Ethnio Pen Test - September 2019 Penetration Test Report

https://ethn.io/rest/screeners

TEST PERIOD	STATUS
Sep 13, 2019 → Sep 27, 2019	Final

TEST PERFORMED BY

 David Gouveia — Lead

 Martino Sani — Pentester

Cobalt

Contents

Executive Summary

A black box penetration test of the Ethnio web application was conducted in order to assess its risk posture and identify security issues that could negatively affect Ethnio's data, systems, or reputation. The scope of the assessment covered Ethnio Web application and included credentials for various levels of privilege within the application(s). The pentest was conducted by 2 pentester(s) between Sep 13, 2019 and Sep 27, 2019.

This penetration test was a manual assessment of the security of the application's functionality, business logic, and vulnerabilities such as those catalogued in the OWASP Top 10. The assessment also included a review of security controls and requirements listed in the OWASP Application Security Verification Standard (ASVS). The pentesters leverage tools to facilitate their work, however, the majority of the assessment involves manual analysis.

The pentesters identified 1 High, 15 Medium, and 4 Low risk vulnerabilities.

The pentest revealed that the web application presents a few problems most of which are related to insufficient user input validation mechanisms. Most of these vulnerabilities resulted in potential Cross-Site Scripting attacks and Remote Code Execution, which could be used to completely compromise data and its integrity as well as 3rd party dependencies such as Paypal which could result in severe financial loss. Significant findings include:

- Stored Cross-Site Scripting via multiple vectors
- Remote Code Execution via malicious image upload
- Arbitrary File Uploads
- Server-Side Denial of Service

Specific recommendations are provided for each finding. As a whole, the recommendations indicate gaps that can be addressed by improvements to the validation of user inputs and the implementation of passive security addons such as the security headers.

Scope of Work

Coverage

This penetration test was a manual assessment of the security of the app's functionality, business logic, and vulnerabilities such as those catalogued in the OWASP Top 10. The assessment also included a review of security controls and requirements listed in the OWASP Application Security Verification Standard (ASVS). The pentesters conduct manual analysis assisted by tools.

The team had access to authenticated users, enabling them to test security controls across roles and permissions. This included attempting "vertical" privilege escalation (access to information not authorized within the container/project) and "horizontal" privilege escalation (access to information in other containers/projects without authorization).

The following is a brief summary of the main tests performed on the Web Application:

- Authenticated user testing for session and authentication issues
- Authorization testing for privilege escalation and access control issues
- Input injection tests (SQL injection, XSS, and others)
- Platform configuration and infrastructure tests
- OWASP Top 10 testing

The following is a brief summary of the main tests performed on the API:

- Authenticated endpoint testing for missing access control issues

- Authorization testing for privilege escalation and access control issues
- Input injection tests (SQL injection, XSS, and others)
- OWASP Top 10 testing

Below is the summary of methodologies used to assess security at mentioned endpoints:

Inventory of web service endpoints:

We inventoried the calls made by Ethnio during all user activities. We then proceeded to analyze requests and responses to observe underlying technology and any possible vulnerabilities.

Manual and automated fuzzing of web service endpoints:

We proceeded to reverse engineer the endpoints and perform modified calls using manual and automated methods. We attempted the following:

- Parameter manipulation (adding / modifying parameters to perform new functions, horizontal privilege escalation, etc.)
- Code injection (SQL injection attempts, Template injection, Cross Site Scripting attacks, etc.).
- XML External Entity attacks. Observation: No endpoints accepting XML were found.
- Server-Side Request Forgery (attempt to reach internal areas of the infrastructure/server).
- Perform state-changing actions via Cross Site Request Forgery attacks

- Client-side poisoning manipulation of data (localStorage, webSQL, cookies).
- Remote Code Execution via common attack vectors (e.g., server-side unsafe file management of user-controlled assets).
- Disclosure of sensitive information

Test Cases that successfully thwarted exploitation

From a security perspective, several common attack patterns were successfully blocked. For instance, Cross-site request forgery attacks were not possible due to a solid combination of request tokens with other factors (e.g . non-use of POST methods or non-standard content types which can trigger the CORS policy). Common injection attacks such as SQL or template injection methods were also prevented as no positive response was obtained either by manually testing the input fields or by using automated tools such as SQLmap or TPLmap. As for insecure direct object references (commonly known as IDOR), it was not possible to disclose or set internal data, nor perform lateral movements which could leak or change information from other accounts. The authorization mechanism in place seemed to provide good protection. Other less common attacks on user-controlled inputs were also tested. Data based on cookies, the local storage , webSQL or even HTTP headers was also assessed. Fuzzing these inputs didn't yield any relevant results from a security perspective.

Target description

Application: Ethnio

Environment: Production

Assumptions/Constraints

All the data exchanged and connections made to any Ethnio component (running in different subdomains) were considered in scope.

Methodology

The test was done according to penetration testing best practices. The flow from start to finish is listed below.

Pre Engagement

- Scoping
- Customer
- Documentation
- Information
- Discovery

Penetration Testing

- Tool assisted assessment
- Manual assessment of OWASP top 10 & business logic
- Exploitation
- Risk analysis
- Reporting

Post Engagement

- Prioritized remediation
- Best practice support
- Re-testing

Risk Factors

Each finding is assigned two factors to measure its risk. Factors are measured on a scale of 1 (very low) through 5 (very high).

Impact

This indicates the finding's effect on technical and business operations. It covers aspects such as the confidentiality, integrity, and availability of data or systems; and financial or reputational loss.

Likelihood

This indicates the finding's potential for exploitation. It takes into account aspects such as skill level required of an attacker and relative ease of exploitation.

Criticality Definitions

Findings are grouped into three criticality levels based on their risk as calculated by their business impact and likelihood of occurrence, `risk = impact * likelihood`. This follows the OWASP Risk Rating Methodology.

High

Vulnerabilities with a high or greater business impact and high or greater likelihood are considered High severity. Risk score minimum 16.

Medium

Vulnerabilities with a medium business impact and likelihood are considered Medium severity. This also includes vulnerabilities that have either very high business impact combined with a low likelihood or have a low business impact combined with a very high likelihood. Risk score between 5 and 15.

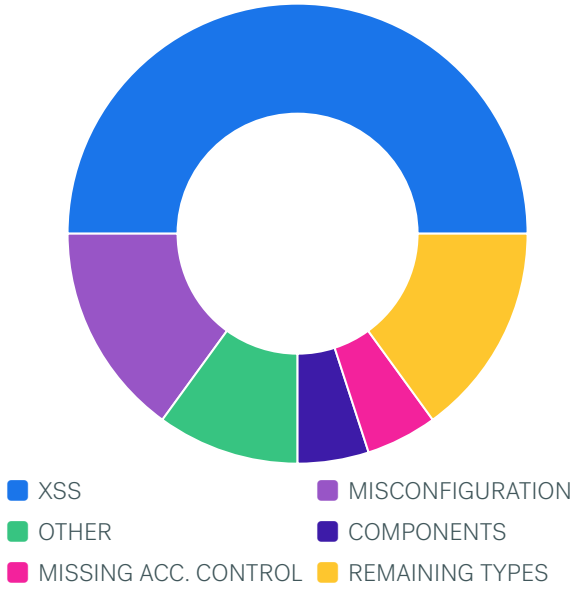
Low

Vulnerabilities that have either a very low business impact, maximum high likelihood, or very low likelihood, maximum high business impact, are considered Low severity. Also, vulnerabilities where both business impact and likelihood are low are considered Low severity. Risk score 1 through 4.

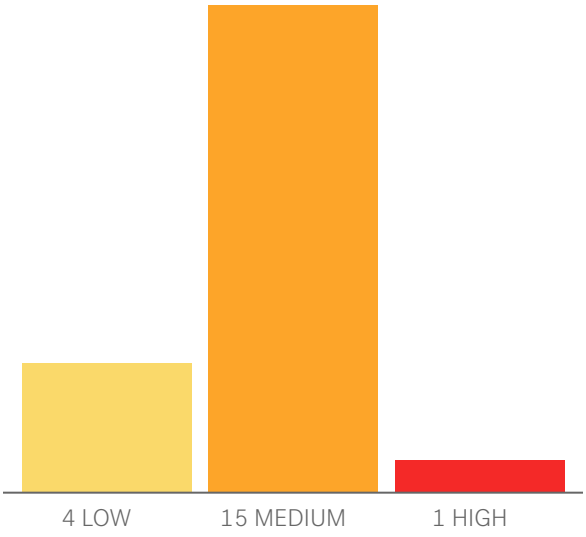
Summary of Findings

The following charts group discovered vulnerabilities by OWASP vulnerability type, and by overall estimated severity.

BY VULNERABILITY TYPE



BY CRITICALITY



Analysis

The results of the pentest indicate that there is a clear trend regarding improper validation of user inputs that get injected in the DOM. These issues may allow an attacker to perform arbitrary actions on behalf of the victims via cross-site scripting attacks. Other user inputs such as file uploads corroborate this trend as their content is not being inspected which can trigger multiple attacks such as Remote Code Execution, Cross-site scripting or simply store malware remotely.

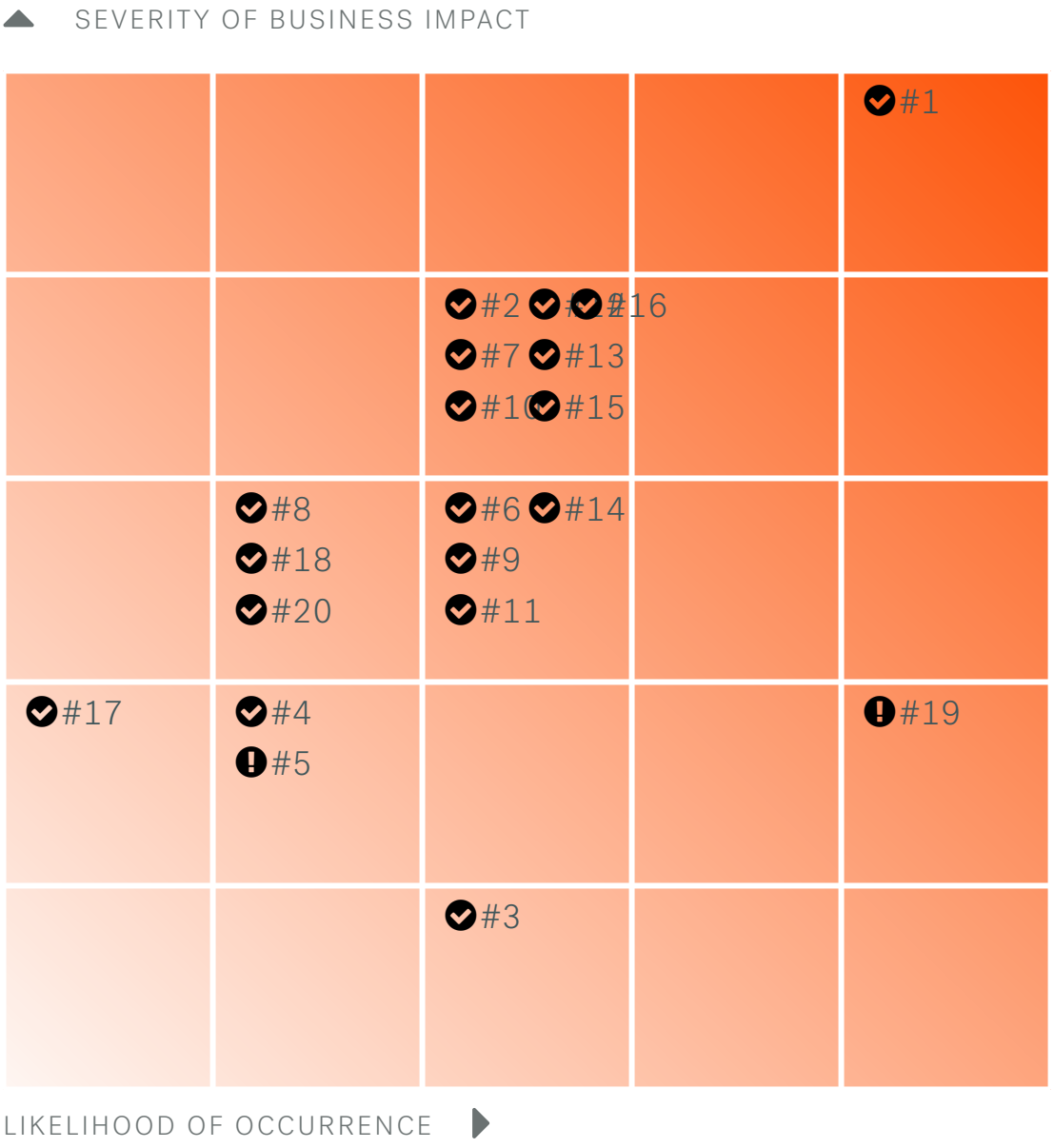
Open Ports and Services

This section documents all open ports and services identified for the target as well as any potential action needed.

Port	Service and Version	Action needed (if any)
80	http web server	none
443	https web server	none

General Risk Profile

The chart below summarizes vulnerabilities according to business impact and likelihood, increasing to the top right.



Recommendations

Based on the findings the following remediation is suggested:

1. Improve input validation methods. For data that is passed to DOM, many javascript frameworks already try to address this issue (e.g. reactjs). As an alternative, tools like DOMPurify can also be very handy (https://github.com/cure53/DOMPurify).
2. Validate all the uploads. For instance, create a whitelist of allowed file extensions and ensure that their content does match the extension provided.
3. Use the browser security features whenever possible, this is particularly valid for HTTP headers and flags that can harden the overall security with little effort.
4. Keep software up to date and do not disclose details that show the underlying tech stack.

Post-Test Remediation

As of the conclusion of this document, the following mitigations have been implemented for the identified vulnerabilities.

FINDING	LIKELIHOOD / IMPACT	STATE	RETESTED
#PT2325_1	Very High / Very High	Fixed	Sep 16
#PT2325_2	Medium / High	Fixed	Sep 16
#PT2325_3	Medium / Very Low	Fixed	Sep 17
#PT2325_4	Low / Low	Fixed	Sep 17
#PT2325_5	Low / Low	Pending Fix	
#PT2325_6	Medium / Medium	Fixed	Sep 18
#PT2325_7	Medium / High	Fixed	Sep 17

FINDING	LIKELIHOOD / IMPACT	STATE	RETESTED
#PT2325_8	Low / Medium	Fixed	Sep 16
#PT2325_9	Medium / Medium	Fixed	Sep 17
#PT2325_10	Medium / High	Fixed	Sep 16
#PT2325_11	Medium / Medium	Fixed	Sep 20
#PT2325_12	Medium / High	Fixed	Sep 16
#PT2325_13	Medium / High	Fixed	Sep 16
#PT2325_14	Medium / Medium	Fixed	Sep 17
#PT2325_15	Medium / High	Fixed	Sep 16

FINDING	LIKELIHOOD / IMPACT	STATE	RETESTED
#PT2325_16	Medium / High	Fixed	Sep 18
#PT2325_17	Very Low / Low	Fixed	Sep 17
#PT2325_18	Low / Medium	Fixed	Sep 18
#PT2325_19	Very High / Low	Pending Fix	
#PT2325_20	Low / Medium	Fixed	Sep 20

Terms

Please note that it is impossible to test networks, information systems and people for every potential security vulnerability. This report does not form a guarantee that your assets are secure from all threats. The tests performed and their resulting issues are only from the point of view of Cobalt Labs. Cobalt Labs is unable to ensure or guarantee that your assets are completely safe from every form of attack. With the ever-changing environment of information technology, tests performed will exclude vulnerabilities in software or systems that are unknown at the time of the penetration test.

Appendix A - Finding Details

David G.
(zatarra)

Remote Code Execution via profile picture upload

#PT2325_1

Valid High · Remote Code Execution (RCE) · Sep 13, 2019

DESCRIPTION

During the pentest it was possible to identify a Remote Code Execution vulnerability related to an Image Management tool named ImageMagick. This tool/library allows several filetypes to be handled, including PostScript. When this filetype is not disabled by default, it can be. used to perform some attacks which can result in remote code execution.

URL

https://ethn.io/users/1383/edit

POC

a) Login to the service and open the profile details (e.g. use the direct link above).
b) Upload a new image. This image has to contain a malicious payload. As a reference, you can you the sample attached to this finding. By default it creates a /tmp/test file but feel free to edit its content.
c) After uploading the malicious image, the commands will be executed on the server side.

CRITICALITY

An attacker can use this vector to get total control of the servers, including customer data or credentials to access other services.

SUGGESTED FIX	Disable the PostScript filters on ImageMagick. For more details please check the following guide: https://imagemagick.org/script/security-policy.php

David G.
(zatarra)

Stored Cross Site Scripting via Rich Text Editor #PT2325_2

Valid Medium · Cross-Site Scripting (XSS) · Sep 13, 2019

DESCRIPTION

During the pentest, it was possible to identify an issue regarding the Rich Text Editors used across the site to provide text formatting capabilities to multiple text fields. The way that the text editors are used, allow dangerous payloads to be passed to the server which can be used to trigger XSS vulnerabilities.

URL

https://ethn.io/rest/screeners/29464/schedules/confirmation_email

POC

a) login to the service and open the URL mentioned above.
b) Edit the text and make sure you are able to save the request that is generated when the new data gets saved. You can do this using a proxy tool such as Burp or even using the Browser's development tools.
c) Edit the payload, add some javascript to it and replay it. For this PoC, I've added a simple payload URL encoded: E%3Cimg+src%3Dx+onerror%3Dalert()%3E

d) Preview the changes. An alert box will be triggered.

Please check the image attached to this finding as it can make the explanation more clear.

CRITICALITY

An attacker might use this vector to perform arbitrary actions on behalf of the victims

SUGGESTED FIX	Improve the rich text editor and ensure that no javascript can be processed. These protections should be server-side based, not client-side. OWASP provides a good guide regarding XSS prevention. Feel free to check it here: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html

David G.
(zatarra)

Username enumeration via password recovery workflow #PT2325_3

Valid Low · Misconfiguration · Sep 13, 2019

DESCRIPTION	During the pentest it was possible to enumerate valid users by abusing the password recovery workflow. Currently, the service is disclosing whether or not a valid user is found which can be used by an attacker to perform targeted actions.
URL	https://ethn.io/password/new
POC	a) Open the URL mentioned above. b) Use an e-mail that is not registered. The system will show a message stating that the e-mail is not valid. c) Repeat step b, this time using a valid e-mail. The system will show a message stating that a recovery e-mail was sent.
CRITICALITY	An attacker might use this vector to enumerate valid e-mails which can be later used to perform more targetted attacks (e.g. bruteforce attempt only on valid accounts)
SUGGESTED FIX	Display a generic message for both cases stating that "if an account was found, then an e-mail will be sent with details for the recovery process"

David G.
(zatarra)

Missing multiple security headers #PT2325_4

Valid Low · Misconfiguration · Sep 13, 2019

DESCRIPTION	During the pentest it was possible to identify several security headers that can be used to increase the overall security of the service.
URL	https://securityheaders.com/?q=http%3A%2F%2Fethn.io&hide=on&followRedirects=on
POC	a) To confirm that the headers are missing please open the URL mentioned above. You will find a list of HTTP headers that the webserver is sending and the ones that are currently missing as well as a brief explanation.
CRITICALITY	This issue might increase the success chance of an attack. For instance,
SUGGESTED FIX	Implement the security headers. For instance, the HSTS headers it not properly implemented. As a consequence, it might be possible to perform man in the middle attacks. Mozilla also provides good information regarding the Content Security Policy framework. Please check the following guide: https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP

David G.
(zatarra)

Vulnerable javascript libraries in use #PT2325_5

Valid Low · Components with Known Vulnerabilities · Sep 13, 2019

DESCRIPTION

Using retirejs, it was possible to identify multiple javascript frameworks that are outdated and vulnerable to some attacks. In this particular case, we have seen some JQuery (and its plugins) and AngularJS libraries which are old.

The attacks might have special requirements (e.g. use a specific feature of the framework)

URL

https://ethn.io/rest/screeners

POC

a) Login to the service and load the URL mentioned above.
b) Check the source code you will find three references to:

AngularJS 1.6.1
JQuery-ui-dialog 1.11.4
JQuery 2.2.4

Found in https://s3.amazonaws.com/ethnio-demo/assets/application-3254e71e7360e4174761c6d230dfdaab3f554873ff83bce172340954537aeed6.js

CRITICALITY

A malicious actor might use these vectors to perform multiple attacks ranging from denial of service to XSS. Until now, it was not possible to identify any use of the vulnerable features, but because this is black box scanning, it is not possible to ensure all the use cases were covered.

SUGGESTED FIX	Update the javascript frameworks to their latest versions.

David G.
(zatarra)

Stored Cross Site Scripting via Screener Name #PT2325_6

Valid Medium · Cross-Site Scripting (XSS) · Sep 13, 2019

DESCRIPTION	During the pentest it was possible to detect a vulnerability regarding the Screener Name field. As a consequence, an attacker can use this vector to execute arbitrary actions on behalf of the victims
URL	https://ethn.io/rest/screeners/29484/edit?step=design
POC	a) Login to the service and load the URL mentioned above. b) edit the screener name and set it to something. Save the request. c) edit the request and set the name to "><svg/onload=alert()> d) replay the request and reload the page. an alert will be triggered.
CRITICALITY	an attacker might use this vector to perform arbitrary actions on behalf of the victims.
SUGGESTED FIX	Validate the user inputs and ensure that data is only passed to the right context (e.g. encode user data before passing it to DOM)

David G.
(zatarra)

Stored Cross Site Scripting via data uri injection

#PT2325_7

Valid Medium · Cross-Site Scripting (XSS) · Sep 13, 2019

DESCRIPTION

Suring the pentest it was possible to identify a vulnerability regarding the standard text editor (not the rich text editors). After some tests, it was possible to inject a malicious payload in the newly created link which can be used to execute javascript commands.

URL

https://ethn.io/rest/screeners/29484/edit?step=invite

POC

a) create a new screening process or feel free to use the example above.
b) Create a new link using the builtin tool (check the attachment). Data will be automatically u
request.
c) Edit the request generated during step b and change the value of the href attribute to :

data%3Atext%2Fhtml%3Bbase64%2CPHNjcmlwdD5hbGVydCgiSGVsbG8iKTs8L3NjcmlwdD4%

d) replay the request to update data.
e) preview the invite. click on the link you've just created. It willshow an alert box.

CRITICALITY

An attacker might use this vector to execute arbitrary actions on behalf of the victims.

SUGGESTED FIX

Ensure that the href attribute can only hold a standard value (E.g. starting with http(s) protocol, or and using a regular expression to validate a domain).

David G.
(zatarra)

Stored Cross Site Scripting via `last name` parameter under team member management. #PT2325_8

Valid Medium · Cross-Site Scripting (XSS) · Sep 13, 2019

DESCRIPTION	During the pentest, it was possible to identify a Stored XSS vulnerability related to the first and last name fields of the user management system. Using a malicious payload it is possible to trigger javascript actions which can be used for malicious activities.
URL	https://ethn.io/people
POC	Login to the service and open the URL mentioned above so that you can create/invite new team membeers. a) Invite a new team member and set the first and last name to: "><svg/onload=alert(1)> b) Click on the permissions button of this new user. c) An alert box will be triggered.
CRITICALITY	An attacker might use this vector to execute arbitrary actions on behalf of the victim.
SUGGESTED FIX	Sanitize the input and do not allow arbitrary characters to be used (e.g. use a whitelist of allowed characters).

David G.
(zatarra)

Arbitrary file upload via malicious screenshot uploading feature #PT2325_9

Valid Medium · Cross-Site Scripting (XSS) · Sep 13, 2019

DESCRIPTION	Using the screenshot feature it was possible to upload files with malicious payloads that can be used to perform XSS based attacks.
URL	https://ethn.io/preview/54981
POC	a) Login to the service and create a screener (or load the one in the URL above) b) Select the upload functionality and pick a file such as a SVG (you can use the one attached to this finding as a template)
CRITICALITY	An attacker might use this vector to upload malicious files which can be used to perform XSS attacks or even to disseminate malware.
SUGGESTED FIX	Ensure that only acceptable files are uploaded (e.g. check their file extension and ensure it matches the content). It is also recommended to disable SVG uploads.

David G.
(zatarra)

Stored Cross Site Scripting via custom location parameter #PT2325_10

Valid Medium · Cross-Site Scripting (XSS) · Sep 13, 2019

DESCRIPTION	During the pentest it was possible to identify a Cross Site Scripting (XSS) vulnerability due to improper validation of the custom `location` parameter available in the targeting section of the screeners.
URL	https://ethn.io/rest/screeners/29464/edit?step=targeting
POC	a) Login to the service. b) Create a new screener, or use the one mentioned above. c) Try to add a new location (you can input a valid value here). Save the request (using the browser's developer tools or a proxy service like burp). d) modify the value using the following payload: "><svg/onload=alert()> e) replay the request and refresh the page. An alert box will be shown.
CRITICALITY	An attacker might use this vector to perform arbitrary actions on behalf of the victims.
SUGGESTED FIX	Validate the inputs. In this particular case, do not allow any other characters other than letters or specific symbols (such as hyphens).

David G.
(zatarra)

Server Side Denial of Service #PT2325_11

Valid Medium · Other · Sep 13, 2019

DESCRIPTION	During the pentest it was possible to identify an issue regarding the custom locations that can be set. In this particular case, whenever an invalid location id is passed, the feature stops responding which will prevent anyone else from using it.
URL	https://ethn.io/rest/screeners/29533/edit?step=targeting
POC	a) Login to the service and load the URL mentioned above (or any other screener) b) Select a custom location. Save the request. c) Edit the request and inject an invalid location id (e.g. add multiple zeros just like in the screenshot). Replay the request. d) The service will no longer return valid locations.
CRITICALITY	An attacker might use this vector to interfere with the service and prevent valid users from using it fully.
SUGGESTED FIX	Ensure that all invalid values are ignored/discarded.

David G.
(zatarra)

Stored Cross Site Scripting via redeem_page_header parameter injection @ Redeem Pages #PT2325_12

Valid Medium · Cross-Site Scripting (XSS) · Sep 14, 2019

DESCRIPTION	During the pentest it was possible to trigger a cross-site scripting vulnerability by injecting a malicious payload in the redeem_page_header field. The backend does not validate the user input which can be used to inject malicious content.
URL	https://ethn.io/rest/screeners/29503/incentives/customize#!#pages
POC	a) login to the service and load the url mentioned above. b) edit the page header (not the title) and add a simple payload like this: <img src=x onerror=prompt()> c) Click on the "preview" button at the bottom of the page. A prompt will be shown.
CRITICALITY	An attacker might use this vector to execute arbitrary actions on behalf of the victims.
SUGGESTED FIX	Validate the inputs and make sure they are only executed within the right context (e.g. encode data properly)

David G.
(zatarra)

Stored Cross Site Scripting via redeem_page_sub_header parameter injection @ Redeem Pages #PT2325_13

Valid Medium · Cross-Site Scripting (XSS) · Sep 14, 2019

DESCRIPTION

During the pentest it was possible to trigger a cross-site scripting vulnerability by injecting a malicious payload in the redeem_page_subheader field. The backend does not validate the user input which can be used to inject malicious content.

This issue is very similar to the previous one, except that contents get encoded before being sent to the backend. Unfortunately, this is not properly protected and an attacker can still inject the malicious payload by calling the backend directly.

URL

https://ethn.io/rest/screeners/29503/incentives/customize#!#pages

POC

a) login to the service and load the url mentioned above.
b) edit the page header (not the title) and add a simple payload like this:

<img src=x onerror=prompt()>

c) Click on the "preview" button at the bottom of the page. A prompt will be shown.

CRITICALITY

An attacker might use this vector to execute arbitrary actions on behalf of the victims.

SUGGESTED FIX	Validate the inputs and make sure they are only executed within the right context (e.g. encode data properly)

David G.
(zatarra)

Open Redirect via subdomain injection #PT2325_14

Valid Medium · Redirects and Forwards · Sep 14, 2019

DESCRIPTION	During the pentest it was possible to identify an open redirect vulnerability due to the way that subdomains are handled. In this particular case, the custom subdomain that can be set for custom redeem pages is not being properly validated. As a consequence, an attacker might pass arbitrary values appending a slash which will redirect the user to any external site, eg. "google.com/" gets translated to https://google.com/.ethn.io/redeem/preview
URL	https://ethn.io/rest/screeners/29503/incentives/customize#!#pages
POC	a) login to the service and open the URL mentioned above. b) locate the subdomain section (bottom left) and set it to google.com/ (slash included) c) Open the "Payment Methods" tab. Click on the "Preview Reward as Recipient" button. d) you will be redirected to google.com
CRITICALITY	An attacker might use this vector to perform malicious activities such as phishing campaigns.

SUGGESTED FIX	Validate the user input. In this particular case, a regular expression can be used to ensure only characters, numbers and hyphens are used.

David G.
(zatarra)

Stored Cross Site Scripting via `First Name` and `Last Name` parameters @ profile settings #PT2325_15

Valid Medium · Cross-Site Scripting (XSS) · Sep 15, 2019

DESCRIPTION

During the pentest, it was possible to identify a stored Cross-Site Scripting vulnerability related to the First Name and Last Name parameters that are available via profile settings.

Although the malicious payloads do not get reflected in the existing page, they were found to be triggered in the billing section of the website.

URL

https://ethn.io/users/1383/edit

POC

a) Login to the service and load the URL mentioned above.
b) Add a malicious payload to the first name and last name parameters such as:
<img src=x onerror=alert()>
c) Save it.
d) Navigate to the billing section of the website (https://ethn.io/users/1383/billing)
e) Select one of the active invoices and click on "edit".

The payloads will be executed.

CRITICALITY

An attacker might use this vector to execute arbitrary actions on behalf of the victims.

SUGGESTED FIX	Validate user inputs and ensure that they are only passed to the right contexts (e.g. encode data before passing it to DOM)

David G.
(zatarra)

Stored Cross Site Scripting via Custom Targeting Filter

#PT2325_16

Valid Medium · Cross-Site Scripting (XSS) · Sep 15, 2019

DESCRIPTION	During the pentest, it was possible to identify a stored cross-site scripting via custom targeting filter. All of the fields to not validate the user input. As a consequence, an attacker might use this vector to inject malicious javascript code which can then be used to perform arbitrary actions.
URL	https://ethn.io/rest/screeners/29503/edit?step=targeting
POC	a) Login to the service and load the URL mentioned above (or load the targeting tab of any screener) b) Click on the "Add Custom filter" button. Add the following payload to any of the fields: <svg/onload=alert()> c) Save the data. d) Refresh the page. The alert box will be shown.
CRITICALITY	an attacker might use this vector to perform arbitrary actions on behalf of the victim.
SUGGESTED FIX	Validate the user input and ensure that it is only used in the right context (e.g. by encoding all of the user data before passing it to DOM).

David G.
(zatarra)

_ethnio_session without secure flag set #PT2325_17

Valid Low · Misconfiguration · Sep 17, 2019

DESCRIPTION	If the secure flag is set on a cookie, then browsers will not submit the cookie in any requests that use an unencrypted HTTP connection, thereby preventing the cookie from being trivially intercepted by an attacker monitoring network traffic. If the secure flag is not set, then the cookie will be transmitted in clear-text if the user visits any HTTP URLs within the cookie's scope. An attacker may be able to induce this event by feeding a user suitable links, either directly or via another web site. Even if the domain that issued the cookie does not host any content that is accessed over HTTP, an attacker may be able to use links of the form http://example.com:443/ to perform the same attack. In this particular case,
URL	https://ethn.io
POC	a) Login to the service. While doing so, check the response headers. You will find that the _ethnio_session will be set without the SSL flag.
CRITICALITY	An attacker might be able to retrieve a valid session chaining this vulnerability with other attacks (e.g. Man-in-the-Middle).

SUGGESTED FIX

The most important action is to enable HSTS which is already mentioned in #PT2325_4. This way, all traffic will be secured. But adding the SSL flag to all cookies is a good practice since the webserver could get misconfigured and this way the application side will always protect its data.

David G.
(zatarra)

Team Account Incentive Limits accept negative values

#PT2325_18

Valid Medium · Other · Sep 17, 2019

DESCRIPTION

During the pentest we were able to identify an issue regarding the "Team Account Incentive Limits " that can be found in the billing section of the site. The culprit is the "limit" field which can be set to negative values.

Depending on the business logic behind it, this can cause financial losses (e.g. if this value is somehow multiplied by another positive number, the result will be negative which can completely change calculations).

URL

https://ethn.io/users/1383/billing

POC

a) Login to the service and open the URL mentioned above
b) Locate the "Team Account Incentive Limits" section.
c) set the "limit" value to anything. Save the request.
d) Edit the request, change the previously set value to a negative number. Replay it.

The value is now negative.

CRITICALITY

An attacker might use this vector to cause some financial losses.

SUGGESTED FIX	Ensure that only valid values are parsed (e.g values bigger than 0 and smaller than X).

David G.
(zatarra)

Calendar details are publicly exposed #PT2325_19

Valid Medium · Missing Access Control · Sep 18, 2019

DESCRIPTION	During the pentest it was possible to identify an issue which allows any user to download all the of the screener calendars. This can be used to retrieve data regarding candidates as well as the interviewers.
URL	https://ethn.io/calendar_feed/29532/feed.ics
POC	To confirm this issue, please locad the URL mentioned above. It will disclose information regarding the screener id which includes names and email addresses.
CRITICALITY	An attacker can easily bruteforce the calendar system to retrieve all calendar events as well as identify the intervenients.
SUGGESTED FIX	As discussed in the channel, please implement a mechanism that disables bruteforce. This can be achieved either by changing the id to a randomly generated uuid, or by adding a key (e.g. a randomly generated hash)

Martino S.
(ilsani)

Username and password submitted on the query string #PT2325_20

Valid Medium · Sensitive Data Exposure · Sep 19, 2019

DESCRIPTION

It was found that when a user logs in, the application sends user credentials within the query string parameters. It isn't a critical issue, but sending such information through query string is not recommended.

Sending such information through query string is not recommended as they are saved in web server log files on the server side and they are kept in the browser history on the client side. Anyone with read access to them will be able to read and collect them and use them to log in using those credentials.

URL

https://ethn.io/rest/screeners

POC

1) Navigate to https://ethn.io/login
2) Intercept the network traffic using a web proxy (e.g. Burp Suite)
3) Perform the login
4) Observe the intercepted traffic
5) Username and password are submitted in query string

CRITICALITY

Anyone with read access to the server log files or browser history on the client side will be able to read and collect the sensitive data submitted on the query string and use them to log in using those credentials.

SUGGESTED FIX

It is recommended to avoid public exposure of private information such as username and password.
It's also recommended to send this kind of information in headers or in the body POST method but, most importantly, to be sure they are not logged in clear text.

HTTP REQUEST

```
GET /validate_user?user%5Bemail%5D=<EMAIL>&user%5Bpassword%5D=<PASSWORD
Host: ethn.io
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://ethn.io/
X-CSRF-Token: nkQSe75S2327IZoN3KmTUgVw+wtT4Ga0w6uioFdzK+QQsL/5UAt3t2kc
X-Requested-With: XMLHttpRequest
Connection: close
```