Shibboleth Identity Provider Security Advisory [12 December 2022] OpenSAML-Java Security Advisory [12 December 2022]

Shibboleth konsortiet skickade ut nedanstående Security Advisory den 16 december 2022. Alla som använder Shibboleth Identity Provider eller OpenSAML-Java rekommenderas starkt att följa konsortiet rekommendationer snarast.

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Shibboleth Identity Provider Security Advisory [12 December 2022]
OpenSAML-Java Security Advisory [12 December 2022]

Older releases of the Shibboleth Identity Provider and OpenSAML-Java
library are potentially vulnerable to attacks ranging from denial of
service to remote code execution when given specially-crafted
encrypted XML to decrypt. Some decryption use cases include
unauthenticated message processing, so are widely accessible.

While the current releases of both software products (V4.2.0+) are
believed to be protected from the worst implications of this issue,
many people operate older releases of both the Identity Provider and
OpenSAML, so we are publishing this advisory in part as a courtesy.

It is believed that the worst outcomes on older versions also depend
on the use of non-current releases of the Java JDK, even as recent
as those prior to July of 2022.

At this time, it is believed that the risk of similar attacks in the
Shibboleth SP are not significant. We will update this, and publish a
second advisory in the event this determination changes.

OpenSAML and IdP mis-handle malicious encrypted XML content
===========================================================================
The XML Encryption specification, much as the original XML Signature
specification, contains an over-abundance of generality in various
areas, including the potential use of XSLT and XPath "transforms"
as processing instructions while calculating the encrypted content
to decrypt.

SAML provides very specific guidelines for how signed XML needs to
look, allowing pre-validation to prevent many problems. It provides
much less guidance on this matter for encrypted XML.

The OpenSAML Java library, up to and including V4.2.0, does not
perform sufficient validation on the content to outright prevent
execution of some malicious transforms. (OpenSAML V4.2.0 is present
in the V4.2.x releases of the Shibboleth Identity Provider.)
However, the Santuario XML Signature/Encryption library release (2.3.0)
used by these versions of our software contains a default-on secure
processing mode that precludes the worst of these attacks out of the
box, and so we know of no immediately practical attacks against these
versions of our software.

Unfortunately, older versions of our software rely on older Santuario
versions that do not default to this secure processing mode. They are
therefore potentially vulnerable, and these attacks are able to exploit
critical security vulnerabilities in versions of Java whose XSLT
implementation has not been patched.

There have been at least two remote code execution vulnerabilities
reported against Xalan-J, and in turn against Java, in the past 8 years,
one as recent as this past summer. As a result. versions of Java with
patch levels older than August 2022 are believed to be vulnerable to
at least one such issue.

Note that the actual Java LTS release (Java 8, Java 11, Java 17) is
not the relevant issue, but the underlying patch level of a given Java
version. All of the sources of OpenJDK provide routine patch updates
on a quarterly basis and these updates are critical. It is also crucial
to use these LTS releases rather than "feature releases" such as
Java 12-16, etc. due to the risk of missing out on such fixes.

We will include an enhancement in all future releases of OpenSAML
to harden the library against this class of attacks to the greatest
extent possible.
```

Recommendations
===============
Review the versions of Java and the Identity Provider and OpenSAML
software in your deployment. Make sure that Java is fully patched
and current for whichever LTS release you are using, and take steps
to ensure this remains true.

Note that the Shibboleth Project does not distribute Java in any of
our packaging, most particularly on Windows, where there is no
built-in source of Java and maintaining currency requires additional
effort. Some third-party packaging sources do include Java and you
should ensure you are staying up to date via those sources.

This Red Hat bug report [1] on one of the vulnerabilities mentions
the specific patch levels released this summer that address the
latest issue. Those Oracle patch levels refer to Java "in general"
and may not apply in specific instances where vendors have distributed
Java themselves (e.g., Debian and so on). When in doubt, always use
"the latest" Java for your LTS version and platform.

Obviously you should be taking steps to upgrade to a current IdP
version, but the advice above is critical if you cannot do so quickly.

While we do believe the older versions of our software are likely safe
from the worst of the vulnerabilities provided Java is current, we do
not believe that the risk of future attacks is insignificant, and so
this is not a recommended long term answer.

In the event that upgrading promptly is impossible (which would imply
you should be considering alternatives), a reasonable precaution is to
update your older deployment to xmlsec-2.3.0.jar by following these
steps:

1. Stop servlet container.
2. Remove the existing version of xmlsec-X.X.X.jar from
dist/webapp/WEB-INF/lib
3. Download, verify, and copy in the newer version from [2] to
that same location (or if you prefer, to edit-webapp/WEB-INF/lib).
4. Rebuild the war via the usual build.sh/build.bat command.
5. Start servlet container.

This is a compatible change for all V4.x IdP releases. We do not know
whether this is compatible with older versions.

Credits
=======
Khoadha of Viettel Cyber Security

[1] https://bugzilla.redhat.com/show_bug.cgi?id=2108554
[2] https://repo1.maven.org/maven2/org/apache/santuario/xmlsec/2.3.0/

URL for this Security Advisory:
https://shibboleth.net/community/advisories/secadv_20221216.txt

-----BEGIN PGP SIGNATURE-----

iQIzBAEBCgAdFiEE3KoVAHvtneaQzZUjN4uEVAIneWIFAmOc8EwACgkQN4uEVAIn
eWIIHg//b7+f2VHroUOKSksIvO8mkx5wrpS7U9pXh7eJV7eiDYYz3zHv27kwNgWt
yDAjzdYm73btjABgpR5pZOFwf93abJEEY883rFsx5YHwiHdNAe96sFBkupJCRj7i
cnNeW6g0AEb7pSBof4BNQBTWjyDf2c/upRV1QiXt5DTaBiJ1q01P+ZSKIhdc31NE
jtp6GdFLQyLw4m8UVHuqDQyngVt8q/Q2+OA3A/9e5kgO7G2z+hmgjiZVh23xwzt6
MEXOLGSdeCjOqstofRM8KROlrXzsyKsXSlgWa/v3Z8mnVS9iBEiWR7R6A2BZXMO3
7sc4jcMkcr1Cp2v1Fs18r3nbu9kQzbL5jxMjBNKlFwoP/FQLCeL4mGh1WK72o1DY
RanrFQce9m81EBA6YZ40QwnIPRRmXa8vnXb0nFovy9dwQImeNqxP/XUTM79MHRcR
zvQ+qztj0WPRi9sHtZ40rpp2U4luulcTLKpFgYLq00Yx3b1EQlQVCuM5auiE+lNp
pmzl4aHkNHJU3hflNzOWmyi44aUC3yrf6LXHutvqspz3TKbU8YJfL1cScSBHlVNT
8JKGFmdveRXdnnB5Px01HjPHgQHVobo6rP57WjQv70ntYUkBGL0u6cWvOJczqIwR
H6WG37YsTM3XjbVmO9WjwC5SyBEPLn04bpzxVX5Bls23QzlNjAw=
=8w+O
-----END PGP SIGNATURE-----