VANJA KORAĆ
Mathematical Institute SASA
Belgrade, Serbia
E-mail: vanja@mi.sanu.ac.rs

DRAGAN PRLJA
Institute for Comparative Law
Belgrade, Serbia

004.383.2.056
COBISS.SR-ID 272034316
Original research article

Received: October 14th 2018
Accepted: October 30th 2018

WEB SERVER SECURITY ASPECT

ABSTRACT

This work covers a security aspect when it comes to designing a secure web server that uses certain public services and public open source software. The uninstalling of unnecessary applications and services has been performed, since unnecessary applications or services can be vulnerable, thus resulting in a potential intrusion vector. A metric that is crucial for the system is defined. Since the database is an essential part of the web server, the required processor power, amount of memory, network speed and disk capacity are defined, so that the server can provide service in all conditions of operation without being disturbed. The metric is important in the phase when the parameters are defined according to which the system will operate, such that the parameters can be checked. Also, a vulnerability scan of the operating system must be performed after the implementation of the security mechanisms.

KEY WORDS: INTRUSION DETECTION ENVIRONMENT, VULNERABILITY SCAN, SSH BRUTE-FORCE PREVENTION.

WEBSERVEROPERATINGSYSTEM

The operating system on which the web server was launched is **Ubuntu 16.04.2 LTS**.[1] The name of the server is demo.mi.sanu.ac.rs, and the IP address of the server is ###.###.###.###.[2]

The RSA2 key fingerprint (ssh-rsa) of the server is:
demo.mi.sanu.ac.rs ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDL2lelofsoQDSpQwELhC6qpjI73lNXKoa0FVIh61w7L4rn9hfbSWg2P3wYPHJUHceMSFQYW4sa9+MPEY1mz4Bug/NvA82gwaRw6L6M/a2/ntdaYZHMpzQ5nWpv71wgKUBoIIftLBPPOkzRv1PO7koy/LXnkE6tcmdVmE8MUnfnPGALPew8+s7XZE/4T6lKvzdfBBCPecrNE2Eatqr9uU+7qjoM0OhpF12SeeMNezuLT2P/r4/RPigomzkiHjMt9PpykAjGsxDuvfuCvhuYHlxwmTKZCVOKMprYgdqVM00yv6b0FbV/PJFAxVavMafYbud2cGuR5nIZO7JcA7N+eXdh

The ECDSA key fingerprint is:
demo.mi.sanu.ac.rs ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBApODNJa1cxRoXN0BMPUJOPQvf9o/6E4at9DG59kYxJhmtjlrSa/FGbztp7pD5j8Mou8RaaF5A+BiAucZBajC58=

1 The article results from the project *IRS - Viminacium, Roman city and military legion camp – research of the material and nonmaterial culture of inhabitants by using the modern technologies of remote detection, geophysics, GIS, digitalization and 3D visualization* (*no 47018*), funded by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

2 For security reasons, the IP address is not displayed, and the name of the server on which the protection measures have been implemented has been changed.

SERVICES ON THE OPERATING SYSTEM

Only the necessary services of ssh, mysql, apache, and postfix were booted on the system. The following ports are used on the server: 22(sshd), 25(smtp) and 80 (httpd).

MODULE FOR TESTING FILE AND DIRECTORY INTEGRITY

The AIDE (Advanced Intrusion Detection Environment) module for testing files and directories is applied on the system[3]. AIDE creates a database with regular expression rules read from the configuration file. After initialisation of the database, the verification (integrity check) of the files can be performed. There are several different Message Digest algorithms used for the file integrity check. The supported algorithms are as follows: md5, sha1, rmd160, tiger, crc32, sha256, sha512, and whirlpool (and with libmhash: gost, haval, and crc32b). Support for the file attribute check includes: File type, Permissions, Inode, Uid, Gid, Link name, Size, Block count, Number of links, Mtime, Ctime and Atime. The following algorithms "sha256+sha512+rmd160+haval+gost+crc32+tiger" were applied on the demo.mi.sanu.ac.rs server. All of the usual file attributes can be checked for inconsistencies. (Korać, Todorović and Mihaljević 2017).

AIDE has the following configuration:

```
# AIDE conf
# The daily cron job depends on these paths
database=file:/var/lib/aide/aide.db
database_out=file:/var/lib/aide/aide.db.new
database_new=file:/var/lib/aide/aide.db.new
gzip_dbout=yes

summarize_changes=yes

grouped=yes
verbose = 6

# Set to yes to print the checksums in the report in hex format
report_base16 = no

# if you want to sacrifice security for speed, remove some of these
# checksums. Whirlpool is broken on sparc and sparc64 (see #429180,
# #420547, #152203).
Checksums = sha256+sha512+rmd160+haval+gost+crc32+tiger

# The checksums of the databases to be printed in the report
# Set to 'E' to disable.
database_attrs = Checksums

# check permissions, owner, group and file type
OwnerMode = p+u+g+ftype

# Check size and block count
Size = s+b

# Files that stay static
InodeData = OwnerMode+n+i+Size+l+X
StaticFile = m+c+Checksums

# Files that stay static but are copied to a ram disk on startup
# (causing different inode)
RamdiskData = InodeData-i

# Check everything
Full = InodeData+StaticFile

# Files that change their mtimes or ctimes but not their contents
VarTime = InodeData+Checksums
# Files that are recreated regularly but do not change their contents
VarInode = VarTime-i

# Files that change their contents during system operation
```

3 http://aide.sourceforge.net/.

VarFile = OwnerMode+n+l+X

# Directories that change their contents during system operation
VarDir = OwnerMode+n+i+X

# Directories that are recreated regularly and change their contents
VarDirInode = OwnerMode+n+X

# Directories that change their mtimes or ctimes but not their contents
VarDirTime = InodeData
# Logs grow in size. Log rotation of these logs will be reported, so
# this should only be used for logs that are not rotated daily.
Log = OwnerMode+n+S+X

# Logs that are frequently rotated
FreqRotLog = Log-S

# The first instance of a rotated log: After the log has stopped being
# written to, but before rotation
LowLog = Log-S

# Rotated logs change their file name but retain all their other properties
SerMemberLog =Full+I

# The first instance of a compressed, rotated log: After a LowLog was
# compressed.
LoSerMemberLog = SerMemberLog+ANF

# The last instance of a compressed, rotated log: After this name, a log
# will be removed
HiSerMemberLog = SerMemberLog+ARF

# Not-yet-compressed log created by logrotate'sdateext option:
# These files appear one rotation (renamed from the live log) and are gone
# the next rotation (being compressed)
LowDELog = SerMemberLog+ANF+ARF

# Compressed log created by logrotate'sdateext option: These files appear
# once and are not touched any more.
SerMemberDELog = Full+ANF

# For daemons that log to a variable file name and have the live log
# hardlinked to a static file name
LinkedLog = Log-n

/journals Full
/usr/share/exist-db/webapp/WEB-INF/data Full
/home/bibladmin/exist_backup Full

E-mail sending after the integrity check was adjusted to operate every 24h at 07h.
/etc/default/aide
# Set this to no to disable daily aide runs
CRON_DAILY_RUN=yes
MAILTO=####@mi.sanu.ac.rs (#### has been placed in this report due to security reasons)

Postfix log after sending the AIDE report:

July 25 21:55:35 demo postfix/qmgr[2741]: 2D1EDCC0C7B: from=<root@demo.mi.sanu.ac.rs>, size=92790, nrcpt=1 (queue active)
July 25 21:55:35 demo postfix/smtp[8922]: 2D1EDCC0C7B: to=<###@mi.sanu.ac.rs>, relay=mi.sanu.ac.rs[147.91.96.2]:25, delay=0.13, delays=0.05/0.01/0.01/0.05, dsn=2.0.0, status=sent (250 2.0.0 v6PJtZUu026615 Message accepted for delivery)
July25 21:55:35 demo postfix/qmgr[2741]: 2D1EDCC0C7B: removed

MODULE FOR THE PREVENTION OF A BRUTE-FORCE ATTACK ON THE SSH SERVER

The denyhosts module, which prevents brute-force attacks on the SSH service of the server, has been implemented. A brute force attack is a method used by malicious attackers to obtain access to servers, by using hundreds and thousands of random combinations of user names and passwords. This module is designed to prevent a brute-force attack on the SSH server, by tracking inadequate attempts of logging into the system from the authentication log file of the server itself, blocking malicious IP addresses through/etc/hosts.deny.

DENYHOST CONFIGURATION

On Ubuntu Linux systems, the mode in which this module is started is daemon mode and the associated configuration file is /etc/denyhosts.conf.

```
# Debian and Ubuntu
SECURE_LOG = /var/log/auth.log
# Most operating systems:
HOSTS_DENY = /etc/hosts.deny
#
# PURGE_DENY: removed HOSTS_DENY entries that are older than this time
#               when DenyHosts is invoked with the --purge flag
#
#     format is: i[dhwmy]
#     Where ‘i’ is an integer (eg. 7)
#          ‘m’ = minutes
#          ‘h’ = hours
#          ‘d’ = days
#          ‘w’ = weeks
#          ‘y’ = years
#
# never purge:
PURGE_DENY =
# To block only sshd:
BLOCK_SERVICE  =sshd
# DENY_THRESHOLD_INVALID: block each host after the number of failed login
# attempts has exceeded this value.  This value applies to invalid
# user login attempts (eg. non-existent user accounts)
#
DENY_THRESHOLD_INVALID = 3
DENY_THRESHOLD_VALID = 5
DENY_THRESHOLD_ROOT = 2
DENY_THRESHOLD_RESTRICTED = 1
WORK_DIR = /var/lib/denyhosts
ETC_DIR = /etc
SUSPICIOUS_LOGIN_REPORT_ALLOWED_HOSTS=YES
HOSTNAME_LOOKUP=NO
LOCK_FILE = /run/denyhosts.pid
ADMIN_EMAIL = ###@mi.sanu.ac.rs
SMTP_HOST = XXX.XXX.XXX.XXX ()
SMTP_PORT = 25
SMTP_FROM = DenyHosts<nobody@localhost>
SMTP_SUBJECT = DenyHosts Report
ALLOWED_HOSTS_HOSTNAME_LOOKUP=NO
AGE_RESET_VALID=5d
AGE_RESET_ROOT=25d
AGE_RESET_RESTRICTED=25d
AGE_RESET_INVALID=10d
DAEMON_LOG = /var/log/denyhosts
DAEMON_LOG_MESSAGE_FORMAT = %(asctime)s - %(name)-12s: %(levelname)-8s %(message)s
DAEMON_SLEEP = 30s
DAEMON_PURGE = 1h
SYNC_DOWNLOAD = no
```

ENABLED FIREWALL

The firewall that is located in the kernel itself is a mechanism that manages network traffic (network packets). Its frontend is called iptables. It controls incoming and outgoing traffic, and routing and network address translation (NAT) can be performed. Iptables does not analyse the content of network packages (tcp/ip/udp), but it can function as a stateful firewall, on the basis of which connections can be paired. For example, the ftp protocol operates on two channels over ports 20 and 21, where one channel serves for data flow and the other for connection control. Iptables is aware of such connections and if there are such so-called stateful linked connections, it dynamically allows the required connection.

Logging of root users via ssh is not allowed with password only. Logging with root account via ssh is possible only with the help of a private ssh key.

The applied firewall on the demo.mi.sanu.ac.rs server has the following configuration:

root@demo:~# ufw status

Status: active

To	Action	From
22	ALLOW	Anywhere
80	ALLOW	Anywhere
443	ALLOW	Anywhere
8080	ALLOW	Anywhere
22 (v6)	ALLOW	Anywhere (v6)
80 (v6)	ALLOW	Anywhere (v6)
443 (v6)	ALLOW	Anywhere (v6)
8080 (v6)	ALLOW	Anywhere (v6)

An NMAP external system scan shows which ports are active:

#nmap 147.91.96.100
Starting Nmap 7.50 (https://nmap.org) at 2017-07-25 17:55 CEST
Nmap scan report for demo.mi.sanu.ac.rs (147.91.96.16)
Host is up (0.026s latency).
Not shown: 997 filtered ports
PORT STATE SERVICE
22/tcp openssh
80/tcp open http
443/tcp closed https

Nmap done: 1 IP address (1 host up) scanned in 8.14 seconds

Important logs and examples:

ufw.log blocked port 25

Successful sending of an e-mail to the mail server:

July 25 17:18:20 demo postfix/qmgr[2741]: 59BB1CC0E49: from=<root@demo.mi.sanu.ac.rs>, size=396, nrcpt=1 (queue active)
July 25 17:18:20 demo postfix/smtp[2773]: 59BB1CC0E49: to=<vanja@mi.sanu.ac.rs>, relay=mi.sanu.ac.rs[xxx.xxx.96.2]:25, delay=0.05, delays=0.02/0/0.01/0.01, dsn=2.0.0, status=sent (250 2.0.0 v6PFIKXr021367 Message accepted for delivery)

Unsuccessful sending of an e-mail to gmail from root:

July 25 17:25:41 demo postfix/qmgr[2741]: 0EAB3CC0E3F: from=<root@demo.mi.sanu.ac.rs>, size=317, nrcpt=1 (queue active)
July 25 17:25:41 demo postfix/smtp[3988]: connect to gmail-smtp-in.l.google.com[xxx.102.1.27]:25: No route to host
July 25 17:25:45 demo postfix/smtp[3988]: 0EAB3CC0E3F: to=<XYZ@gmail.com>, relay=none, delay=488, delays=483/0.02/4.3/0, dsn=4.4.1, status=deferred (connect to alt4.gmail-smtp-in.l.google.com[74.125.30.27]:25: No route to host)

Log of blocked access to ports 23 and 25:

July 25 18:07:20 demo kernel: [14806.975866] [UFW BLOCK] IN=ens160 OUT= MAC= 00:50:56:a2:4c:e4:00:19:e8:3d:11:42:08:00 SRC=81.248.41.124 DST=###.###.###.### LEN=44 TOS=0x00 PREC=0x00 TTL=47 ID=63551 PROTO=TCP SPT=57103 DPT=23

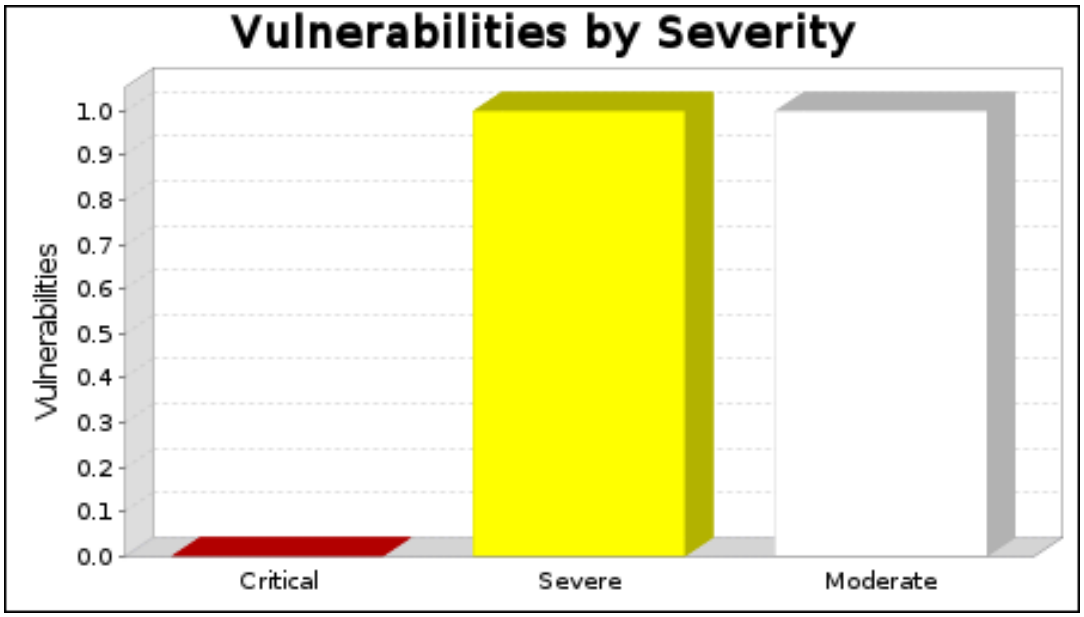

Fig. 2

WINDOW=59711 RES=0x00 SYN URGP=0
July 25 18:08:50 demo kernel: [14897.518444] [UFW BLOCK] IN=ens160 OUT= MAC= 00:50:56:a2:4c:e4:00:19:e8:3d:11:42:08:00 SRC=89.248.160.252 DST=###.###.###.### LEN=40 TOS=0x00 PREC=0x00 TTL=243 ID=54321 PROTO=TCP SPT=58269 DPT=25 WINDOW=65535 RES=0x00 SYN URGP=0

aide.log file was accessed but has not been changed:
Directory: /usr/share/exist-db/webapp/WEB-INF/data/fs/db/elb/2016/ActaStomatNis
Mtime : 2017-05-17 10:34:15 +0200 | 2017-07-04 10:25:56 +0200
Ctime : 2017-05-17 10:34:15 +0200 | 2017-07-04 10:25:56 +0200
Linkcount: 3
auth.log example of sudo command use
July 25 18:24:32 demosystemd: pam_unix(systemd-user:session): session opened for user bibladmin by (uid=0)
July25 18:24:32 demosystemd-logind[1218]: New session 21 of user bibladmin.
July 25 18:24:40 demosudo: bibladmin : TTY=pts/6 ; PWD=/home/bibladmin ; USER=-root ; COMMAND=/bin/ls /root/
July 25 18:24:40 demosudo: pam_unix(sudo:session): session opened for user root by bibladmin-(uid=0)
July 25 18:24:40 demosudo: pam_unix(sudo:session): session closed for user root

Operating system vulnerability scan after implementation of safety mechanisms

By scanning and showing the vulnerability of implemented systems, vulnerabilities which can potentially be used by safety threats (malicious programs or malicious attackers that can endanger computer systems and information) are preventatively detected [Korac 2014]. With the proactive elimination of these vulnerabilities, preventive protection is accomplished. Protection of the system precisely involves prevention with detection. Prevention includes risk assessment, access control, encryption and firewalls (Korać, Todorović and Prlja 2017).

Vulnerability scanning is performed with the Nexpose security audit tool from Rapid7 LLC.

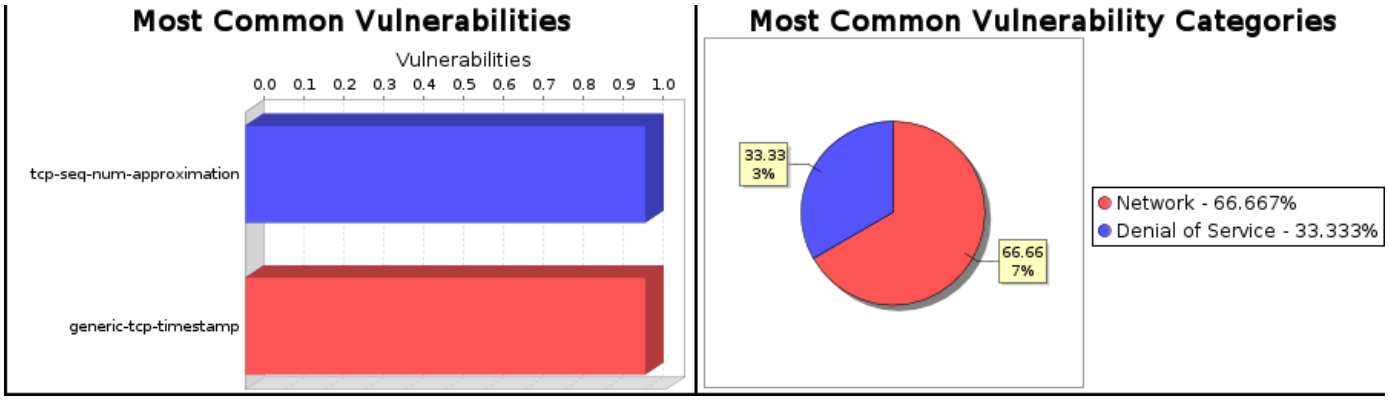

Fig. 3 Presentation of detected vulnerabilities

Site Name	Start Time	End Time	Total Time	Status
###.mi.sanu.ac.rs	July 13, 2017 19:14, CEST	July 13, 2017 19:15, CEST	1 minute	Success

The audit was performed on an active system and complete scanning was executed.

In Figure 2, two non-critical vulnerabilities found in the system scanning process are noticeable. Critical vulnerabilities have not been detected and such vulnerabilities require special attention and must be dealt with promptly. Critical vulnerabilities are used relatively easily by malicious attackers who, with the help of the exploit, can gain complete control over the affected system. There is one vulnerability on the system that is designated as severe, which is difficult to be exploited by the attacker and which does not provide access to the attacker to the server. The second vulnerability is designated as a moderate vulnerability. Moderate vulnerability types are those vulnerabilities that allow a malicious attacker to obtain information useful for planning a specific attack on the network. They need to be analysed and resolved, but they are not as urgent as critical vulnerabilities.

The detected severe vulnerability is called **tcp-seq-num-approximation**, and the moderate vulnerability is called **generic-tcp-timestamp**.

Description of the tcp-seq-num-approximation vulnerability:

When TCP uses a large window size, it enables the remote malicious attacker to hit the sequence number and thus cause denial of service (connection loss) of the established TCP connections based on the continuous injection of TCP RST packets, especially in long-lived protocol connections such as BGP.

To solve the vulnerability:

Enable TCP MD5 Signatures – the options for allowing TCP MD5 signatures are described in the RFC 2385[4] document. In this way, the risk of certain security attacks of BGP, such as TCP reset, is reduced.

Descripion of the generic-tcp-timestamp vulnerability:

The tested host corresponds with the TCP timestamp. Based on the TCP timestamp response, a malicious attacker can detect certain information such as the server's uptime, thus providing additional information to the attacker when planning

4 http://www.ietf.org/rfc/rfc2385.txt

future attacks. In addition, in certain operating systems, TCP timestamp responses differ, so the malicious attacker can also obtain the fingerprint of the OS, i.e., the OS type and OS version.

To solve the vulnerability:

Disable TCP timestamp responses on the system. Set the net.ipv4.tcp_timestamps value to 0 with the following command:

#sysctl -w net.ipv4.tcp_timestamps=0

Additionally, set the displayed value in the default sysctl configuration file (sysctl.conf) to:

net.ipv4.tcp_timestamps=0

CONCLUSION

After booting up the operating system on the server, it is necessary to set up the *demo.mi.sanu.ac.rs* web server, implement the module used for preventing brute-force attacks on the SSH server, implement the file integrity monitoring module, and start a vulnerability analysis of the booted server. The end users and system users who run different services (e.g., ssh, mysql, apache, postfix) are differentiated on the system. Each application should have its own username and its group under which it will be executed such that the processes, i.e., the relationship between the applications and the server itself, can be managed. With audit tools, you can get a picture of the condition and history of what happened on the machine. A vulnerability analysis on the demo.mi.sanu.ac.rs server was performed with the Rapid 7 Nexpose tool. The analysis found no critical vulnerabilities, which is normal, since a modern server has been installed and patched. In addition, tcp timestamp is allowed, which is standard and is not only desirable for high security systems; the demo server is not considered to be in that category because it is a public server that uses public services and public open source software such as Apache. Solutions have been proposed for all detected vulnerabilities, if explicitly required by the organisation's security policy.

* * *

BIBLIOGRAPHY

Korać, V. 2014
Digitalna forenzika u funkciji zaštite informacionog sistema baziranog na Linux i Windows platformama, nepublikovana doktorska disertacija, Univerzitet u Beogradu, 2014.

Korać, V., Todorović M. and Mihaljević M.2017 Metod I realizacija inicijalne zaštite bibliometrijskog sistema Ministarstva prosvete, nauke I tehnološkog razvoja,Tehničko rešenje, Beograd 2017.

Korać, V., Todorović M. and Prlja D. 2017 Windows default services vulnerabilities assessment, Archaeology and Science 12, Centar za nove tehnologije Viminacium Arheološki Institut Beograd, ISSN 1452-7448, UDK 004.451.9.056.57, COBISS.SR-ID 254104844, p. 195-210, Beograd, 2017.

REZIME
SIGURNOSNI ASPEKT WEB SERVERA

KLJUČNE REČI: PROVERA INTEGRITETA FAJLOVA/DIREKTORIJUMA, SKENIRANJE RANJIVOSTI, SSH BRUTE-FORCE PREVENCIJA.

Ovim radom je obuhvaćen sigurnosni aspekt kada je u pitanju dizajn bezbednog web servera koji koristi određene javne servise i javni softver otvorenog koda. Izvršeno je deinstaliranje nepotrebnih aplikacija i servisa, jer nepotrebne aplikacije ili servisi mogu biti ranjivi čime se ostvaruje potencijalni vektor upada. Definisana je metrika koja je ključna za sistem. S obzirom da je suštinski deo web servera baza podataka, definisana je potrebna procesorska snaga, količina memorije, mrežna brzina i kapacitet diska, da bi taj server u svim uslovima rada mogao nesmetano da obezbedi servis. Metrika je važna u fazi kada se definišu parametri prema kojima će sistem da radi da bi se imali parametri pomoću kojih se može proveriti ispravnost rada računarskog sistema. Nakon podizanja operativnog sistema na serveru, neophodnih servisa za postavljanje web servera, implementiranja modula koji služi za sprečavanje brute-force napada na SSH server i modula za proveru integriteta fajlova, izvršena je analiza ranjivosti podignutog servera. Analizom je utvrđeno da ne postoje kritične ranjivosti, što je i normalno s obzirom je instaliran moderan server i zakrpljen (pečovan), pored toga dozvoljen je tcp timestamp koji je standardan i nije poželjan samo kod visoko bezbednih sistema, gde server demo ne spada sobzirom da je u pitanju javni server koji koristi javne servise i javan software otvorenog koda poput apachea. Za sve uočene ranjivosti predložena su i razrešavanja istih ukoliko bezbednosna politika u organizaciji izričito zahteva.