# ROBOT Attack on Attribute Authority “Back Channels”

Shannon Roddy

Security Lead

Trust & Identity, InCommon, Internet2

# Who

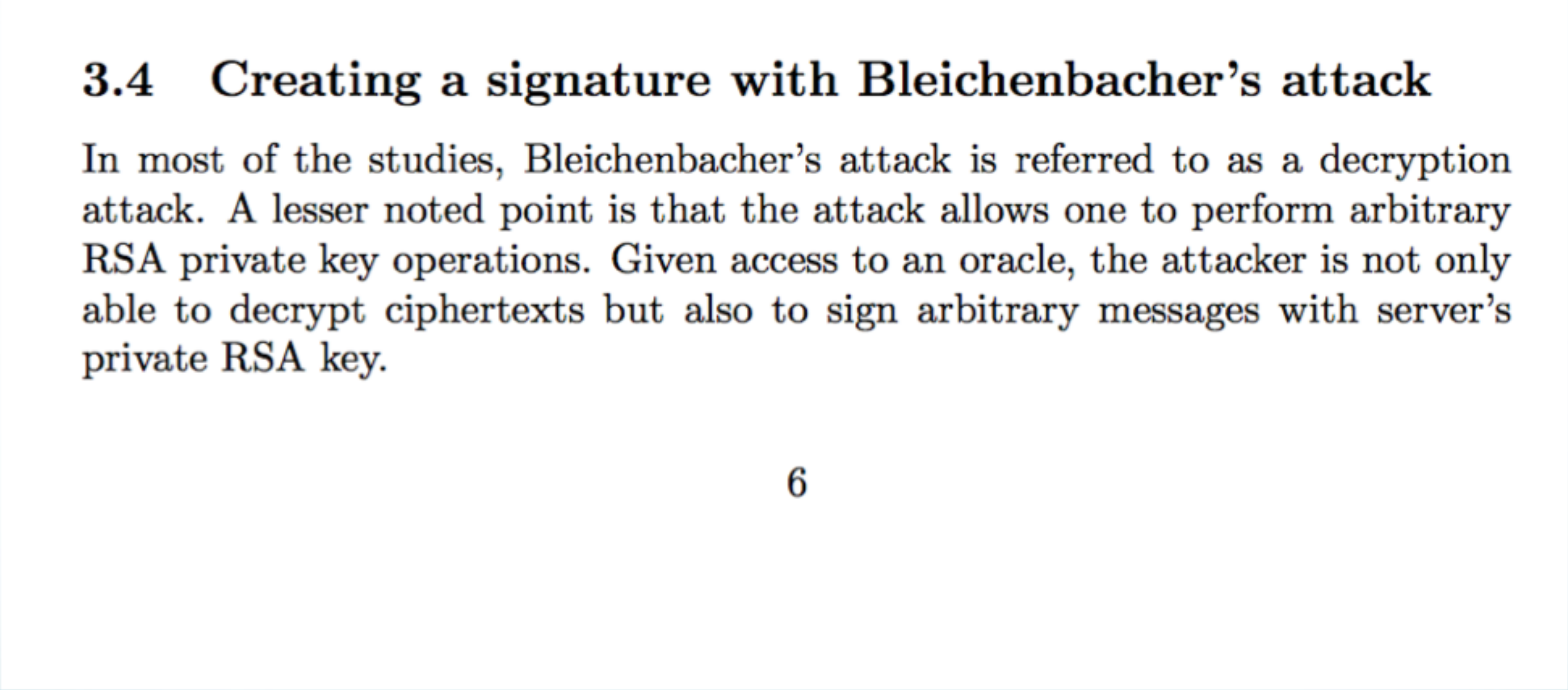
**Shannon Roddy, Security Lead, Trust & Identity**

- **Internet2 since August, 2017**
- **Previously Penn State & LIGO**

# What

- ROBOT Attack (https://robotattack.org/)
  - Revived Bleichenbacher style attacks from 1998
  - Implementation issues related to PKCS#1v1.5
  - Complicated standard & work arounds. Implementation is very difficult to get right.
- Another TLS bug/vuln. Ho-Hum, right?
  - MitM, Credential theft, offline decryption, etc.
  - But, wait…

# … There's more!



### 3.4 Creating a signature with Bleichenbacher's attack

In most of the studies, Bleichenbacher's attack is referred to as a decryption attack. A lesser noted point is that the attack allows one to perform arbitrary RSA private key operations. Given access to an oracle, the attacker is not only able to decrypt ciphertexts but also to sign arbitrary messages with server's private RSA key.

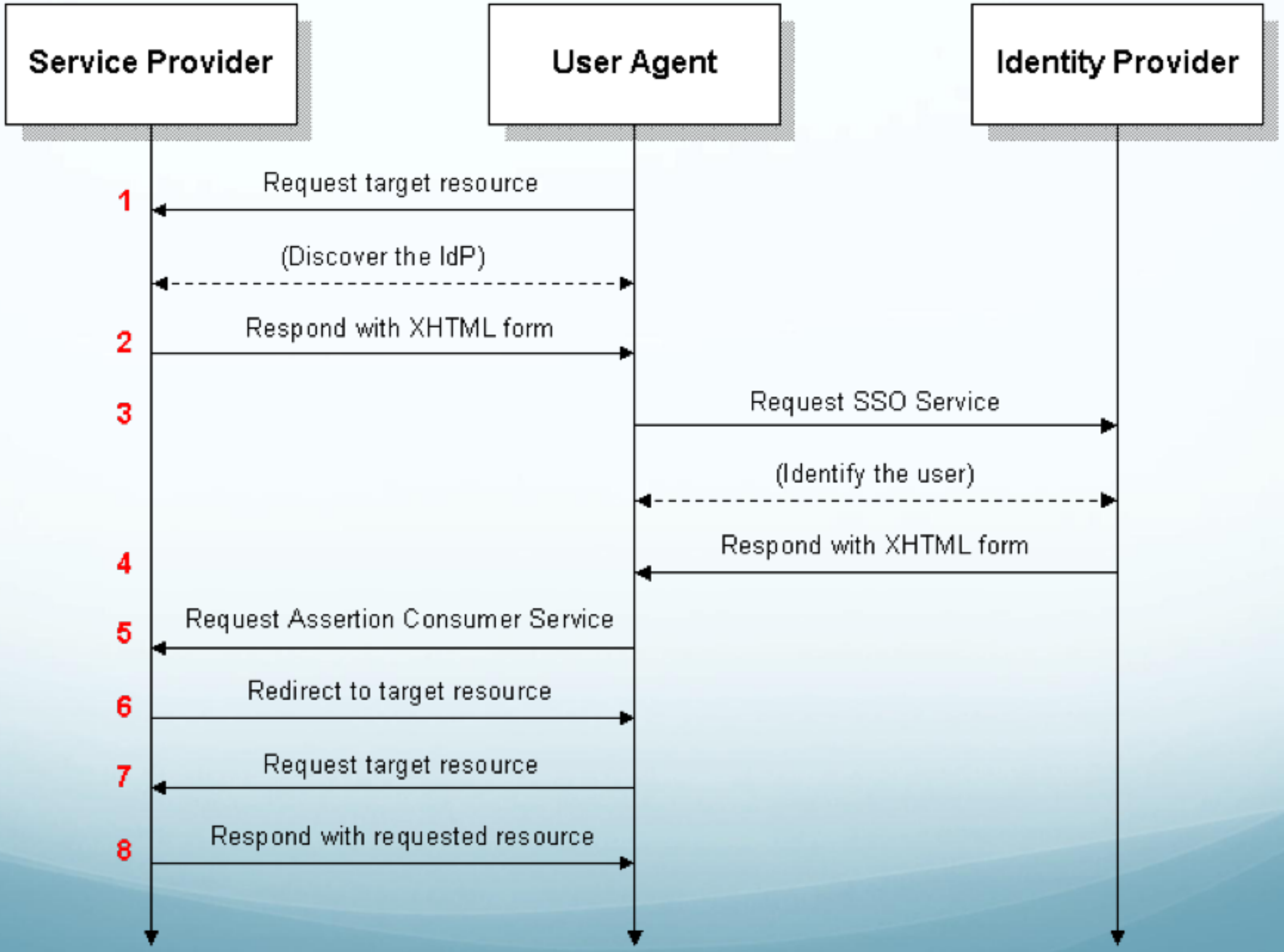
6

# What

SAML Assertions are signed by signing keys

- IdPs usually have self signed, long lived, certificate/key pairs
- Signatures from SAML signing keys are crucial in verifying validity of assertions
- Service Providers rely on these signatures applied to SAML assertions for access controls
- In a federation, trust is rooted in the signed metadata containing these certificates.

# Typical flow



Service Provider

User Agent

Identity Provider

1 Request target resource

(Discover the IdP)

2 Respond with XHTML form

3 Request SSO Service

(Identify the user)

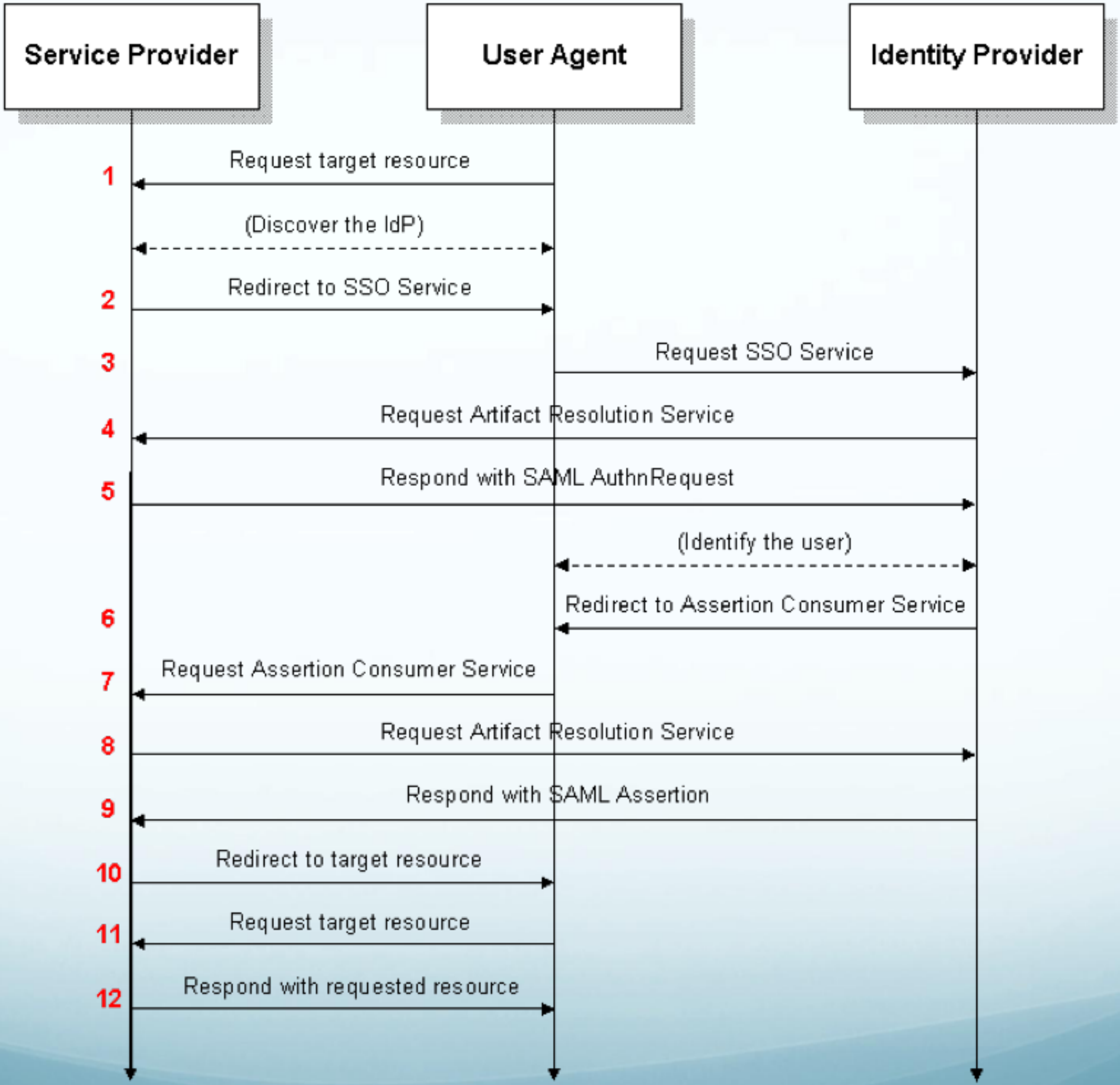
4 Respond with XHTML form

5 Request Assertion Consumer Service

6 Redirect to target resource

7 Request target resource

8 Respond with requested resource



Service Provider

User Agent

Identity Provider

1 Request target resource

(Discover the IdP)

2 Redirect to SSO Service

3 Request SSO Service

4 Request Artifact Resolution Service

5 Respond with SAML AuthnRequest

(Identify the user)

6 Redirect to Assertion Consumer Service

7 Request Assertion Consumer Service

8 Request Artifact Resolution Service

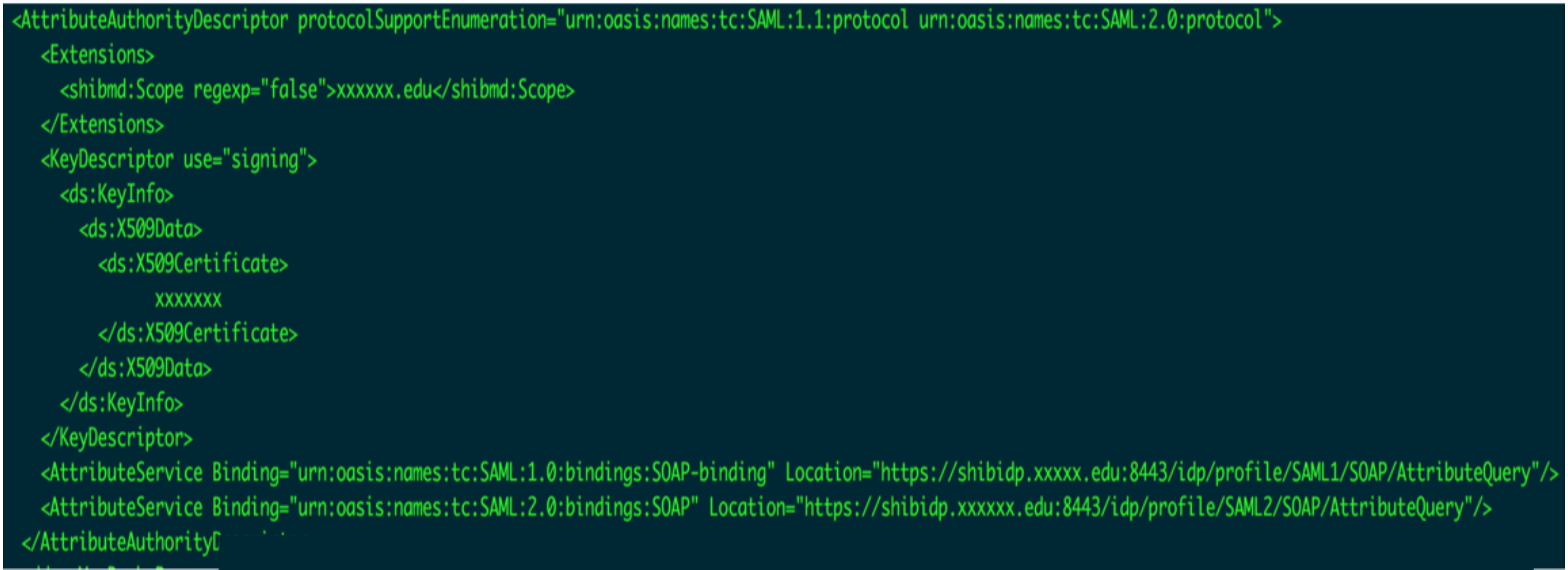
9 Respond with SAML Assertion

10 Redirect to target resource

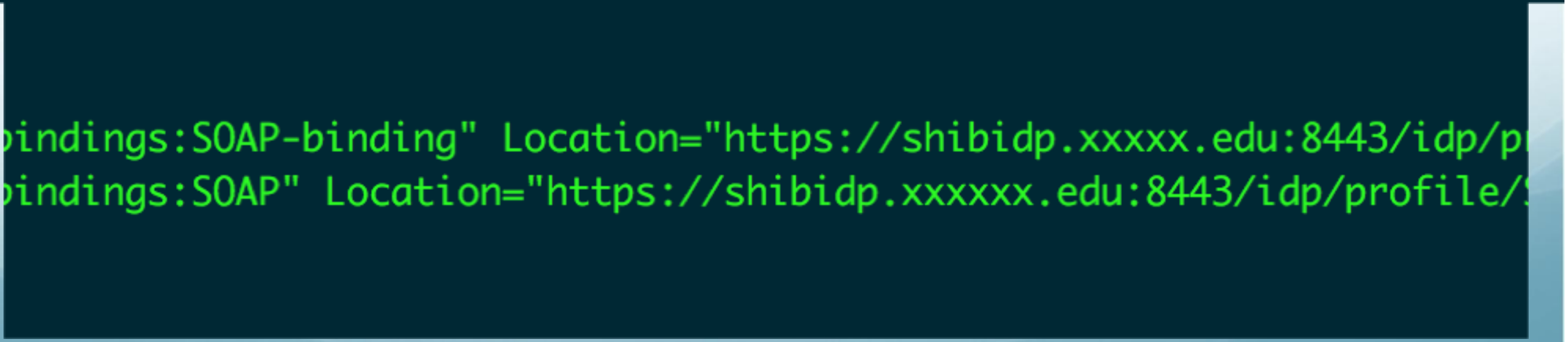
11 Request target resource

12 Respond with requested resource

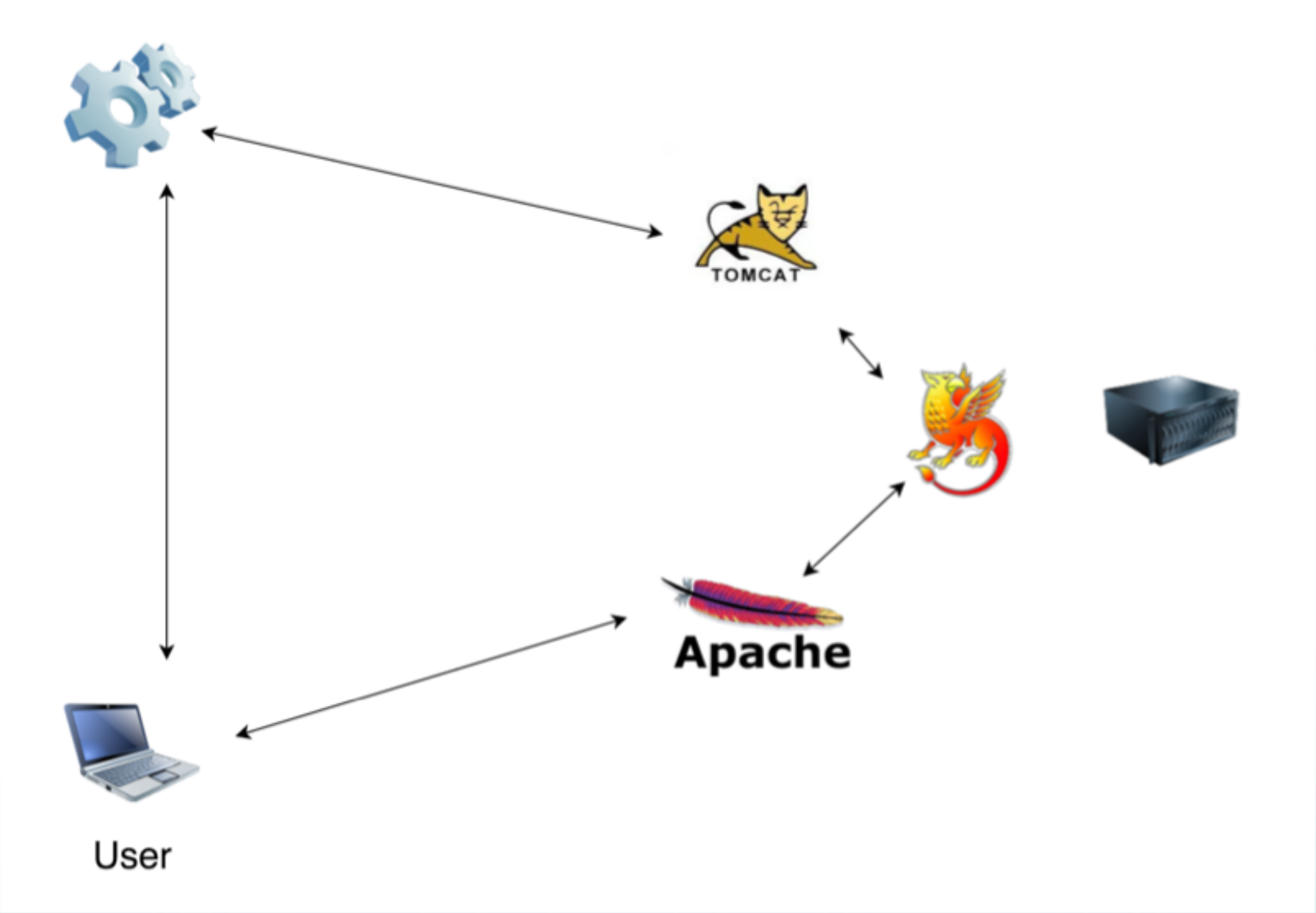
# What

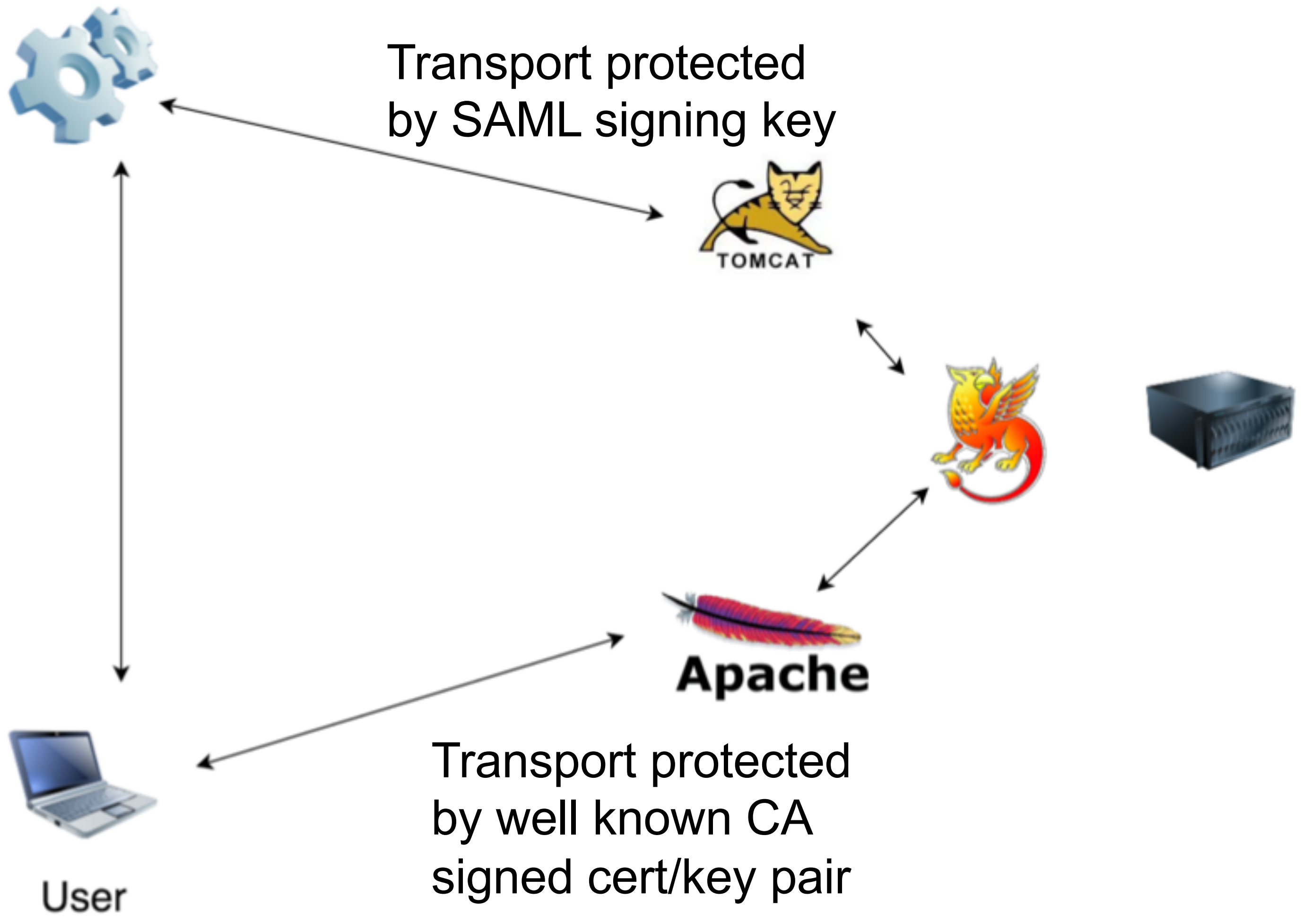


```
<AttributeAuthorityDescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:1.1:protocol urn:oasis:names:tc:SAML:2.0:protocol">
   <Extensions>
     <shibmd:Scope regexp="false">xxxxxx.edu</shibmd:Scope>
   </Extensions>
   <KeyDescriptor use="signing">
     <ds:KeyInfo>
       <ds:X509Data>
         <ds:X509Certificate>
               xxxxxxx
         </ds:X509Certificate>
       </ds:X509Data>
     </ds:KeyInfo>
   </KeyDescriptor>
   <AttributeService Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding" Location="https://shibidp.xxxxx.edu:8443/idp/profile/SAML1/SOAP/AttributeQuery"/>
   <AttributeService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP" Location="https://shibidp.xxxxxx.edu:8443/idp/profile/SAML2/SOAP/AttributeQuery"/>
 </AttributeAuthorityD
```



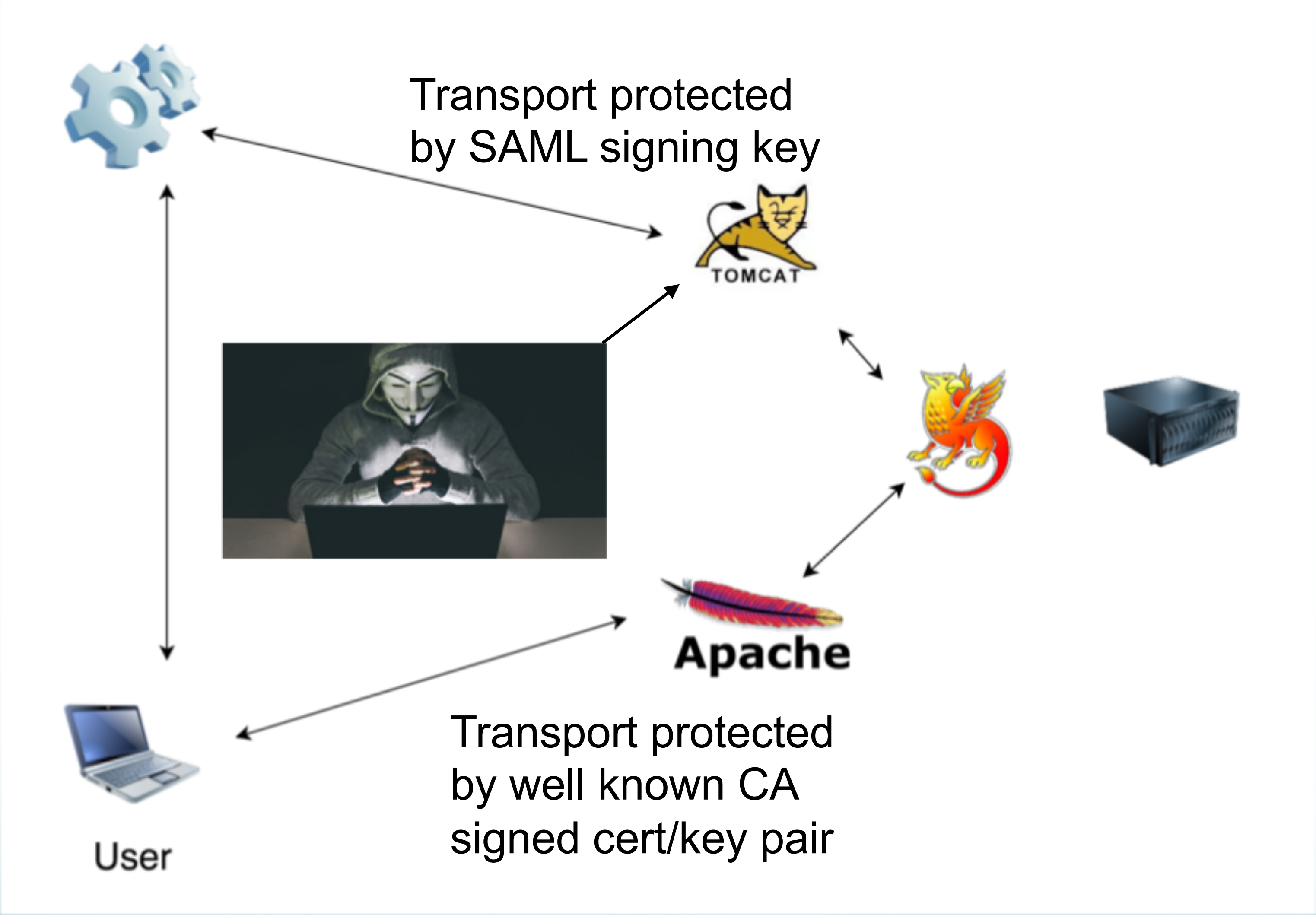
```
bindings:SOAP-binding" Location="https://shibidp.xxxxx.edu:8443/idp/pr
bindings:SOAP" Location="https://shibidp.xxxxxx.edu:8443/idp/profile/
```


TOMCAT
Apache
User



Transport protected
by SAML signing key

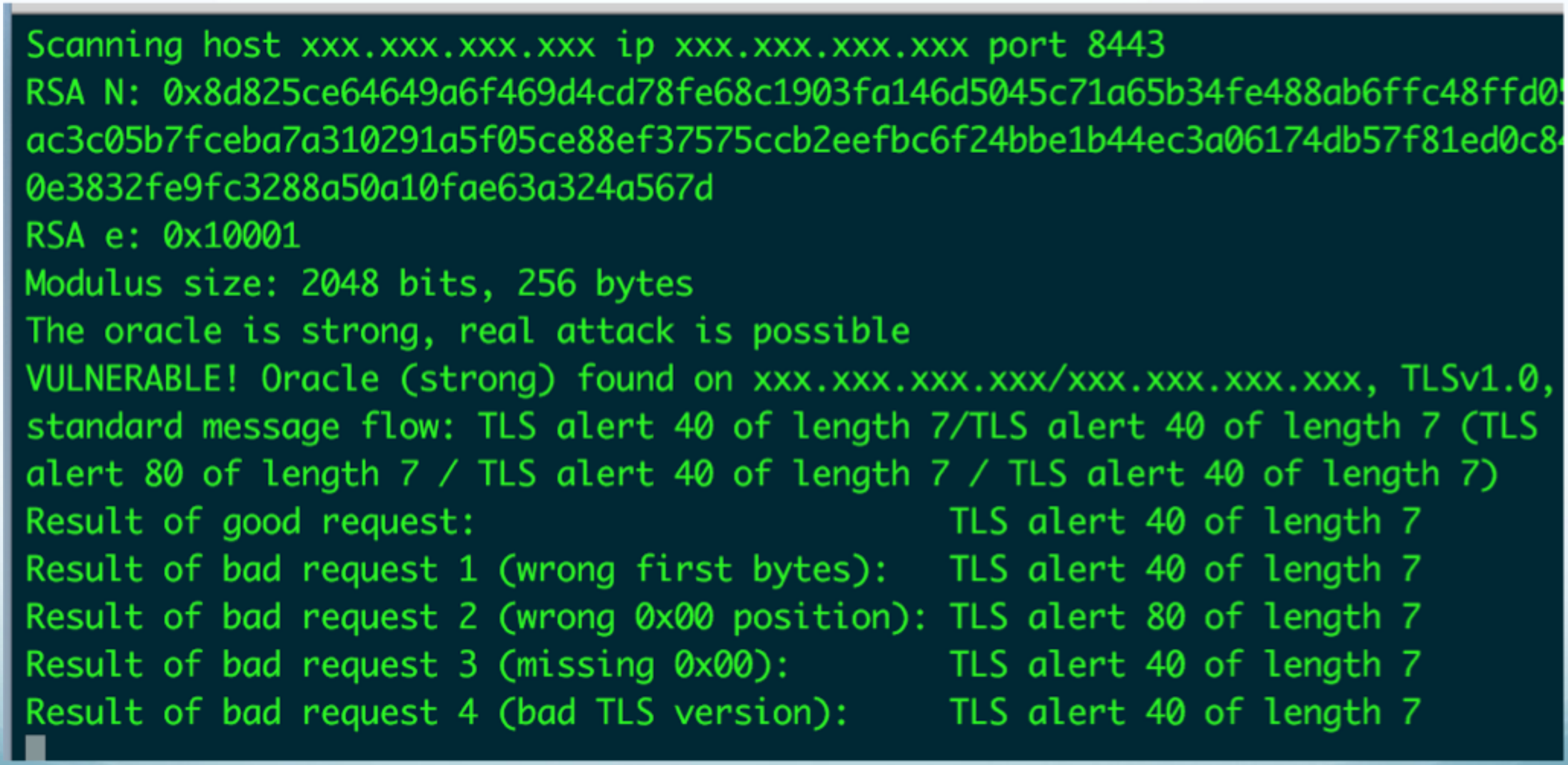
Transport protected
by well known CA
signed cert/key pair



Transport protected by SAML signing key

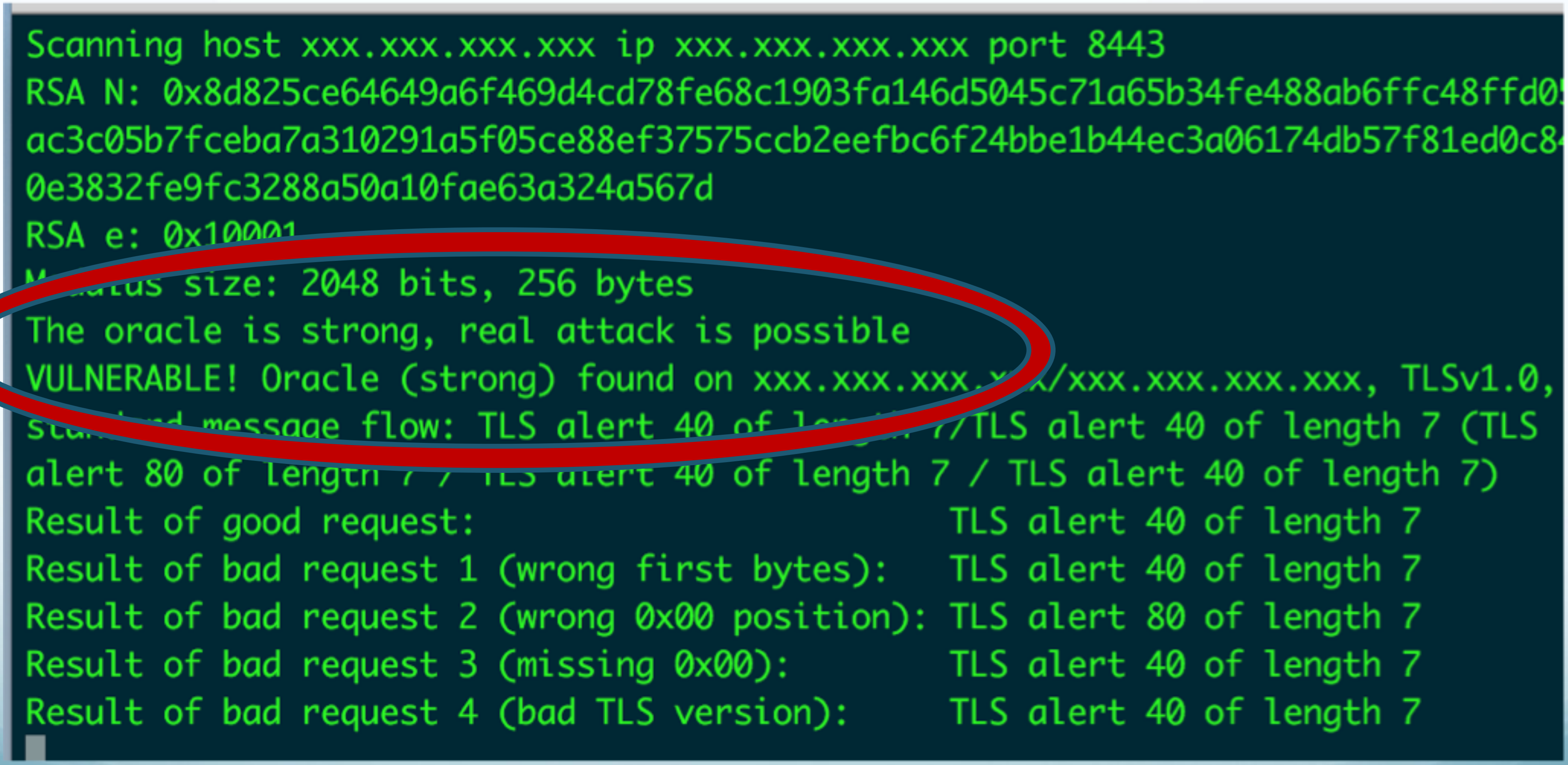
Transport protected by well known CA signed cert/key pair

# Actual vulnerable back channel
IP masked



```
Scanning host xxx.xxx.xxx.xxx ip xxx.xxx.xxx.xxx port 8443
RSA N: 0x8d825ce64649a6f469d4cd78fe68c1903fa146d5045c71a65b34fe488ab6ffc48ffd0
ac3c05b7fceba7a310291a5f05ce88ef37575ccb2eefbc6f24bbe1b44ec3a06174db57f81ed0c8
0e3832fe9fc3288a50a10fae63a324a567d
RSA e: 0x10001
Modulus size: 2048 bits, 256 bytes
The oracle is strong, real attack is possible
VULNERABLE! Oracle (strong) found on xxx.xxx.xxx.xxx/xxx.xxx.xxx.xxx, TLSv1.0,
standard message flow: TLS alert 40 of length 7/TLS alert 40 of length 7 (TLS
alert 80 of length 7 / TLS alert 40 of length 7 / TLS alert 40 of length 7)
Result of good request:                        TLS alert 40 of length 7
Result of bad request 1 (wrong first bytes):   TLS alert 40 of length 7
Result of bad request 2 (wrong 0x00 position): TLS alert 80 of length 7
Result of bad request 3 (missing 0x00):        TLS alert 40 of length 7
Result of bad request 4 (bad TLS version):     TLS alert 40 of length 7
```

# Actual vulnerable back channel
# IP masked



```
Scanning host xxx.xxx.xxx.xxx ip xxx.xxx.xxx.xxx port 8443
RSA N: 0x8d825ce64649a6f469d4cd78fe68c1903fa146d5045c71a65b34fe488ab6ffc48ffd0
ac3c05b7fceba7a310291a5f05ce88ef37575ccb2eefbc6f24bbe1b44ec3a06174db57f81ed0c8
0e3832fe9fc3288a50a10fae63a324a567d
RSA e: 0x10001
Modulus size: 2048 bits, 256 bytes
The oracle is strong, real attack is possible
VULNERABLE! Oracle (strong) found on xxx.xxx.xxx.xxx/xxx.xxx.xxx.xxx, TLSv1.0,
standard message flow: TLS alert 40 of length 7/TLS alert 40 of length 7 (TLS
alert 80 of length 7 / TLS alert 40 of length 7 / TLS alert 40 of length 7)
Result of good request:                        TLS alert 40 of length 7
Result of bad request 1 (wrong first bytes):   TLS alert 40 of length 7
Result of bad request 2 (wrong 0x00 position): TLS alert 80 of length 7
Result of bad request 3 (missing 0x00):        TLS alert 40 of length 7
Result of bad request 4 (bad TLS version):     TLS alert 40 of length 7
```

# When

December 12th, 2017 ROBOT paper published

December 13th, 2017 We first realize there is a theoretical problem

December 13th-14th 2017, discussion with Shibboleth developer

December 19th, Communication with REN-ISAC TAG

January 6th, 2018, First instance of vulnerable IdP detected

January 10th, 2018, Comprehensive scans

January 12th, 2018, Communications to vulnerable universities

January 12th, 2018, Communications to FOG list

January 23rd, 2018, Shibboleth security advisory published

# Why

From InCommon IdP-only metadata:

- 94 “back channels” of interest
- Of those, 9 were vulnerable to ROBOT
- Of those, 3 had an exposed SAML signing key
- 8 of 9 were load balancers.  One was a Linux host
- 6 back channels were listening with a CA signed cert, not SAML cert
- Sites with vulnerable & exposed SAML signing keys have been resolved for ROBOT.

# Conclusion

Some questions:

- What should the role of federation operators be in securing the federation?
- What could we have done better?
- Did we make the right decision to embargo the “0Day” for a coordinated release with the Shibboleth Consortium? Should we have released sooner? Likely would have conflicted with the holiday break. Advisory may have competed with time off at many of the IdP operators.
- We had mixed results working with the three institutions that were vulnerable. Everything from sub-2 hour response to silence. However, even the silent institution is no longer vulnerable to ROBOT. So, presumably, notification worked.