(/)

Curriculum

SE Foundations

Average: **118.92%**

New server created!

0x10. HTTPS SSL

DevOps **SysAdmin** **Security**

Weight: 1

Project over - took place from Apr 4, 2024 4:00 AM to Apr 5, 2024 4:00 AM

An auto review will be launched at the deadline

In a nutshell...

- **Auto QA review:** 0.0/8 mandatory & 0.0/1 optional
- **Altogether: 0.0%**
  - Mandatory: 0.0%
  - Optional: 0.0%
  - Calculation: 0.0% + (0.0% * 0.0%) == **0.0%**

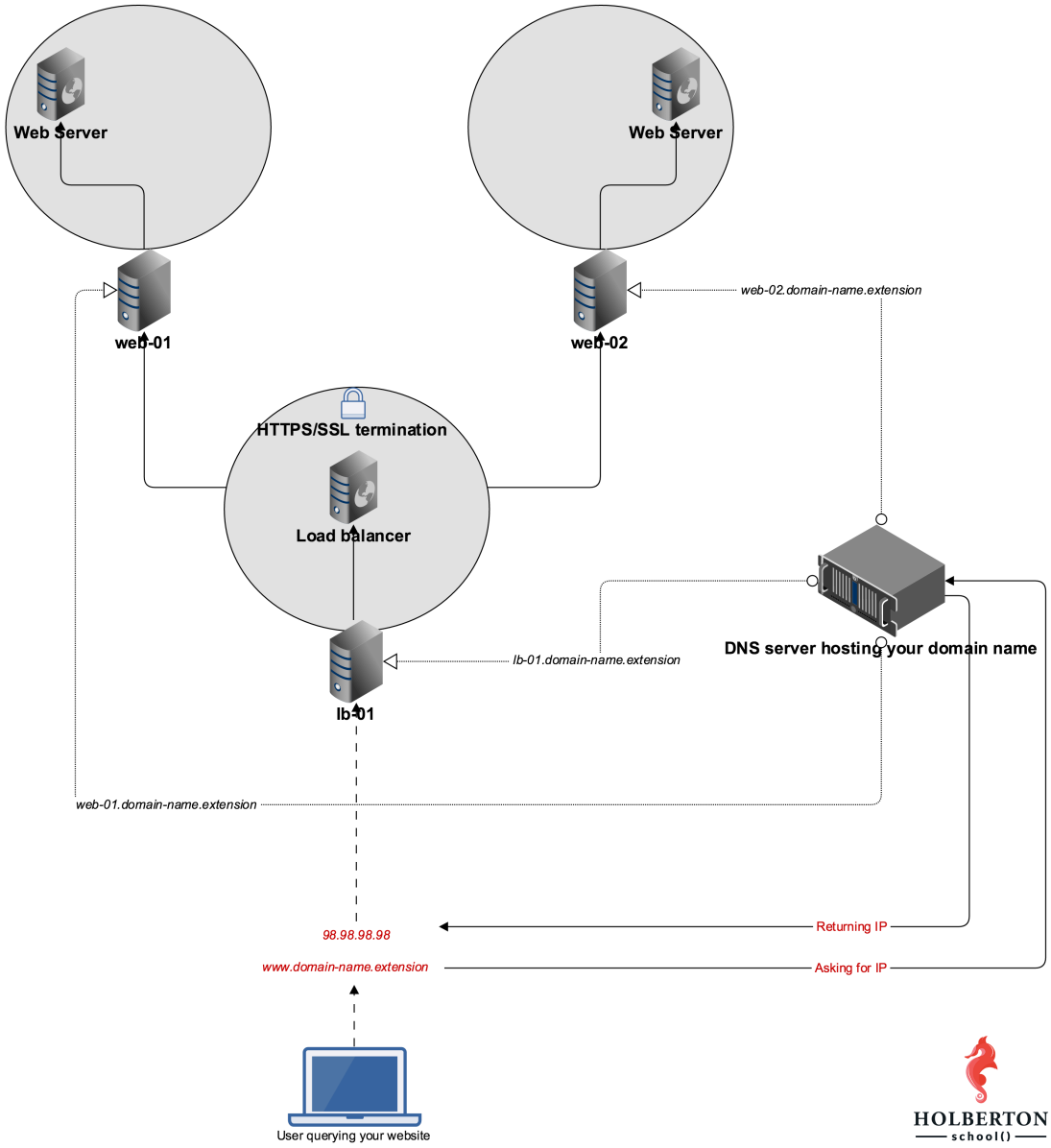
Concepts

For this project, we expect you to look at these concepts:

- DNS (/concepts/12)
- Web stack debugging (/concepts/68)

(/)

HTTPS



Background Context

What happens when you don't secure your website traffic?

Resources

Read or watch:

- What is HTTPS? (/rltoken/XT1BAiBL3Jpq1bn1q6lYXQ)
- What are the 2 main elements that SSL is providing (/rltoken/STj5WkAPACBxOvwB77Ycrw)
- HAProxy SSL termination on Ubuntu16.04 (/rltoken/asrMHTWJxWQ2x-Sn6snHow)
- SSL termination (/rltoken/CKUICfppIWI6UC0coEMB8g)
- Bash function (/rltoken/zPjZ7-eSSQsLFsGA16C1HQ)

man or help:

- `awk`
- `dig`

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/rltoken/fJ20wsMngb_yNAhGgBwzlQ), **without the help of Google**:

General

- What is HTTPS SSL 2 main roles
- What is the purpose encrypting traffic
- What SSL termination means

Requirements

General

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be interpreted on Ubuntu 16.04 LTS
- All your files should end with a new line
- A `README.md` file, at the root of the folder of the project, is mandatory
- All your Bash script files must be executable
- Your Bash script must pass `Shellcheck` (version `0.3.7`) without any error
- The first line of all your Bash scripts should be exactly `#!/usr/bin/env bash`
- The second line of all your Bash scripts should be a comment explaining what is the script doing

Quiz questions

Great! You've completed the quiz successfully! Keep going! (Hide quiz)

Question #0

Why do you need HTTPS?

(/)

- (•) To encrypt all communication between the client and the website servers
- ( ) To encrypt credit card and social security number information going between the client and the website servers
- ( ) To accelerate the communication between the client and the website servers

Question #1

You want to setup HTTPS on your website, where shall you place the certificate?

- ( ) You can host it anywhere but you have to share the link to it on your website
- ( ) In a secure location where nobody can access it
- (•) On your website web server(s)

Question #2

What is HTTPS?

- ( ) A faster version of HTTP
- (•) A secure version of HTTP
- ( ) A superior version of HTTP

Your servers

Name	Username	IP	State	
521277-web-01	`ubuntu`	`54.144.130.32`	running	Actions
521277-web-02	`ubuntu`	`54.158.81.249`	pending	Actions
521277-lb-01	`ubuntu`	`107.23.116.189`	pending	Actions

Tasks

0. World wide web

mandatory

Score: 0.0% (*Checks completed: 0.0%*)

Configure your domain zone so that the subdomain `www` points to your load-balancer IP (`lb-01`). Let's also add other subdomains to make our life easier, and write a Bash script that will display information about subdomains.

Requirements:

- Add the subdomain `www` to your domain, point it to your `lb-01` IP (your domain name might be configured with default subdomains, feel free to remove them)
- Add the subdomain `lb-01` to your domain, point it to your `lb-01` IP
- Add the subdomain `web-01` to your domain, point it to your `web-01` IP
- Add the subdomain `web-02` to your domain, point it to your `web-02` IP
- Your Bash script must accept 2 arguments:
    1. `domain` :
        - type: string
        - what: domain name to audit
        - mandatory: yes
    2. `subdomain` :
        - type: string
        - what: specific subdomain to audit
        - mandatory: no
- Output: `The subdomain [SUB_DOMAIN] is a [RECORD_TYPE] record and points to [DESTINATION]`
- When only the parameter `domain` is provided, display information for its subdomains `www` , `lb-01` , `web-01` and `web-02` - in this specific order
- When passing `domain` and `subdomain` parameters, display information for the specified subdomain
- Ignore `shellcheck` case `SC2086`
- Must use:
    - `awk`
    - at least one Bash function
- You do not need to handle edge cases such as:
    - Empty parameters
    - Nonexistent domain names
    - Nonexistent subdomains

Example:

```
sylvain@ubuntu$ dig www.holberton.online | grep -A1 'ANSWER SECTION:'
;; ANSWER SECTION:
www.holberton.online.   87  IN  A   54.210.47.110
sylvain@ubuntu$ dig lb-01.holberton.online | grep -A1 'ANSWER SECTION:'
;; ANSWER SECTION:
lb-01.holberton.online. 101 IN  A   54.210.47.110
sylvain@ubuntu$ dig web-01.holberton.online | grep -A1 'ANSWER SECTION:'
;; ANSWER SECTION:
web-01.holberton.online. 212    IN  A   34.198.248.145
sylvain@ubuntu$ dig web-02.holberton.online | grep -A1 'ANSWER SECTION:'
;; ANSWER SECTION:
web-02.holberton.online. 298    IN  A   54.89.38.100
sylvain@ubuntu$
sylvain@ubuntu$
sylvain@ubuntu$ ./0-world_wide_web holberton.online
The subdomain www is a A record and points to 54.210.47.110
The subdomain lb-01 is a A record and points to 54.210.47.110
The subdomain web-01 is a A record and points to 34.198.248.145
The subdomain web-02 is a A record and points to 54.89.38.100
sylvain@ubuntu$
sylvain@ubuntu$ ./0-world_wide_web holberton.online web-02
The subdomain web-02 is a A record and points to 54.89.38.100
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x10-https_ssl`
- File: `0-world_wide_web`

Done? | Check your code | Ask for a new correction | Get a sandbox | QA Review

1. HAproxy SSL termination

mandatory

Score: 0.0% (*Checks completed: 0.0%*)

"Terminating SSL on HAproxy" means that HAproxy is configured to handle encrypted traffic, unencrypt it and pass it on to its destination.

Create a certificate using `certbot` and configure `HAproxy` to accept encrypted traffic for your subdomain `www.`.

Requirements:

- HAproxy must be listening on port TCP 443
- HAproxy must be accepting SSL traffic
- HAproxy must serve encrypted traffic that will return the `/` of your web server
- When querying the root of your domain name, the page returned must contain `Holberton School`

- Share your HAproxy config as an answer file (`/etc/haproxy/haproxy.cfg`)

(/)

The file `1-haproxy_ssl_termination` must be your HAproxy configuration file

Make sure to install HAproxy 1.5 or higher, SSL termination (/rltoken/CKUICfppIWI6UC0coEMB8g) is not available before v1.5.

Example:

```
sylvain@ubuntu$ curl -sI https://www.holberton.online
HTTP/1.1 200 OK
Server: nginx/1.4.6 (Ubuntu)
Date: Tue, 28 Feb 2017 01:52:04 GMT
Content-Type: text/html
Content-Length: 30
Last-Modified: Tue, 21 Feb 2017 07:21:32 GMT
ETag: "58abea7c-1e"
X-Served-By: 03-web-01
Accept-Ranges: bytes
sylvain@ubuntu$
sylvain@ubuntu$ curl https://www.holberton.online
Holberton School for the win!
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x10-https_ssl`
- File: `1-haproxy_ssl_termination`

☐ Done? | Check your code | Ask for a new correction | >_ Get a sandbox | QA Review

2. No loophole in your website traffic

#advanced

Score: 0.0% (*Checks completed: 0.0%*)

A good habit is to enforce HTTPS traffic so that no unencrypted traffic is possible. Configure HAproxy to automatically redirect HTTP traffic to HTTPS.

Requirements:

- This should be transparent to the user
- HAproxy should return a 301 (/rltoken/yGdTSvZAzHMnDEhalTjNUw)
- HAproxy should redirect HTTP traffic to HTTPS
- Share your HAproxy config as an answer file (`/etc/haproxy/haproxy.cfg`)

The file `100-redirect_http_to_https` must be your HAproxy configuration file

Example:

(7)

```
sylvain@ubuntu$ curl -sIL http://www.holberton.online
HTTP/1.1 301 Moved Permanently
Content-length: 0
Location: https://www.holberton.online/
Connection: close

HTTP/1.1 200 OK
Server: nginx/1.4.6 (Ubuntu)
Date: Tue, 28 Feb 2017 02:19:18 GMT
Content-Type: text/html
Content-Length: 30
Last-Modified: Tue, 21 Feb 2017 07:21:32 GMT
ETag: "58abea7c-1e"
X-Served-By: 03-web-01
Accept-Ranges: bytes

sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x10-https_ssl`
- File: `100-redirect_http_to_https`

Done? | Check your code | Ask for a new correction | Get a sandbox | QA Review