Final task

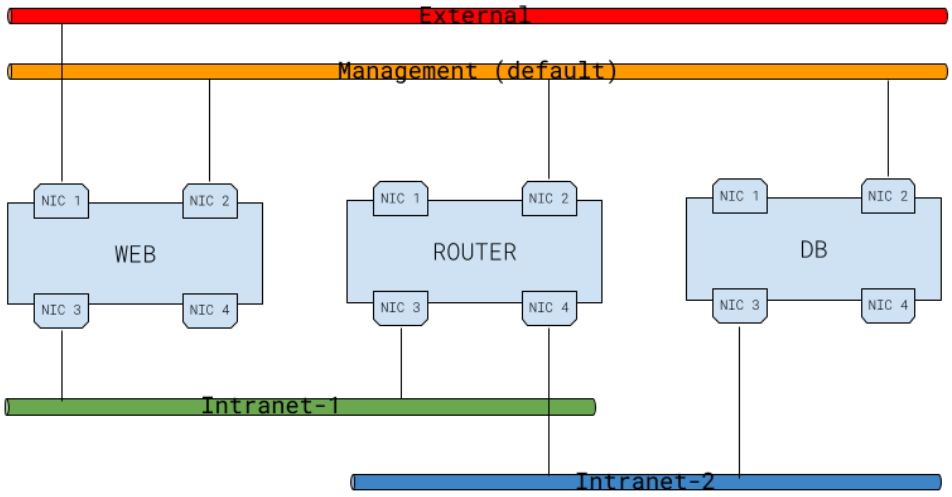
1. Deploy three Linux machines (you can use any convenient virtualization tools and hardware), divide them into the following roles: ROUTER, WEB, & DB
2. It is desirable but not required to use operating systems of the same vendor
3. Configure Linux machines and the network according to the requirements
4. Complete the story using your out-of-the-box infrastructure

Presentation and defense

1. A presentation (.ppt or other similar presentation format) must be prepared in English which contains description of the solution
2. The presentation must be provided and sent to a commission before a demo session
3. On the demo session you will present your solution using the presentation, live demo and answer the commission's questions. The defense could be hold either in English or in Russian.

Infrastructure requirements

1. Services and applications must communicate seamlessly with each other
2. Applications also must be available from the host



Linux & Networking

1. Three Linux machines must be created and configured
2. Last mainline kernel installed
3. Local timezone configured
4. Control plane remote access to machines allowed through SSH and authentication with SSH keys only
5. Operating system (OS) services and processes must use DNS names for inter-VM (virtual machine) communication.
6. There are three networks that must be configured and one network that is optional:
    a. External – network that simulates Internet, the application User Interface (UI) published here.
    b. [Optional] Management (default) – to manage WEB/ROUTER/DB VMs and their services.
    c. Intranet-1 – to provide connectivity between WEB and ROUTER VMs
    d. Intranet-2 – to provide connectivity between ROUTER and DB VMs.
    e. Network access should be secured/hardened:
        i. Only trusted/approved connections can be established (see Router requirements below).
        ii. Access lists should be documented and demonstrated.
        iii. Unused TCP/UDP ports and services must be disabled.

f. [Optional] Simulate latency between VMs.

7. Configure mdraid, lvm.зне
    a. /local/files directory should be mounted on lvm partitions on DB server (2 extra block devices added to a server)
    b. /local/backups directory should be mounted on mdraid (RAID1/mirror) partition on DB server (2 extra block devices added to a server)

Router

One of the VMs created should serve the ROUTER role. The purpose of this VM is to emulate simple IP-router behavior.

Requirements:

- IP packets with source/destination IP addresses from Intranet-1 and Intranet-2 should be routed and forwarded
- Other packets should be dropped, hit counters demonstrated.

Monitoring

1. Basic Linux host resource utilization:
    - CPU
    - Mem
    - Disk (IOPs)
    - [Optional] accounting events, system errors using Elasticsearch stack
2. Network monitoring (ROUTER only). An engineer should be able to easily access and review these networking stats:
    - TCP sessions established
    - Packet counters (IN/OUT/DROPPED/ERRORs)
    - Packets forwarded/routed
    - Utilization:
        - broadcast
        - unicast
        - multicast
    - NAT sessions if exist
    - [Optional] Retrospective (history must be stored/visualized using Grafana or any other appropriate tool)

DB

1. Database and other user with “production” privileges created
2. Four tables with relationship created and filled with data
3. DB application should be available from host machine for any GUI DB clients (pgadmin, dbeaver, etc.)
4. One of the table values imported from a file
5. Tables and data

Articles			
id	**magazines_id**	**article_type_id**	**author_id**
1	1	2	3
2	3	3	2
3	2	2	4
4	1	1	1

magazines	
id	**name**
1	it herald
2	IT STORIES
3	IT with kids

article_types	
id	**type**
1	news
2	tech
3	entertainment

author	
id	**author**
1	Chappie
2	Wall-e
3	Atom
4	T1000

Web

1. Web server application installed (apache, nginx, etc.)
2. End user should be able to connect to the application as follows:
   - Open web-browser on the host (hypervisor) machine
   - Enter URI as follows:
     <scheme>://<authority>/<path>/<query>/<fragment>, where
     - scheme: **http**
     - authority: **host:port** from “External network”
     - path, query, fragment: according to the application designed
   - Application front end should appear in the browser tab
3. Data shown at UI should be synchronized with the Database
4. Static “Hello World” web page should be written and available from DB, ROUTER servers
5. [Optional] Files in /local/files/* should be available from DB, ROUTER via http (https) protocols (read-only)

Bash scripts

Script1

Write a bash script that retrieves data about articles from the DB.

- Received data should be stored as separate files in /local/files directory (one script launch – one data file)
- The script should also compress and move files to /local/backups when there are more than 3 files in /local/files

Script2

- Script2 works in detached mode and starts at system boot
- Script2 has /var/run/script_name.pid (should not be run by a second instance)
- Script2 Must be configurable (ENV, configuration, etc.)
- Script2 checks /local/backups and sends mail to root according to its configuration (a or b points):

- number of files in /local/backups directory is more than X
- total size of /local/backups directory is more than Y bytes

Python/Go script

1. Must be added to CRON (runs every N minute)
2. The script is located on WEB server /local/scripts and receives data from DB with SELECT query
3. The script generates a static HTML web page from the given data
4. Received data must be added as tags (<p>, , etc.) to the <body> block
5. Generated HTML page should be served by WEB server

The story: Attack (MITM)

You need to complete this story on your infrastructure. For the correct execution of the story, it is required to comply with all the requirements specified above.

1. Use DB machine to receive data from the WEB server (curl, etc.), preliminary add SRC ip:port, DST ip:port to the corresponding Access Rules (see Linux & Networking 6.e.ii)
2. Capture the traffic (see point 1) on the ROUTER machine (tcpdump, etc.)
3. After successful intercepting of the page content, configure WEB server to encrypt data with SSL/TLS
4. [Optional] Intercept data, implement MITM attack (assumption: it is possible to update “trusted root CAs on the DB server side)