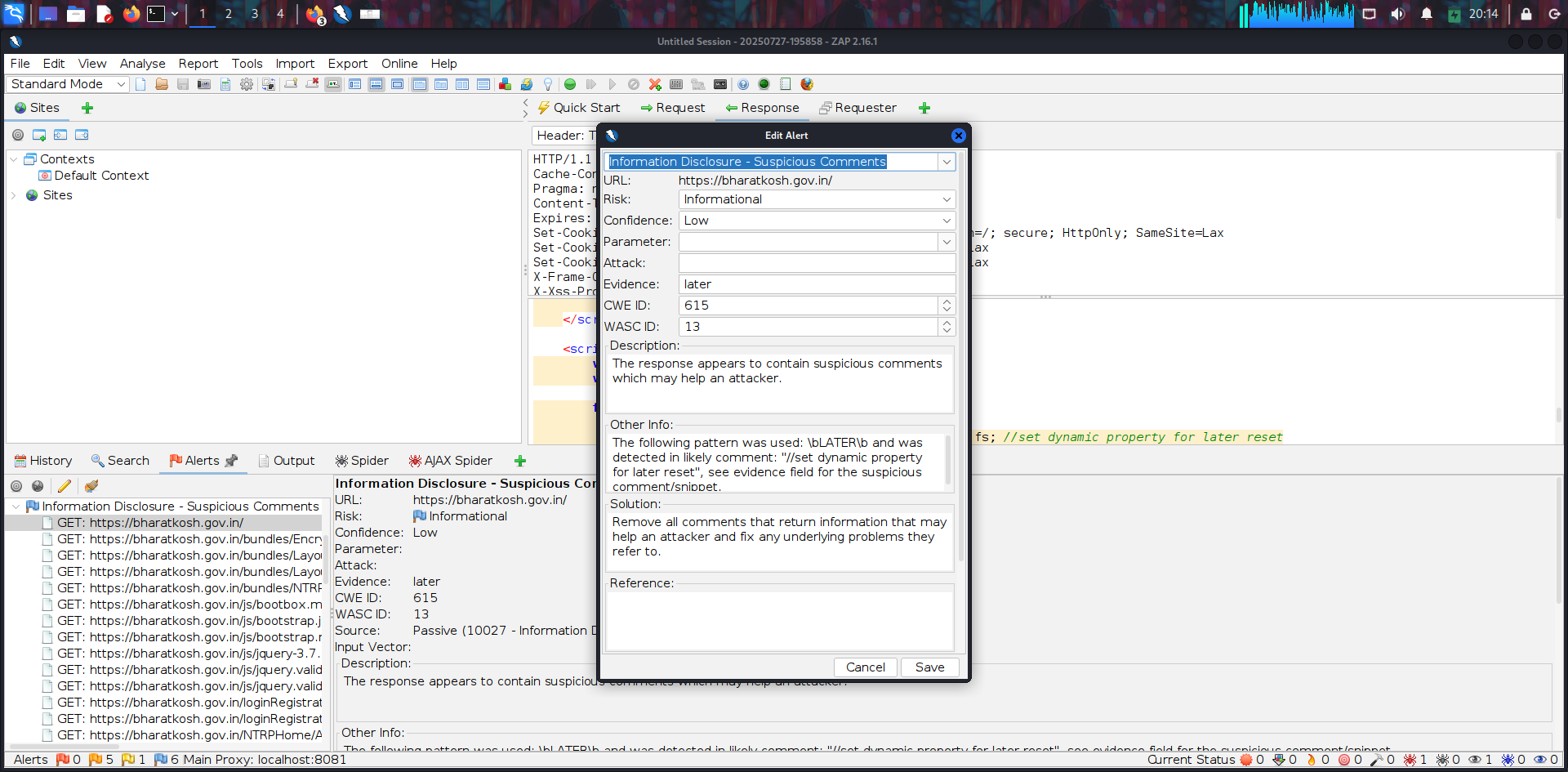
**Solution:** Remove all comments that return information that may help an attacker and fix any underlying problems they

refer to..

**Tool Used:** OWASP ZAP

**Affect URL :-** [**https://bharatkosh.gov.in/**](https://bharatkosh.gov.in/)

Proof Of Concept 12.12



**12.13** Session Management Response Identified

**Relative risk:** Information

**Description:** The given response has been identified as containing a session management token. The 'Other Info' field

contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect" then this rule will change the session management to use the tokens identified.

**Impact:**

This finding highlights potential caching behavior that could expose sensitive data, but does not confirm an

actual vulnerability.

**Solution:**

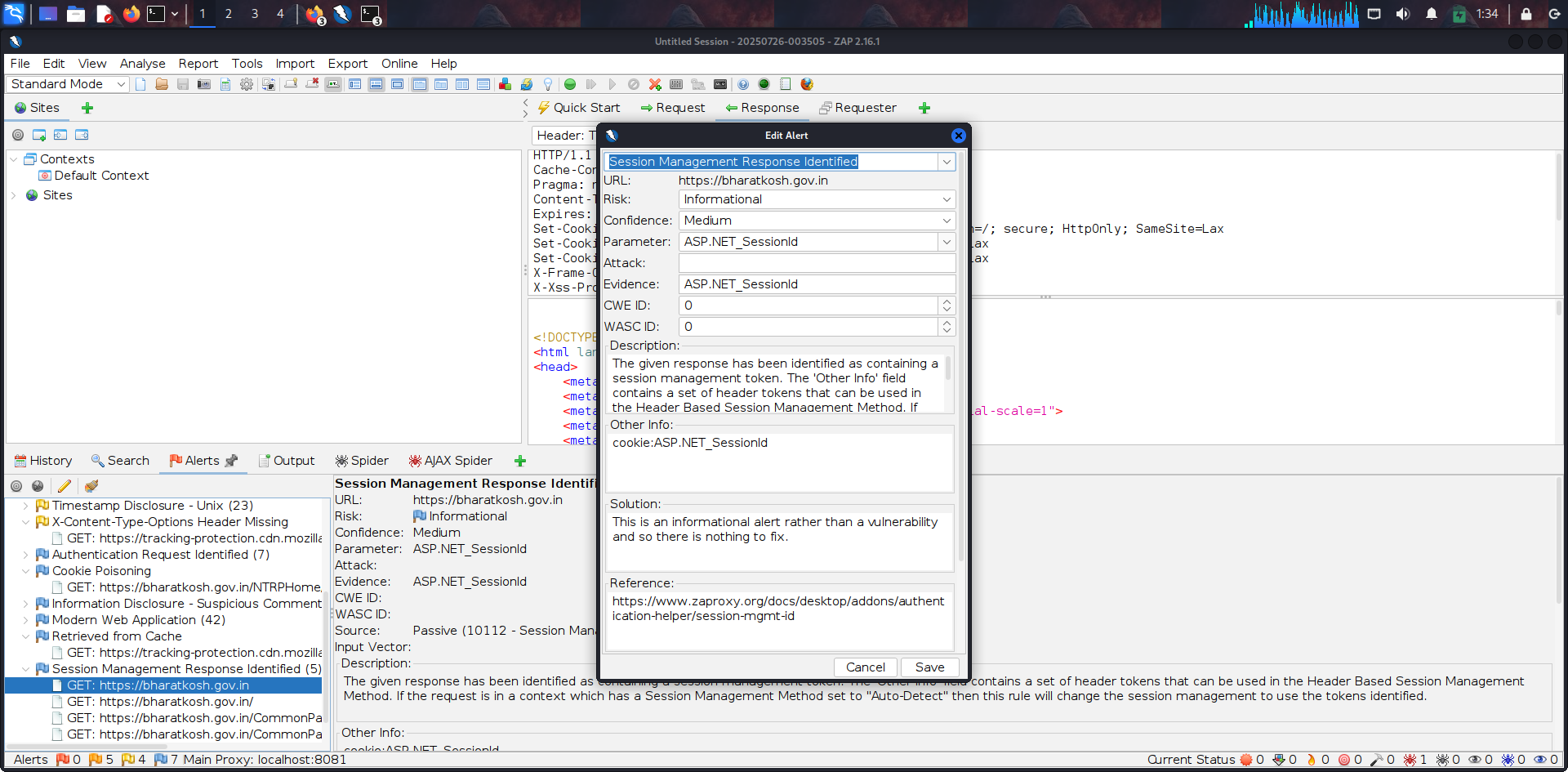
This is an informational alert rather than a vulnerability and so there is nothing to fix.

**Tool Used:** OWASP ZAP

**Affect URL:** <https://bharatkosh.gov.in/>

**Evidence:** ASP.NET\_SessionId

Proof Of Concept 12.13



# High-Level Recommendations

###### Taking into consideration all issues that have been discovered, we highly recommend to:

* + - Conduct current vs. future IT/Security program review
    - Conduct Static code analysis for codebase
    - Establish Secure SDLC best practices, assign Security Engineer to a project to monthly review code, conduct SAST & DAST security testing
    - Review Architecture of application
    - Deploy Web Application Firewall solution to detect any malicious manipulations
    - Continuously monitor logs for anomalies to detect abnormal behavior and fraud transactions. Dedicate security operations engineer to this task
    - Implement Patch Management procedures for whole IT infrastructure and endpoints of employees and developers
    - Continuously Patch production and development environments and systems on regular bases with latest releases and security updates
    - Conduct annual Penetration test and quarterly Vulnerability Scanning against internal and external environment
    - Develop and Conduct Security Awareness training for employees and developers
    - Develop Incident Response Plan in case of Data breach or security incidents
    - Analyze risks for key assets and resources
    - Update codebase to conduct verification and sanitization of user input on both, client and server side
    - Use only encrypted channels for communications
    - Do not send any unnecessary data in requests and cookies
    - Improve server and application configuration to meet security best practices