Západočeská univerzita
Fakulta aplikovaných věd
Katerdra Kybernetiky

Dokumentace distribuci projeku ITE-YELLOW

XXXXXXXXX

1 Úvod

V rámci předmětu KKY/BSOI jsme si vytvořili a nastavili virtuální stroj s operačním systémem Linux. Tento stroj je momentálně používán k provozu aplikace vyvíjené v rámci předmětu KKY/ITE. Zde je na aplikaci a zde je odkaz na Githubový repozitář, kde jsou k dipozici veškeré zdojové kódy aplikace jak části ITE, tak části BSOI. Tento referát by měl popsat virtuální stroj, jeho nastavení, činnosti a realizaci způsobů přihlášení do již zmíněné aplikace.

2 Řešení

2.1 Vlastnosti virtuálního stroje

Při zakládání stroje (sulis162, 147.228.173.162) byly zvoleny tyto parametry; 1 procesor se 2 vlákny, 2 GB operační paměťi a 20 GB paměti pevného disku. Na stroji běží operační systém Debian 11.6.

2.2 Uživatelské účty

Na virtuálním stroji jsou tři uživatelské účty - každý pro jednoho člena týmu (martin, skokoska a radek). Tyto účty mohou používat sudo a mohou se na stroj přihlásit přes ssh bez hesla díky jejich veřejným ssh klíčům. Přímé přihlášení na root má povoleno jen zakladatel (skokoska).

2.3 Monitoring

Dva členové týmu využil pro monitoring služeb UptimeRobot (https://uptimerobot.com/), kde byl nastaven ping na IP adresu virtuálního stroje každých 5 minut. V výpadku nás služba upozorní emailem.

2.4 Zálohování

Zálohování databáze bylo implementováno pomocí nástroje Borgbackup, specificky funkcí prune pro automatizaci celého procesu k ukládání každodenní zálohy na účty skokoska a radek, v případě poruchy či potřeby vrátit databázi do předchozího stavu tedy existují dvě zálohy.

Na každém účtě je složka 'backup' obsahující zašifrovanou zálohu pod defaultním heslem 'gobackImadeamistak'.

2.5 Šifrování

Certifikáty pro šifrování pomocí TLS (Transport Layer Security) byly vystaveny od autority 'Let's encrypt' a byl použit šifrovací klíč ECC. Certifikát je platný 90 dní a má celkem tři části:

1. key.pem - obsahuje soukromý klíč, který je pužit k dešifrování dat
2. cert.pem - obsahuje certifikát serve ru
3. fullchain.pem - řetězec důvěry

.

2.6 Zabezpečení virtuálního stroje

Samotnáwebová aplikace je přístupná pouze ze sítě ZČU a to na portu 442, zde také v našem případě probíhá HTTPS komunikace. Poté bylo třeba otevřít port 80 (HTTP) kvůli vystavení TLS certifikátu. Žádné jiné porty otevřené nejsou.

2.7 Provoz distribuované aplikace

Součástí aplikace je PostgreSQL databáze a dva python skripty, jeden z nich řeší MQTT subscribera, abychom mohli data přijímat, validovat a ukládat a druhý tvoří backend naší webové aplikace. K založení PostgreSQL databáze byl vytvořen python skript, který stačí spustit pouze jednou, aby vytvořil tabulky. Skripty na subscribera a backend musí běžet neustále, aby aplikace mohla být v provozu. Subscriber a backend spolu 'komunikují' pouze přes databázi, ke které mají oba skripty přístup.

Spouštění skriptů subscriberu a backendu po pádu a logování výstupů zprostředkovávají 'launchery' - python skripty, které ty původní provozují v nekonečném cyklu s try-except blokem a přesměrovávají jejich stdout (výstup na konzoli) do textových souborů. Launchery samotné jsou spuštěny pomocí systemd služeb, které jsou nastaveny, aby se spouštěly při zapnutí/resetu virtuálního stroje. Z důvodu velkého množství výpisů subscribera jsou jeho výpisy uchovávány jen z posledních dvou dní v `.txt` souborech po hodinách. O to se stará separátní kombinace python skriptu a systemd služby. Nasazení verzí se může provést pomocí programu WinSCP, kterým se nové soubory nahrají na virtuální stroj.

2.8 Přihlášení a biometrická autentizace

Při implementaci byly využity zdroje dodané vyučujícími ve formě skriptů `faceid_server.py`, `extract_embeddings.py`a `train.sh`.

Rozšíření backendu o FaceID
Ve skriptu `faceid_server.py` se nacházejí dva Tornado handlery, první zajišťující komunikaci s frontendem v rámci posílání a ukládání fotek a druhý v rámci detekci obličeje ve fotce. Původně byly oba handlery provozovány v rámci jedné URL adresy, ale eventuelně byly rozděleny. Aplikace funguje následovně:

1. Nepřihlášený uživatel se může buď přihlásit přes uživatelské jméno a heslo (vstupní stránka), přihlásit se přes FaceID nebo se zaregistrovat.
2. Po registraci musí uživatel čekat na schválení adminem, jinak se nepřhlásí (ani v případě, že již fotky má uložené v rámci biometické autentizace).
3. Přihlášený uživatel má přístup k celé aplikaci, včetně možnosti nafotit si své fotky pro FaceID a natrénovat model pro autentizaci obličejů.

Uživatelské účty a identity
Je dostupné celkem 9 uživatelských účtů pro příhlášení, konkrétně pro:

- 3 vyučující (Jan Švec, Vlasta Radová, Martin Bulín)
- všechny 3 členy týmu
- 1 admin účet, ke kterému mají přístup všichni členové týmu

Biometrických identit je v době psaní tohoto dokumentu 7. Jsou vytvořené pro všechny 3 členy týmu, jedna pro kategorii 'unknown' (aby se 'nové' tváře nemohly přihlásit, jinak by je systém identifikoval jako validního uživatele) a 3 identity učitelů. Kategorie 'unknown' má celkem 17 fotek, z toho 3 neobsahují lidský obličej, aby se neuronová síť naučila přiřazovat neplatné fotografie neautorizovanému uživateli. Každý z členů má 10 až 25 fotek, které si každý sám nafotil. Od každého vyučujícího jsme dostali 10 - 25 fotek. Každý s různým počtem, aby šlo testovat úspěšnost identifikace. Rozhodovací práh je nastaven na alespoň 80% pravděpodobnost identifikace uživatele. Byl určen testováním členy týmu za rozumnou hranici mezi bezlečností a přihlášením. Je nutné podotknout, že identita 'unkown' není vůbec uložená v databázi a nemá možnost se do aplikace přihlásit.

Funkčnost přihlašování
V případě, že se uživatel přihlásí, je mu nastaven Json Web Token (JWT). Ten řeší zda je uživatelská session validní či nikoliv. Pokud jsou uživatelské credentials v pořádku, uživatel je zaregistován a schválený adminem, je mu JWT uložen jako secure cookie. Všechny další žádosti nyní automaticky kontrolují zda JWT neexpiroval nebo s ním nebylo manipulováno. JWT má nastavenou dobu expirace jedné hodiny a obsahuje 'signaturu' - tajný klíč, který je znám pouze serveru.

Odhlašování
Pro odhlašování byl vytvořen Tornado handler, který vymaže token obsahující cookie a přesměruje uživatele zpět na přihlašovací stránku.

3 Závěr

Všechny položky zadání byly zodpovězeny.