Project 4 - Centrale Bank

Rick van Vonderen
0945444
TI1C

23 mei 2018

Inhoudsopgave

1 Inleiding

Om een beeld te geven over hoe de inrichting van de centrale bank er uit gaat zien, wordt er een adviesrapport geschreven om antwoord te geven op de vraag: 'Hoe richt ik op een efficiënte manier de centrale bank in?' In dit rapport wordt eerst een beeld gegeven over het beheer. In het analyse gedeelte worden verschillende kwaliteiten uitgelegd, die toegepast zullen worden in het eindproduct. Hierna worden kort de fuctionele en niet-functionele eisen uitgelegd, waarna er een advies gegeven wordt over de inrichting, beargumeteerd met alternatieven.

2 Beheren

2.1 Git

Versiebeheer wordt met Git gedaan. Hier komt zowel de code, ontwerpen (HW en SW) en documentatie op te staan.
https://gitlab.cmi.hro.nl/0945444/arduino/tree/master/project-4

2.2 Risicolog

#	Risico Beschrijving	Kans	Impact	Risico*	Maatregel	?	Status Omschrijving	Datum
R1	De school brandt af	1	5	5	Niet met vuur spelen	N	Net neergezet, nog geen maatregel. Wel brandblussers gezien	25-05-18
R2	Afwezigheid	3	2	6	Straffen en er gewoon zijn	😐	Mensen komen nog wel te laat maar meestal niet afwezig, of met een goede reden	27-05-18
R3	Slechte communicatie	2	3	6	Contact met elkaar houden door vragen (los van het project) te stellen	🙂	Dit gebeurd goed in onze groep!	29-05-18

Kans: schaal 1 (klein) t/m 5 (zeer groot)
Impact: schaal 1 (zeer lage) t/m 5 (zeer hoge)
Risico: = kans * impact
?: [status] 🙂 opgelost, 😐 bezig, ☹ niet opgelost, N nieuw

2.3 Issue Tracker

#	Datum In	Issue	Verantwoordelijk	?	Datum	Beschrijving

?: [status] 🙂 opgelost, 😐 bezig, ☹ niet opgelost, N nieuw

3 Analyseren

3.1 Kwaliteit: Uitbreidbaarheid

3.1.1 Communicatie

- **UDP**:

Het UDP protocol bevind zich op de transport laag van het OSI model. Dit protocol voorziet applicatie's van host-to-host communicatie. Karakteristieken van dit protocol zijn dat het heel simpel en snel is, maar weinig functionaliteit biedt. Het ondersteund bijvoorbeeld geen bevestiging van verstuurde pakketten en daardoor ook geen hertransmissies.

- **TCP**:

Net als UDP bevind het TCP protocol zich op de transport laag van het OSI model. Ook dit protocol voorziet applicatie's van host-to-host communicatie. Karakteristieken van dit protocol zijn dat het veel functionaliteit biedt en betrouwbaar is. Het ondersteund bevestigingen en als een pakket niet ontvangen is wordt deze automatisch opnieuw verstuurd zonder dat de applicatie hier iets vanaf weet, dit scheeld veel programmeerwerk. Zowel UDP als TCP hebben een lage overhead en hoge transmissiesnelheid, maar omdat TCP meer functionaliteit biedt is de overhead iets hoger en de transmissiesnelheid iets lager.

3.1.2 Netwerk uitbreiden

Het is eenvoudig om het netwerk uit de breiden of een nieuwe bank toe te voegen. Dit komt omdat de centrale server gebruikt maakt van IBAN rekening nummers. Elke groepserver die wordt toegevoegd heeft een bepaade prefix die wordt gehanteerd in de IBAN nummers die zij mogen uitgeven. Hierdoor kan de centrale server heel makkelijk transactie's doorsturen naar de bijpassende groep server.

3.2 Kwaliteit: Security

3.2.1 Pincode opslag

- **Plain text**:

De pincode's worden als leesbare ongeformateerde tekst in de database opgeslagen. Tijdens het inloggen wordt de ingevoerde pincode vergeleken met de pincode die is opgeslagen in de database.

- **Hash**:

De unieke identifier (UID) van de pinpas wordt opgeslagen en als salt gebruikt. Tijdens het inloggen wordt de UID + pincode gehashed met het SHA256 hash algoritme. De output wordt daarna vergeleken met de hash die is opgeslagen in de database (zie figuur 3.1).

Figuur 3.1: De data van een pinpas die wordt opgeslagen

id	user_id	password
ED9906850000	1	3082c921e289fc239f0a56138da41caae7f7bb4dc2e8f4712fdc1a4be7a38b67

3.2.2 Versleuteling van communicatie

- **TLS**:

TLS is een cryptografisch protocol die veilige communicatie bied aan een computer netwerk. De verbinding is prive (of veilig) omdat symmetrische cryptografie wordt gebruikt om de data te versleutelen. De sleutels voor deze symmetrische encryptie zijn uniek gegenereerd voor elke connectie en zijn gebaseerd op een gedeelde 'secret' aan het begin van de sessie. De server en client onderhandelen de details over welk encrpytie algoritme en cryptografische sleutels er worden gebruikt, voor dat er ook maar één byte aan data wordt verstuurd. Deze onderhandeling is zowel veilig als betrouwbaar.

De sleutels de gegenereerd worden zijn afhankelijk van een certificaat, zowel de client als de server heeft een certificaat nodig. Dit certificaat zorgt er niet alleen voor dat de beveiligde verbinding tot stand kan worden gezet, maar ook voor het verificeren van de identiteit. De server weet daarom zeker dat het met een bepaalde bank en niet met een hacker de verbinding heeft opgezet.

3.3 Kwaliteit: Vertrouwen

Alle verbindingen zullen ten alle tijden versleuteld d.m.v. TLS worden opgezet. Daar waar gegevens nodig zijn, zullen altijd de minimale gegevens worden uitgewisseld. Dit verlaagd het risico van uitlekken van gegevens.

3.4 Kwaliteit: Tevredenheid

Verbindingen die worden opgezet zullen nooit langer bestaan dan nodig is, zodra de gebruiker zijn sessie afbreekt, zal de verbinding met de centrale server ook worden verbroken. Dit versterkt het vertrouwen en daardoor ook de tevredenheid.

3.5 Functionele eisen

- De geldautomaat moet biljetten van tenminste vier verschillende waarden uit kunnen geven
- De gebruiker kan niet, zonder een pin-opdracht te geven, geld uit de automaat halen
- De geldautomaat geeft altijd het juiste bedrag
- De geldautomaat geeft alleen geld als het saldo toereikend is
- De gebruiker kan zelf selecteren welke biljetten hij/zij wil ontvangen
- De gebruiker kan geen biljetten kiezen die niet aanwezig zijn in de geldautomaat
- De geldautomaat is robuust (kan zelfstandig staan en valt niet om/uit elkaar tijdens gebruik)
- De biljetten in de geldautomaat mogen maximaal de dikte van een speelkaart hebben
- Na het pinnen wordt er een bon geprint met een bon printer. Op deze bon staat in ieder geval hoeveel geld er is opgenomen en bij welke (lokale of individuele) bank dit is gebeurd

3.6 Niet-functionele eisen

- Zowel de centrale server als de groep server moeten gebruik maken van hetzelfde protocol om te communiceren.
- De code wordt geschreven in C++.
- De applicatie van de centrale server zal op het Linux operating system draaien.

3.7 Alternatieven

- UDP wordt niet gekozen omdat dit protocol geen bevestigingen en het hertransmissies onderstuend. Deze functionaliteit zou in de applicatie kunnen worden toegevoegd, maar dit kost extra tijd (en daardoor ook geld) voor de programmeur.
- Plain tekst om de pincode's op te slaan wordt ook niet gekozen omdat hashing een basis security feature is. Dit kost niet veel extra tijd en/of resources voor zowel de programmeur als database.

4 Adviseren

4.1 Communicatie

Er wordt voor TCP gekozen omdat dit protocol bevestigingen en eventuele hertransmissies automatisch verwerkt. De applicatie hoeft hier geen rekening mee te houden omdat dit heel dicht op de hardware afspeeld. Dit scheeld veel tijd tijdens het ontwikkelen en testen van de applicaitie.

4.2 Netwerk uitbreiden

Het makkelijk uitbreiden is mogelijk omdat de centrale server IBAN rekening nummers hanteerd. Dit is de standaard in de EU en zal daarom ook gevolgt worden.

4.3 Pincode opslag

Met pincode's moet veilig worden omgegaan, daarom worden ze niet direct opgeslagen. In plaats daarvan worden ze eerst in een hash functie gestopt. Het wachtwoord is niet af te leiden uit de output van deze functie. Omdat er een salt wordt gebruikt worden rainbow table attacks voorkomen. Deze salt zorgt ervoor dat hashes met dezelfde pincode nooit dezelfde output hebben. Een ander voordeel van hashing is dat de output altijd even lang is, en daardoor zit er geen limiet op de lengte van de pincode.

Bronnen:

- https://en.wikipedia.org/wiki/Cryptographic_hash_function#Password_verification
- https://en.wikipedia.org/wiki/Salt_(cryptography)#Example_usage

4.4 Versleuteling van communicatie

Voor communicatie tussen pinautomaten en servers wordt gebruik gemaakt van een versleutelde verbinding. Het protocol dat deze versleuteling mogelijk maakt heet TLS en staat voor Transport Layer Security.

Als een aanvaller de communicatie onderschept kan de data niet worden ingezien. Dit komt omdat de data versleuteld en dus niet in plain tekst verstuurd wordt.

Bron:

- https://en.wikipedia.org/wiki/Transport_Layer_Security

4.5 Vertrouwen

Privacy van gebruikers blijft gewaarborgd omdat verbindingen ten alle tijden versleuteld worden opgezet. Hierdoor blijft het vertrouwen hoog.

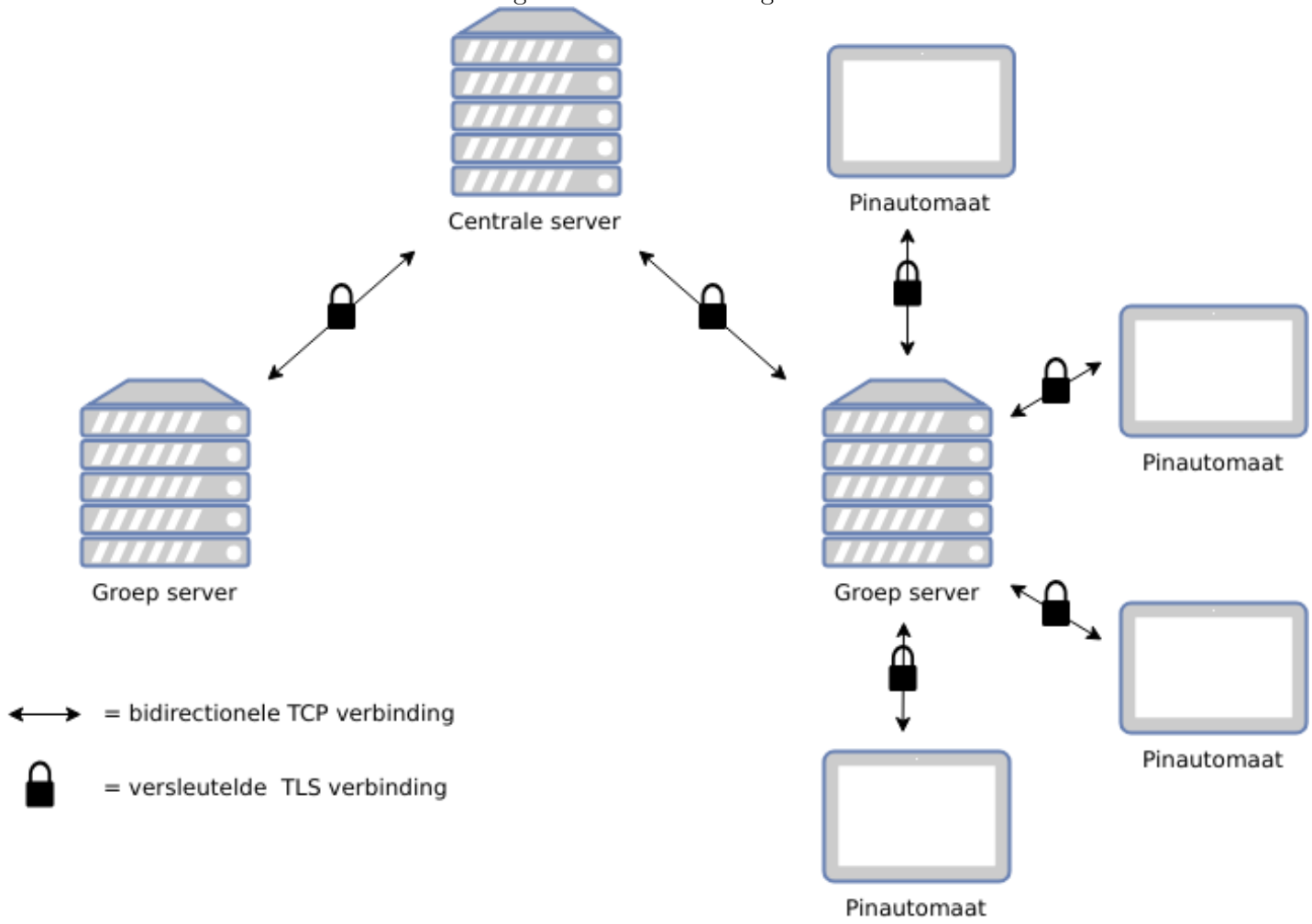
4.6 Tevredenheid

Omdat gegevens alijd versleuteld en correct worden verwerkt zullen er geen onverwachte fouten optreden. Dit zorgt ervoor dat de eindgebruiker tevreden blijft.

5 Ontwerpen

5.1 Netwerkdiagram

Figuur 5.1: Netwerkdiagram



5.2 Dataflow diagram

Figuur 5.2: Dataflow diagram