



Stark0de Sanctuary

# Pwning rConfig part I

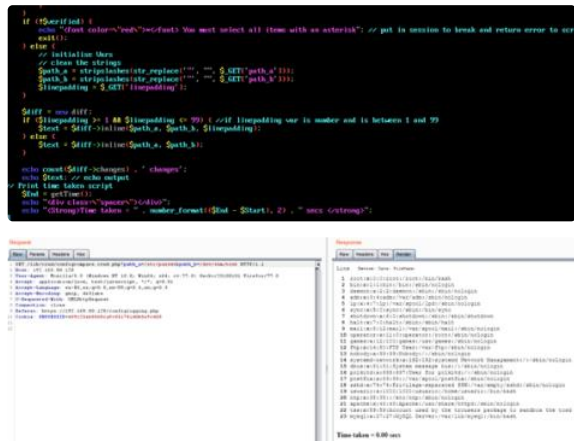
27 Aug 2020

## Pwning rConfig: part one

Hi everyone. I've been quite busy lately, but some weeks ago I decided to review the security of a software called rConfig version 3.9.5 ([www.rconfig.com](http://www.rconfig.com)) which is a Network Monitoring tool and I found several vulnerabilities. So let's take a look at them ;)

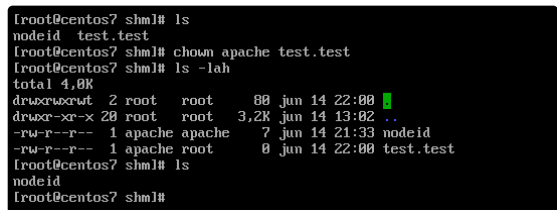
### LFI 1

Local file disclosure in `/lib/crud/configcompare.crud.php`, when comparing the parameter `path_a` to `path_b`, the file specified in `path_a` is disclosed if both files are completely different. Here we can see the contents of `/etc/passwd`



### Arbitrary file deletion

Arbitrary file deletion in `/lib/ajaxHandlers/ajaxDeleteAllLoggingFiles.php`. You can delete any file with an extension by specifying the file with the `path` parameter and the extension with the `ext` extension



### Server-Side Request Forgery

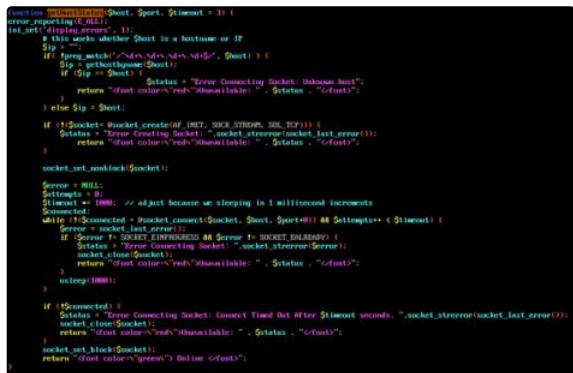
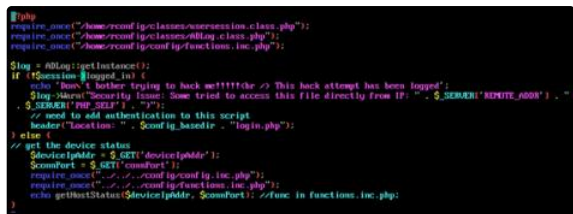
Server-Side Request Forgery in `/lib/ajaxHandlers/ajaxDeviceStatus.php`, you can open a connection to the inside of the machine with the `deviceIpAddr` parameter and the `connPort` parameter.



### OPEN PORT:

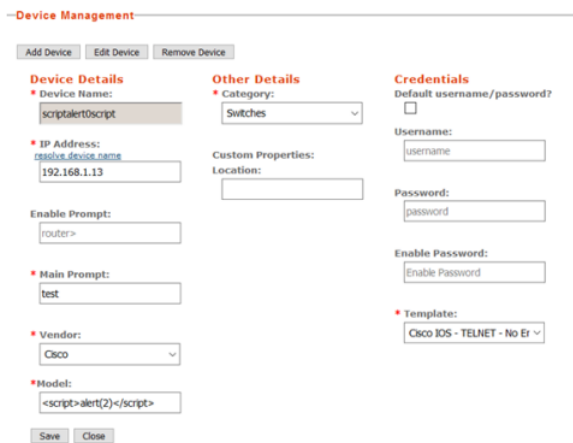


## Error:



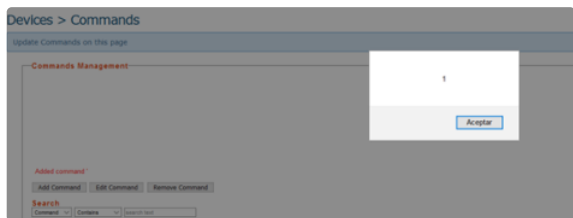
## XSS

Cross-Site Scripting in /devices.php > Add device, introduce a payload such as <svg onload=alert(1)> in the Model field and clicking Save (it's needed to fill the rest of fields with info). Then visiting /devicemgmt.php?deviceId=1&device=deviceName



## XSS 2

Cross-Site Scripting in /commands.php > Add command, introduce a payload such as <svg onload=alert(1)> in the Command field and click Save.



## XSS 3

Cross-Site Scripting in /snippets.php > Add snippet, introduce a payload such as <svg onload=alert(1)> in the Snippet field and click Save(fill the rest of fields first).

The screenshot shows the 'Configuration Snippets' page in the AWS IAM console. The page has a header 'Configuration Tools > Config Snippets' and a sub-header 'Configuration Snippets'. Below this, there is a section 'Edit Snippet "TTF in Database"'. This section contains three buttons: 'Add Snippet', 'Edit Snippet', and 'Remove Snippet'. Below these buttons are three form fields: 'Snippet Name', 'Snippet Description', and a large text area for the snippet content. The 'Snippet Name' field is highlighted with a red box, the 'Snippet Description' field is highlighted with a blue box, and the 'Add Snippet' button is highlighted with a red box. Below the form fields are 'Save' and 'Close' buttons. At the bottom of the page, there is a 'Search' section with a search bar and a 'Search' button. The 'Snippet Name' field is highlighted with a red box, and the 'Snippet Description' field is highlighted with a blue box. A red box also highlights the 'Add Snippet' button. The 'Save' button is highlighted with a green box. The 'Search' section at the bottom is also visible.

## Privilege escalation 1

Using sudo zip:

[illegible]

## Privilege escalation 2

By doing `sudo crontab -e` and then `#!/bin/bash`

**Arbitrary file read:**

Following this we can read any privileged file: <https://gtfobins.github.io/gtfobins/tail/>

So those are some of the vulns I found during the research, hope you enjoyed the article. Back with the second part soon :)

### Comments

[Load Comments](#)

## Related Posts

Pwning rConfig part II 13 Oct 2020

## How I found a Remote Code Execution in OpenEDX 17 May 2020

Certified Red Team Professional Review 30 Apr 2020