CVE-2020-12676 - FusionAuth SAML v2.0 bindings in Java using JAXB - Signature Exclusion Attack

From: Advisories <advisories () compass-security com>
Date: Wed, 30 Sep 2020 12:02:07 +0000

```
#############################################################
#
# COMPASS SECURITY ADVISORY
# https://www.compass-security.com/research/advisories/
#
#############################################################
#
# Product:  SAML v2.0 bindings in Java using JAXB
# Vendor:   FusionAuth
# CSNC ID:  CSNC-2020-002
# CVE ID:   CVE-2020-12676
# Subject:  Signature Exclusion Attack
# Risk:     High
# Effect:   Remotely exploitable
# Author:   Felix Sieges <felix.sieges () compass-security com>
# Date:     2020-09-30
#
#############################################################
```

```
Introduction:
-------------
FusionAuth [1] provides authentication, authorization, and user management for any app.
SAML v2.0 bindings in Java using JAXB are a library used to integrate SAML
Authentication with Java Applications. Compass Security [2] identified a vulnerability
that allows remote attackers to forge messages and bypass authentication via a SAML
assertion that lacks a Signature element, aka a "Signature exclusion attack".


Affected:
---------
Vulnerable:
fusionauth-samlv2 0.2.3

Not vulnerable:
fusionauth-samlv2 0.2.4

Not tested:
No other version was tested, but it is believed for the older versions to be
vulnerable as well.


Technical Description
---------------------
Unauthenticated users can send forged messages to the FusionAuth to bypass
Authentication, impersonate other users or gain arbitrary roles. The SAML
Message can be send to the Application without a signature even if this is
required. The impact depends on individual applications that
implement fusionauth-samlv2.

The code which is responsible to verify the signature is called from the parseResponse
function [3]. The function checks whether a signature must be verified and if so
the function verifySignature is called to do signature verification checks.

https://github.com/FusionAuth/fusionauth-
```

```
  public AuthenticationResponse parseResponse(String encodedResponse, boolean verifySignature, PublicKey key)
      throws SAMLException {

    AuthenticationResponse response = new AuthenticationResponse();
    byte[] decodedResponse = Base64.getMimeDecoder().decode(encodedResponse);
    response.rawResponse = new String(decodedResponse, StandardCharsets.UTF_8);

    Document document = parseFromBytes(decodedResponse);
    if (verifySignature) {
      verifySignature(document, key);
    }
```

```
In the function verifySignature [4] the SAML message is parsed after validation that a
signature is attached to the message. If no signatures exist the function just returns.
Hence the parseReponse method assumes that the signature is valid and the SAML message
will be processed further.
```

```
  private void verifySignature(Document document, Key key) throws SAMLException {
    // Fix the IDs in the entire document per the suggestions at
http://stackoverflow.com/questions/17331187/xml-dig-sig-error-after-upgrade-to-java7u25
    fixIDs(document.getDocumentElement());

    NodeList nl = document.getElementsByTagNameNS(XMLSignature.XMLNS, "Signature");
    if (nl.getLength() == 0) {
      return;
    }

    for (int i = 0; i < nl.getLength(); i++) {
      DOMValidateContext validateContext = new DOMValidateContext(key, nl.item(i));
      XMLSignatureFactory factory = XMLSignatureFactory.getInstance("DOM");
      try {
        XMLSignature signature = factory.unmarshalXMLSignature(validateContext);
        boolean valid = signature.validate(validateContext);
        if (!valid) {
          throw new SAMLException("Invalid SAML v2.0 authentication response. The signature is invalid.");
        }
      } catch (MarshalException e) {
        throw new SAMLException("Unable to verify XML signature in the SAML v2.0 authentication response because we
couldn't unmarshall the XML Signature element", e);
      } catch (XMLSignatureException e) {
        throw new SAMLException("Unable to verify XML signature in the SAML v2.0 authentication response. The
signature
was unmarshalled we couldn't validate it for an unknown reason", e);
      }
    }
  }
}
```

```
Workaround / Fix:
-----------------
```

```
Upgrade to fusionauth-samlv2 0.2.4, where the signature checks are performed correctly

#TODO
Timeline:
---------
2020-04-29:     Discovery by Felix Sieges
2020-05-06:     Assigned CVE-2020-12676
2020-05-06:     Initial vendor notification
2020-05-06:     Initial vendor response
2020-12-05:     Release of fixed Version
2020-09-30:     Coordinated public disclosure date


References:
-----------
[1] https://fusionauth.io/
[2] https://compass-security.com
[3]
https://github.com/FusionAuth/fusionauth-
samlv2/blob/master/src/main/java/io/fusionauth/samlv2/service/DefaultSAMLv2Service.java#L502
[4]
https://github.com/FusionAuth/fusionauth-
samlv2/blob/dbcf3ccabbc89f43ca322c1526a6a3589e3a1f1f/src/main/java/io/fusionauth/samlv2/service/DefaultSAMLv2Service.j
ava#L820


_______________________________________________
Sent through the Full Disclosure mailing list
https://nmap.org/mailman/listinfo/fulldisclosure
Web Archives & RSS: http://seclists.org/fulldisclosure/
```

Current thread:

- **CVE-2020-12676 - FusionAuth SAML v2.0 bindings in Java using JAXB - Signature Exclusion Attack** *Advisories (Oct 02)*