ŠKOLITEĽSKÝ POSUDOK NA BAKALÁRSKU PRÁCU

Kamila Součková:
Design and implementation of an RFID access control system

KATEDRA INFORMATIKY, FMFI UK BRATISLAVA, JÚN 2016

Cieľom práce bolo navrhnúť a implementovať server pre prístupový systém Deadlock vyvíjaný študentským vývojovým tímom. Autorka, ako členka tohto tímu, mala navrhnúť server, ktorý bude udržiavať databázu identifikátorov RFID prístupových kariet a implementovať navrhnuté riešenie v jazyku Python 3. Návrh mal obsahovať najmä architektúru servera, dátové štruktúry pre efektívne uloženie a spracovanie prístupových práv, komunikačný protokol so zariadeniami riadiacimi prístup do chránených priestorov, riešenie aktualizácie firmvéru zariadení pripojených k serveru a zaznamenávanie prístupu do chránených priestorov.

Bakalárska práca je písaná po anglicky a má 44 strán. Pozostáva z úvodu, siedmich kapitol, záveru, slovníka použitých termínov, prílohy so zdrojovým kódom (odkaz na Github a CD) a zoznamu literatúry. Prevedením a vzhľadom pôsobí kultivovane. V prvej kapitole autorka vysokoúrovňovo popisuje požiadavky na systém, ako napríklad spoľahlivosť, bezpečnosť, dostupnosť, rozšíriteľnosť, jednoduchosť vývoja, použitia, nasadenia a údržby.

Nasleduje štvorstranová kapitola „prehľad systému“, ktorá je rozdelená na niekoľko častí. Prvá časť je venovaná prehľadu existujúcich systémov, ktorý je ale všeobecný, bez zmienky konkrétnych systémov. V nasledujúcej časti autorka spomína hlavné princípy návrhu: modularita, princíp najmenšieho „prekvapenia“, a to, že (citujem) „stav je škaredý“. Ďalšia časť je venovaná hlavným komponentom celého systému Deadlock, z ktorého je implementovaný server iba jednou časťou. Štvrtá časť zavádza pojem prístupových pravidiel. Má však iba tri riadky a odkazuje na kapitolu 4, ktorá túto tému podrobnejšie rozoberá. Piata časť sa na necelej strane zaoberá technickými výzvami ako spoľahlivosť, bezpečnosť, jednoduchosť nasadenia a údržby. Pravdepodobne sa mala zaoberať aj škálovateľnosťou riešenia, ale z tohto bodu ostala len nedokončená veta „Deadlock should scale to“.

V tretej kapitole autorka na 13 stranách popisuje jeden z hlavných výsledkov svojej práce a to návrh komunikačného protokolu medzi serverom a zariadeniami ovládajúcimi jednotlivé prístupové body. Musím kladne hodnotiť elegantnosť, až jednoduchosť navrhnutého riešenia, ktoré vďaka bezstavovosti a idenpotentnosti má potenciál jednoducho zvládnuť aj živý prechod na novšiu verziu protokolu, servera, či zariadení.

Štvrtá kapitola popisuje na piatich stranách formát prístupových pravidiel. Prekvapilo ma, že sa v nej vôbec nenachádza popis špecifikácie času, v ktorom prístupové pravidlá platia. Nasledujúca kapitola na 4 stranách popisuje návrh servera a jeho troch častí: „deadserver“ (komunikácia so zariadeniami), „deadapi“ (HTTP API pre rôzne rozhrania) a „deadaux“ (vytváranie databáz pre zariadenia a jednoduchý echo test). V predposlednej kapitole je stručne popísaná implementácia s entitno-relačným diagramom väčšej časti databázy. V tejto kapitole by som ocenil aspoň nejaký pseudokód popisujúci aspoň niektoré zaujímavé myšlienky (napríklad predvýpočet „in expression“). Keďže ide o stromovú štruktúru, bolo by dobré ju znázorniť aj na obrázku a uviesť príklad výpočtu. Posledná kapitola popisuje plány do budúcnosti.

Otázky a pripomienky

Strana 15

PING OK response: … The controller is expected to adjust its clock to match the server time.

Súčasne na strane 12 píšete, že:

... under normal circumstances the server/controller communication is not latency-sensitive, so the round-robin retries approach does not pose a latency problem.

Zohľadňuje sa táto latencia pri nastavovaní času, keďže v tomto prípade by to mohlo viesť k problémom s presnosťou?

Strana 14

As all communication must be initiated by the controller ...

Ako by ste hodnotili možnosť rozšírenia implementácie o zaslanie príkazu zo servera na všetky zariadenia v prípade núdze, napríklad na odomknutie všetkých dverí v prípade požiaru?

Pri špecifikácii protokolu by som ocenil podrobnejší popis ako aj rozobratie okrajových prípadov:

Strana 16 – 3.4.3 XFER: transfer a file chunk

Napríklad *offset* a *length* by mohli mať v popise uvedené, v akých sú jednotkách. Čo sa stane, ak *offset* bude mimo rozsah? V odpovedi **XFER OK,** ak je *length* rovné 0, tak *chunk* je pole bytov dĺžky nula alebo vôbec nie je prítomný?

Strana 17 – 3.4.4 CRITICAL: report a critical problem

Zapisujú sa tieto udalosti aj do logov? Ak áno, nestačí, aby server prijal túto udalosť ako záznam v logu? Ak nie, čo sa stane, ak nie je možné nadviazať spojenie so serverom v čase vzniku udalosti?

Strana 18 – 3.4.6 ECHOTEST: echo for testing purposes

Echoes the request body.

Čo presne sa chápe pod „request body“ (t.j. aká časť správy)?

Strana 22 – Security (poznámka pod čiarou)

... nonces may be predictable ... the only requirement is a uniform distribution to ensure low collision probability.

Je teda možné zvoliť *nonce* ako sekvenčnú postupnosť inkrementovanú zakaždým o 1?

Strana 23 – Access rules

Píšete, že systém bude podporovať vysoko aj nízko-úrovňové pravidlá. Každý systém si môže zaviesť svoje vlastné vysoko-úrovňové pravidlá, ktoré sa realizujú pomocou jednej množiny nízko-úrovňových pravidiel. Systém nesmie ignorovať jemu neznáme množiny nízko-úrovňových pravidiel. Ako s nimi má ale pracovať? Môže ich editovať?

Strana 24

Every access point is ... Find this AP's type ...

V práci ste na strane 2 zaviedli termín „points of access“, v skratke PoA. Ale tu používate termín „access point“ a skratku „AP“.

Strana 25

Kde v práci popisujete, čo je priorita? V tejto časti ju používate, ale myslím, že až zo zdrojových kódov som sa dozvedel, že je to cele číslo a že pre daný typ PoA musí byť jedinečné, t.j. že dve pravidlá pre ten istý typ PoA nemôžu mať rovnakú prioritu.

Strana 28

In order to fulfill the protocol idempotence guarantee, only logs with a unique combination of attributes are stored.

Ako koexistuje toto riešenie s možnosťou korekcie času? Môže sa teoreticky stať, že dva následné prístupy toho istého človeka do jedného PoA sa na serveri uložia iba ako jeden prístup?

Strana 30 – deadapi

Provides the HTTP API used by the web management and monitoring interface, and the provided commandline interface.

Kde je popísané toto HTTP API a spomínané rozhranie pre príkazový riadok?

Strana 34 – entitno-relačné diagramy

V práci nepopisujete, prečo ste zaviedli tabuľku *identity*? Nemôže byť *card* použité priamo v tabuľke *identity_expr_edge*? Myslíte si, že by sa dala spojiť tabuľka *accesspoint* a *controller* do jednej (keďže je medzi nimi väzba 1:1)? Opäť nepoužívate termín „points of access“. Keďže jeden PoA môže mať viacero pripojených čítacích zariadení, nebolo by vhodné do tabuľky *accesslog* pridať aj stĺpec *reader*? Prečo tento diagram nepopisuje aj ostatné tabuľky v systéme, napríklad *ruleset*?

Strana 35

Upon change, it traverses the identity expression DAG recursively, marking what needs recomputing in the auxiliary _mr_recalculate table.

Myslím si, že by si tento algoritmus zaslúžil podrobnejší popis, možno aj s nejakým obrázkom a príkladom výpočtu.

Strana 36

… derive the file name from the contents: we compute the 32-bit FNV-1a hash while writing the file and use that as the version.

V práci vôbec nepopisujete, ako vyzerá výsledné meno súboru (kde v ňom sa nachádza verzia a čo tvorí zvyšok). Táto hodnota je tým celým číslom, ktoré sa ako verzia posiela zo servera na zariadenie? Píšete, že ste si zvolil algoritmus FNV-1a, lebo v 32-bitovej verzii má malú pravdepodobnosť kolízie a dobrý výkon. Porovnávali ste ho aj s inými algoritmami?

tags.py

Definujete tu viacero konštánt, ktorých použitie z práce nie je zrejmé, napr.: CONFIG_MAC, DUMMY.

serializable.py

the_shitty_encode_that_doesnt_handle_Tag = tagmapper.encode

Navrhujem väčšiu konzervatívnosť pri pomenovávaní funkcií a premenných.

Záver

Autorka má zjavne schopnosť spracovať zadaný problém a navrhnúť vlastné riešenia. Je však škoda, že na práci pracovala až príliš samostatne, a asi aj „nárazovito“, kvôli čomu ostalo menej priestoru na vylepšenie funkčnosti aplikácie a doladenie textu práce. Ako aj sama autorka píše, priložený zdrojový kód neimplementuje časti HTTP API, predvýpočet „identifikačných výrazov“ nie je inkrementálny a podobne.

Prácu navrhujem hodnotiť známkou **C – dobre (bežná spoľahlivá práca)**.

V Bratislave, 21. 6. 2016

.....
RNDr. Richard Ostertág, PhD.
školiteľ bakalárskej práce
Katedra informatiky, FMFI UK Bratislava