Biznado (Natalia Wocial Przemysław Adam Mączka) & Klocki (Aga Patro, Jakub Młocek)

ANALYSIS OF CURRENT SYSTEMS, PROCESSES AND ARCHITECTURES AND PROPOSED CHANGES

From our in-depth analysis, changes need to be made at Eklamot in the following areas:

PSM

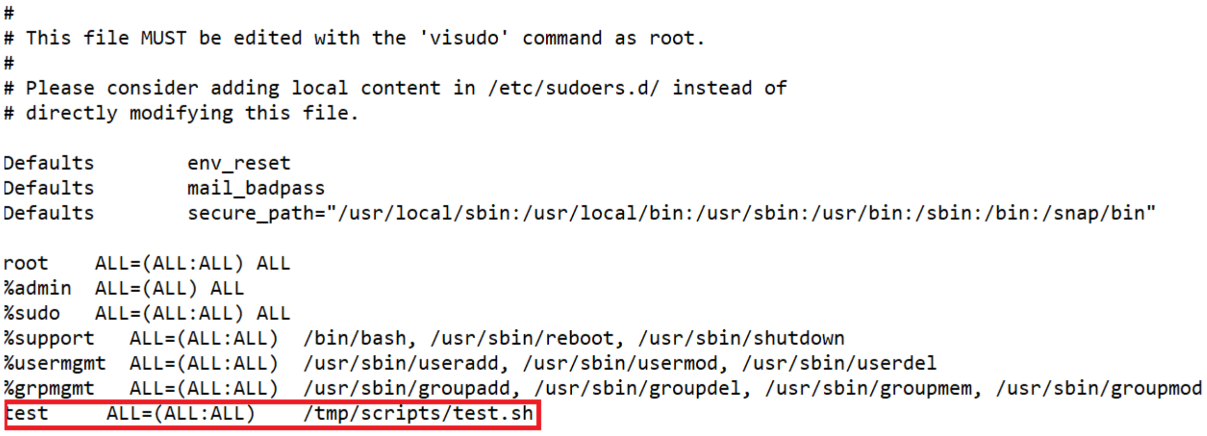
It's a good practice to have Privileged Session Manager. PSM controls all IT manager sessions and creates video recordings of the devices in the network as well as virtual servers, and has the capacity to control any privileged session from start to the end. On the other hand, PSM can also monitor various network components such as employees, third parties and integrated systems via session control authorization. Utilized to monitor, manage and control encrypted manager sessions, PSM acts as a gateway between the session manager, users and the target endpoints. The man-in-the-middle approach, which is the fundamental reason behind Privileged Session Manager's operating principle to be such functional, eliminates the need to establish any middleware software in the target endpoints. Thanks to the man-in-the-middle approach, access portals or client applications are not required to establish connection.

SUDOERS

The sudoers file is a file used by Linux and Unix system administrators to grant system permissions to system users. This allows the administrator to control the activities of other users. For security reasons, before executing a command that requires administrator privileges, Linux checks the sudoers file to see if the user name is included in the administrator's list of permissions, preventing unauthorized persons from executing the action.

It is possible to bypass this security by logging in as "root" with the command "su -l." Adding "- l". It allows you to execute a command that requires administrator privileges without granting those privileges.

The following screenshot documents that after logging in as "root" to Linux, the user has the ability to execute arbitrary commands without the system administrator having granted Eklamot system permissions.



```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.

Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

root    ALL=(ALL:ALL) ALL
%admin  ALL=(ALL) ALL
%sudo   ALL=(ALL:ALL) ALL
%support   ALL=(ALL:ALL)  /bin/bash, /usr/sbin/reboot, /usr/sbin/shutdown
%usermgmt  ALL=(ALL:ALL)  /usr/sbin/useradd, /usr/sbin/usermod, /usr/sbin/userdel
%grpmgmt   ALL=(ALL:ALL)  /usr/sbin/groupadd, /usr/sbin/groupdel, /usr/sbin/groupmem, /usr/sbin/groupmod
test     ALL=(ALL:ALL)    /tmp/scripts/test.sh
```

“test ALL=(ALL:ALL) /tmp/scripts/test.sh” This line is potentially dangerous. User test can execute with sudo privileges **“test.sh”** script which source code can be modified. This means he can execute each command as a root which poses a potential threat to the company. Group **“%support”** shouldn't have access to run/bin/bash as root. It also allows to run any available commands as root which isn’t necessary for support.

NMAP

NMAP is an open-source network mapping tool for network exploration and security audits. It is designed to scan large networks, but it also works well for individual addresses. NMAP uses low-level IP packets to identify which addresses are available on the network, which services are available, which operating systems, which firewall types are used, and dozens of other features. NMAP is commonly used for security audits, many network and system administrators use it to perform routine activities such as inventorying network resources, managing software updates, and monitoring systems and uptime.

The result of NMAP is a list of scanned addresses with additional information depending on the selected options. One of the most important pieces of information is the “List of Interesting Ports.” It contains port numbers, logs, service names, and detected status. The condition can be described as open, filtered, closed, or unfiltered.

- Open means that the application on the investigated address is waiting for incoming connections/packets on this port.
- Filtered means that a firewall system or other device blocking network traffic does not allow communication to that port and therefore NMAP cannot detect whether the port being tested is open or closed.
- The closed port does not have an application that supports network communication.
- Ports classified as unfiltered responded to NMAP queries, but it was not possible to determine whether they were open or closed.

Any open port is potential access for hackers to attack a company. Therefore, it is very important to make sure that ports that do not wait for connections/packets are closed. The following screenshot shows that Eklamot’s IT team neglected this case, and unknowingly allow criminals to launch an attack. It is very important to close ports that are currently not used.

Ports to be opened:

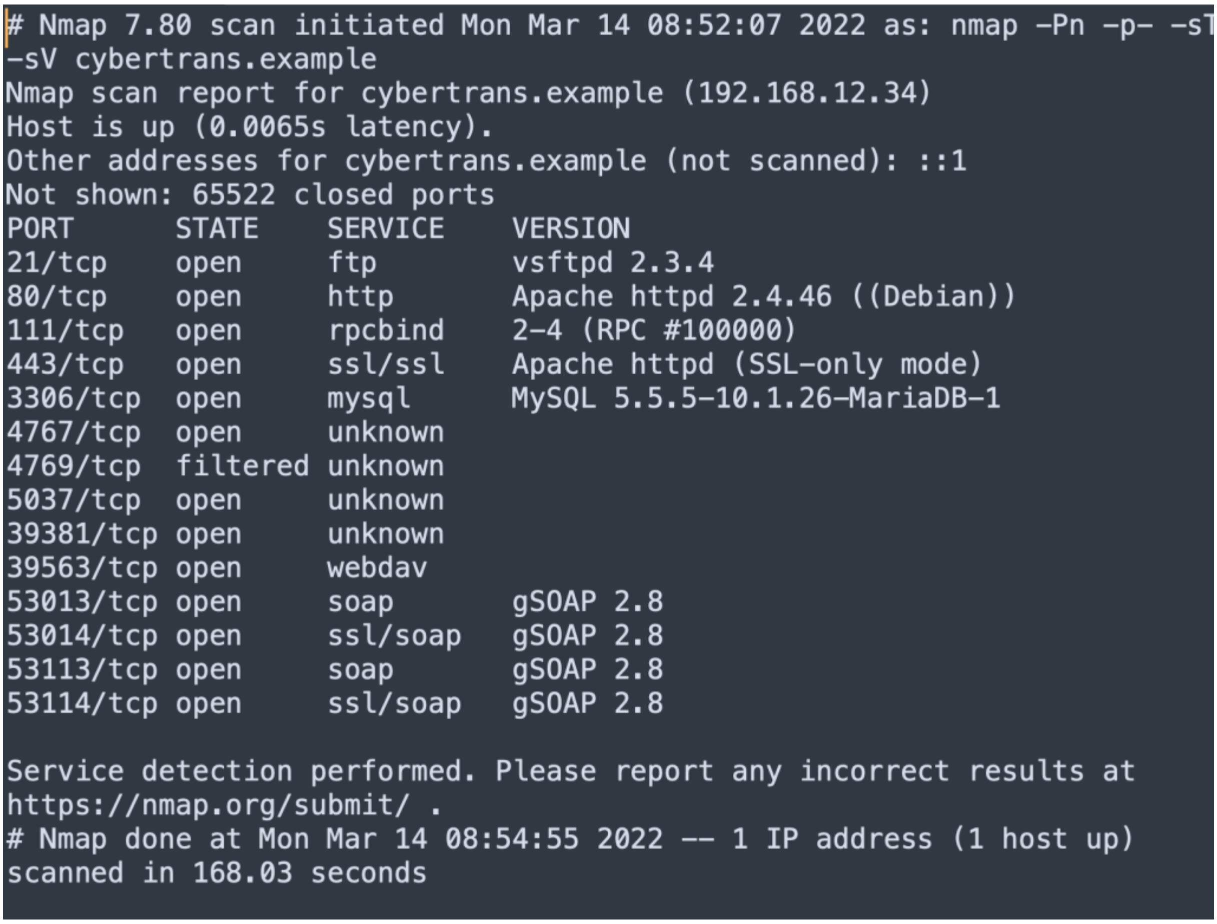
Having too many open ports expose servers to many potential attack vectors. Opened should remain only:

1. “21 (ftp)” for file transfer protocol.
2. “80 (http)” for website to be achieved.
3. “443 (https)” for secure website traffic.

Why close some ports?:

- We decided to close port 111. RPC service has a history of security vulnerabilities. Having this port exposed allows everybody to query information without a need to authentication. It should be opened only for certain whitelist of IPs.
- Port 3306 should also be closed. Exposing port 3306 can make our server vulnerable to attack. If a connection to database is necessary it is preffered to use ssh tunel instead.

- Ports: 4767, 4769, 5037, 39381 haven't got any known reason to be oppened. We need documentation of applications for further decisions.
- Port 39563 should be closed, webdav works on port 80 and 443 by default.
- Ports: 53013, 53014, 53113, 53114 should be closed tempolary, becouse of CVE-2021-21783 vulnerability avaliable for gSOAP 2.8.
- Moreover there is an Exploit avaliable for ftp vsftpd 2.3.4 which is used in our application. It should be updated rapidly to the newest version.



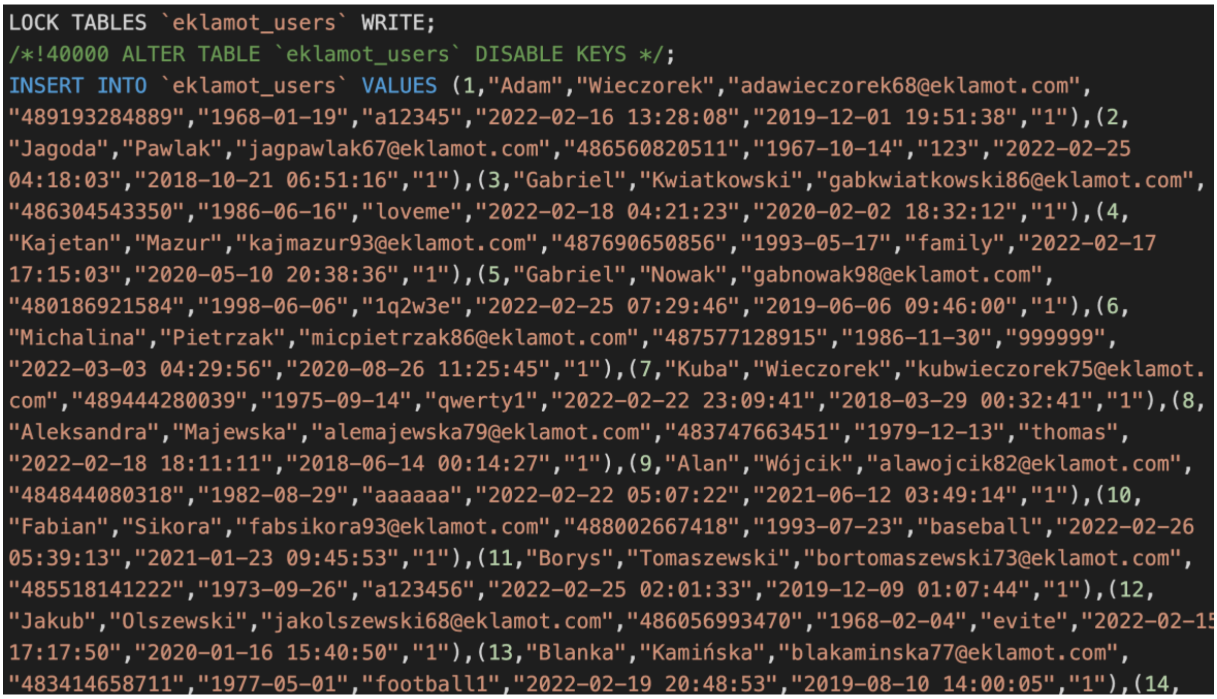
```
# Nmap 7.80 scan initiated Mon Mar 14 08:52:07 2022 as: nmap -Pn -p- -sT
-sV cybertrans.example
Nmap scan report for cybertrans.example (192.168.12.34)
Host is up (0.0065s latency).
Other addresses for cybertrans.example (not scanned): ::1
Not shown: 65522 closed ports
PORT      STATE    SERVICE    VERSION
21/tcp    open     ftp        vsftpd 2.3.4
80/tcp    open     http       Apache httpd 2.4.46 ((Debian))
111/tcp   open     rpcbind    2-4 (RPC #100000)
443/tcp   open     ssl/ssl    Apache httpd (SSL-only mode)
3306/tcp  open     mysql      MySQL 5.5.5-10.1.26-MariaDB-1
4767/tcp  open     unknown
4769/tcp  filtered unknown
5037/tcp  open     unknown
39381/tcp open     unknown
39563/tcp open     webdav
53013/tcp open     soap       gSOAP 2.8
53014/tcp open     ssl/soap   gSOAP 2.8
53113/tcp open     soap       gSOAP 2.8
53114/tcp open     ssl/soap   gSOAP 2.8

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Mon Mar 14 08:54:55 2022 -- 1 IP address (1 host up)
scanned in 168.03 seconds
```

DataBase

The Eklamot company uses two file storage options, it is both a cloud and an on-premise solution. Storing files in the cloud are characterized by the fact that security is automated and all responsibility rests on the contractor from whom we use the cloud. Any changes or requirements that we would like to introduce as a company would entail additional costs. However, on-premise file storage is fully implemented by your company. This allows you to make all kinds of changes in any range with little cost. This solution also allows you to store data offline, which translates into less possibility of their leakage. In our opinion, Eklamot should invest in the development of off-premise data storage. Because the major cost of developing this solution has already been made. Moreover, the company should change the encryption method that makes it more difficult for malware to steal data. We propose these options as the most important to implement:

- Passwords should almost always be stored as hashes of hash function (f.e. sha-256) not a plain text. Hashes are irreversible, so even if the attacker will find our database, he has to put in lots of effort to get plain text passwords.
- Security of personal data of customers is extremely important. Any potential leak can couse destructive consequences for whole company. As so database should be well protected.



```
LOCK TABLES `eklamot_users` WRITE;
/*!40000 ALTER TABLE `eklamot_users` DISABLE KEYS */;
INSERT INTO `eklamot_users` VALUES (1,"Adam","Wieczorek","adawieczorek68@eklamot.com",
"489193284889","1968-01-19","a12345","2022-02-16 13:28:08","2019-12-01 19:51:38","1"),(2,
"Jagoda","Pawlak","jagpawlak67@eklamot.com","486560820511","1967-10-14","123","2022-02-25
04:18:03","2018-10-21 06:51:16","1"),(3,"Gabriel","Kwiatkowski","gabkwiatkowski86@eklamot.com",
"486304543350","1986-06-16","loveme","2022-02-18 04:21:23","2020-02-02 18:32:12","1"),(4,
"Kajetan","Mazur","kajmazur93@eklamot.com","487690650856","1993-05-17","family","2022-02-17
17:15:03","2020-05-10 20:38:36","1"),(5,"Gabriel","Nowak","gabnowak98@eklamot.com",
"480186921584","1998-06-06","1q2w3e","2022-02-25 07:29:46","2019-06-06 09:46:00","1"),(6,
"Michalina","Pietrzak","micpietrzak86@eklamot.com","487577128915","1986-11-30","999999",
"2022-03-03 04:29:56","2020-08-26 11:25:45","1"),(7,"Kuba","Wieczorek","kubwieczorek75@eklamot.
com","489444280039","1975-09-14","qwerty1","2022-02-22 23:09:41","2018-03-29 00:32:41","1"),(8,
"Aleksandra","Majewska","alemajewska79@eklamot.com","483747663451","1979-12-13","thomas",
"2022-02-18 18:11:11","2018-06-14 00:14:27","1"),(9,"Alan","Wójcik","alawojcik82@eklamot.com",
"484844080318","1982-08-29","aaaaaa","2022-02-22 05:07:22","2021-06-12 03:49:14","1"),(10,
"Fabian","Sikora","fabsikora93@eklamot.com","488002667418","1993-07-23","baseball","2022-02-26
05:39:13","2021-01-23 09:45:53","1"),(11,"Borys","Tomaszewski","bortomaszewski73@eklamot.com",
"485518141222","1973-09-26","a123456","2022-02-25 02:01:33","2019-12-09 01:07:44","1"),(12,
"Jakub","Olszewski","jakolszewski68@eklamot.com","486056993470","1968-02-04","evite","2022-02-15
17:17:50","2020-01-16 15:40:50","1"),(13,"Blanka","Kamińska","blakaminska77@eklamot.com",
"483414658711","1977-05-01","football1","2022-02-19 20:48:53","2019-08-10 14:00:05","1"),(14,
```

Password Policy

A password policy is a set of rules designed to enhance the security of the user and the device they work on. Password policies are often part of official corporate policies and can be taught in cybersecurity training. Moreover, we would like to incorporate two more rules which would make it much more difficult for the malware or hackers to get into our accounts or systems:

- We should force changing the temporary password after the first login, not only kindly ask for it. Most of the employees just stick to the generic one which is a perfect opportunity for an easy attack.
- Adding two-factor authorization would be a great idea. Since almost everybody nowadays has a mobile device, authorization application or mobile authorization should not be much of a problem for workers of Eklamot.

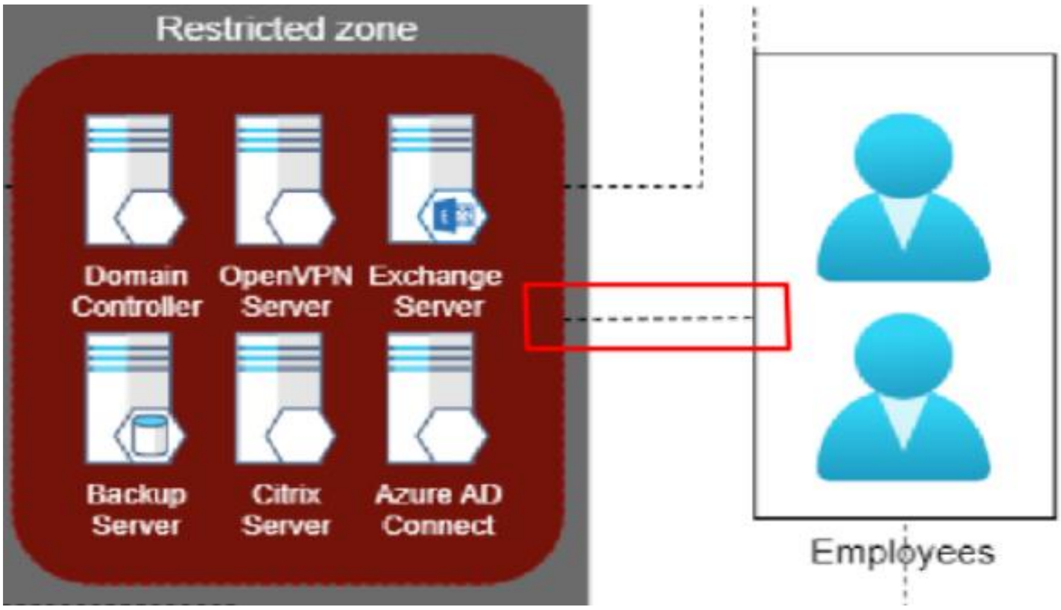
Architecture

In order to run the whole system without further problems, we would like to implement a few changes in the architecture of our project. Key changes would consist of:

- A backup server should be accessible only for admin employees.

Many employees have root privileges and can perform actions that can do big damage to the system. For this kind of situation, the company has a backup server. Only a few people should have access to this server in order to secure the system from accidental damage done by the employees.

- Employees should not have access directly to the restricted zone.



Here we have a connection between employees directly with the restricted zone, from their own computers. We cannot control what employees do or follow them. Thus we deleted this connection because now employees can access the restricted zone only from their business computer, which is followed by the company.

Also, we noticed that employees are connecting with workstations without any protections, so we add VPN Gateway which is shown by the image below.

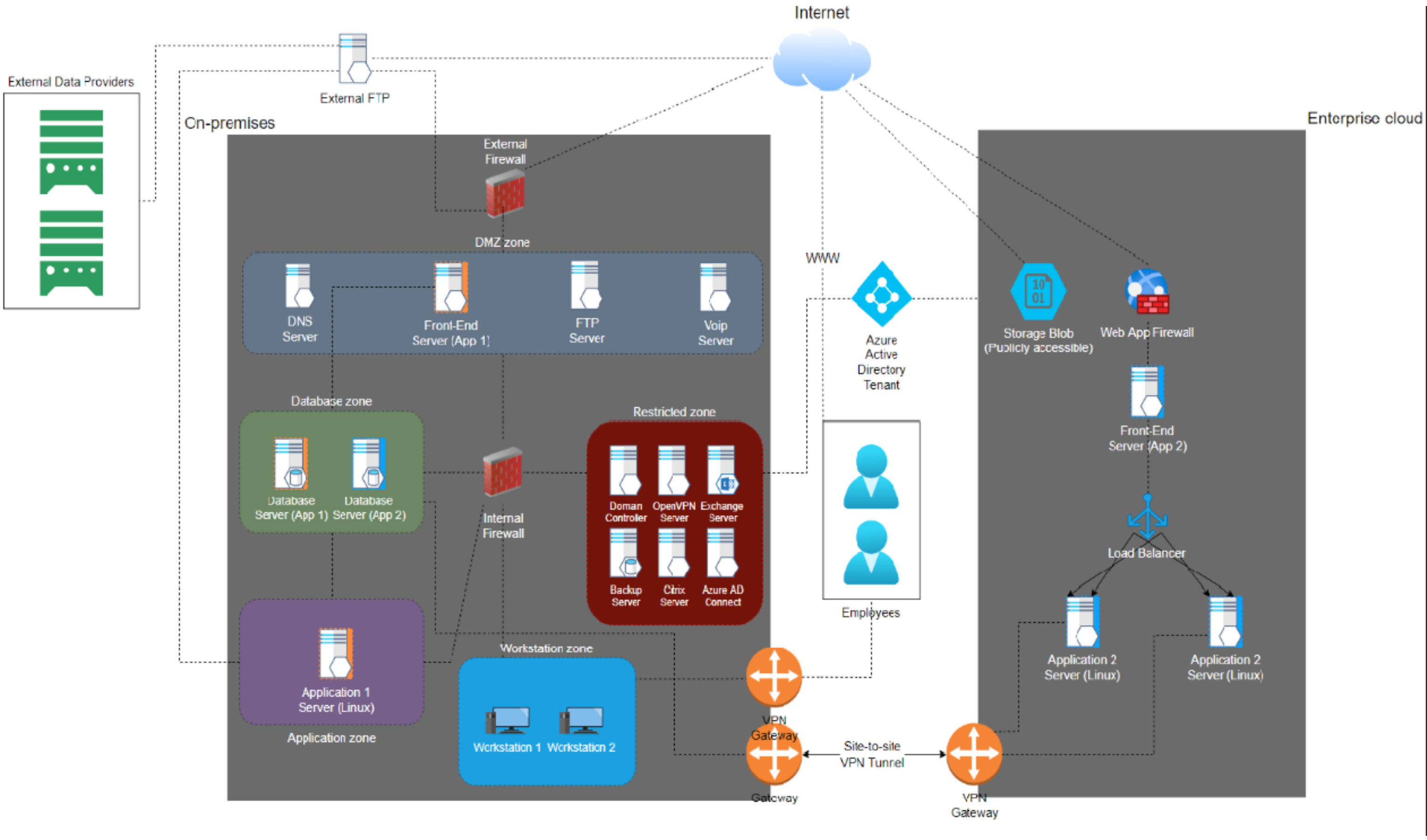
- A new firewall between the Front-End Server and Database Server.

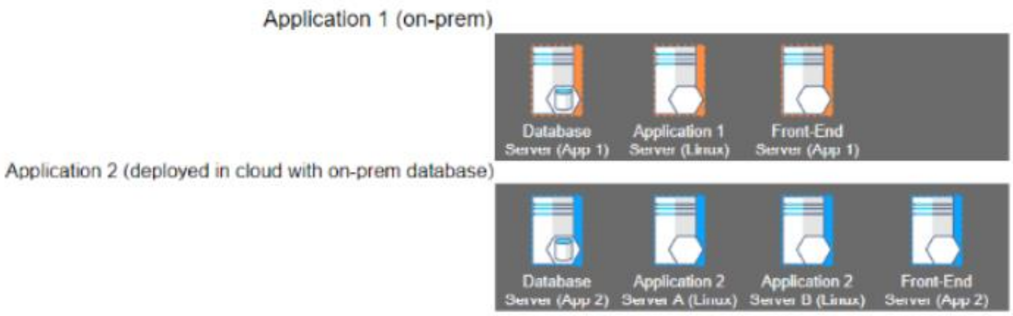
As attacks can be performed from the Front-end level (for example XSS, CSRF, or SQL injection attacks, which can be used to steal data from our database), we should protect our databases (which as we know have sensitive data!). That kind of protection can provide a firewall and for more safety, we decided to add another one, between the Front-end server and our database.

- Both Application2 should have access to database2.

Load balancer balance actions between these two applications. These applications are the same application so we think they both should be connected to VPN.

The graphical representation of how it should look like:




Application 1 (on-prem)
Database Server (App 1)
Application 1 Server (Linux)
Front-End Server (App 1)
Application 2 (deployed in cloud with on-prem database)
Database Server (App 2)
Application 2 Server A (Linux)
Application 2 Server B (Linux)
Front-End Server (App 2)

Changes to the vulnerability administration process in Eklamot

Changes we suggest:

Security scanners are not ideal. We suggest using both famous Nessus and Nexpose for better quality. Moreover even both can miss some dangerous vulnerabilities. As so we insist on sending those raports directly to security team. Security team would then extend reasearch.

Final Process:

1. Prepare a scan request - If there is a need for scanning, the request is made by the System Administrator.
2. Preparation and execution of the scan with Nessus and Nexpose (complementary tools).
3. Provide the results of the scan to the Security Team.
4. The Security Team deepens the vulnerability search.
5. If vulnerabilities are found, they are forwarded to the system administrator with a report describing them.
6. Is a deviation required? A decision point that determines if a deviation is required within the vulnerabilities found. The system administrator reviews and decides whether a derogation request is required.
7. Prepare a deviation request - If the system administrator decides a deviation is required, the system administrator prepares an appropriate request with justification, which it sends to the Security Team.
8. Perform corrective action - If no deviation request is required, the System Administrator proceeds to implement the planned corrective action.
9. Report completion of corrective action - Upon completion of the action, the System Administrator reports its completion to the Security Team.
10. Has the variance request been approved? - A decision point that determines whether the deviation request has been accepted by the Security Team. If the request is accepted, the Security Team prepares an appropriate summary. If the request is not accepted, the system administrator must take corrective action.
11. Prepare Summary - The Security Team prepares an appropriate summary depending on whether the process resulted in the implementation of corrective actions to address the vulnerabilities discovered or the acceptance of the deviation request.