BEVEiliGINGSVERSLAG

Pirmin Kalbermatter, Chevan Ramcharan en Thomas van Moolenbroek

TI1C

Groep 6

Bank of Venezuela

Contents

[Inleiding](#_Toc167366385)

[Dataflow diagram](#_Toc167366386)

[Risicoanalyse](#_Toc167366387)

[Systemen binnen de pinautomaat](#_Toc167366388)

[Verbindingen tussen de verschillende banken](#_Toc167366389)

[De fysieke hardware](#_Toc167366390)

[Maatregelen](#_Toc167366391)

[Systemen binnen de pinautomaat](#_Toc167366392)

[Verbindingen tussen de verschillende banken](#_Toc167366393)

[De fysieke hardware](#_Toc167366394)

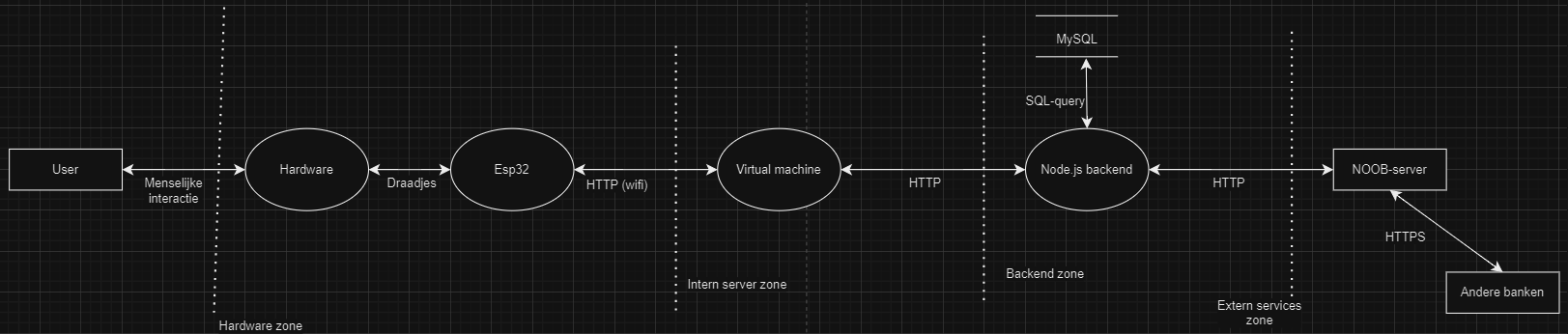
[Conclusie](#_Toc167366395)

## Inleiding

In dit project wordt er een pinautomaat gemaakt. Daar komt veel beveiliging bij kijken, omdat het geld en de gegevens van de klanten veilig moeten zijn. Daarom is het volgende onderzoek uitgevoerd met als onderzoeksvraag:

“Welke beveiligingsmaatregelen moeten genomen worden ter beveiliging van de bank?”

**Dataflow diagram**

In het onderstaande dataflow diagram is te zien welke datastromen er zijn. Aan het begin staat de user input. Die heeft een bepaalde interactie met de pinautomaat hardware, daarom begint daar de hardware zone. De hardware is door middel van draadjes verbonden aan de ESP32, waarna de ESP32 HTTP-code via de wifi doorstuurt naar de virtual machine, de interne server. Die communiceert met de Node.js backend, welke communiceert met de database door middel van SQL-statements, en de noob server ook weer door middel van HTTP. De noob server is een overkoepelende externe server die communicatie onderhoudt met de andere banken.

**Risicoanalyse**

Er zijn verschillende risicogebieden waar aanvallen uitgevoerd kunnen worden:

* Systemen binnen de pinautomaat
* Verbindingen tussen de verschillende banken
* De fysieke hardware

**Systemen binnen de pinautomaat**

Er kunnen aanvallen worden uitgevoerd op meerdere systemen binnen de automaat, namelijk de interface, database, of API.

* Bij de interface is het mogelijk om Cross-Site scripting aanvallen (XSS) uit te voeren, waarbij aanvallers schadelijke scripts injecteren in de webinterface die worden uitgevoerd in de browsers van gebruikers. Ook SQL-injection aanvallen zijn mogelijk, waarbij aanvallers schadelijke SQL-query’s injecteren via bepaalde invoervelden op de webinterface, waardoor ze ongeautoriseerde toegang krijgen tot de database.
* Bij de database is het mogelijk om brute force aanvallen uit te voeren, waarbij aanvallers proberen in de database te komen door brute force of met bijvoorbeeld gelekte inloggegevens.
* Bij de API is het mogelijk om een DoS aanval uit te voeren, waardoor de API overbelast raakt, of om communicatie tussen de client en de API te onderscheppen en te wijzigen.
* De RFID-lezer kan ook gemanipuleerd worden, waardoor bijvoorbeeld een fout pinpas toch door in het systeem kan komen.

**Verbindingen tussen de verschillende banken**

Er kunnen ook aanvallen worden uitgevoerd op de verbinding tussen de verschillende banken. Dit kan op het netwerk of op de gebruikte protocollen.

* Aanvallers kunnen gegevens onderscheppen op het netwerk. Ze kunnen gebruik maken van ‘packet sniffing’, waarbij onversleutelde data die over het netwerk worden verzonden worden verzameld.
* Bij de gebruikte protocollen kunnen aanvallers de systemen forceren om zwakkere protocollen te gebruiken, waardoor het systeem kwetsbaarder wordt. Ook kunnen ze geldige dataoverdrachten onderscheppen en herhalen om daarmee ongeautoriseerde acties uit te voeren.

**De fysieke hardware**

Bij de hardware is het mogelijk om toegang te krijgen tot de fysieke hardware door sabotage. Aanvallers kunnen ook gebruik maken van ‘tampering’, waarbij de hardware wordt gemanipuleerd om kwaadaardige activiteiten mogelijk te maken.

**Maatregelen**

**Systemen binnen de pinautomaat**

* Alle gebruikersinvoer dient zowel aan de server- als de client-side gevalideerd en ontsmet te worden. Dit kan worden bereikt door een whitelist-benadering te hanteren, waarbij alleen expliciet toegestane invoer geaccepteerd wordt. Bibliotheken zoals OWASP’s ESAPI kunnen hiervoor worden gebruikt. Deze maatregel voorkomt dat schadelijke scripts en code worden geïnjecteerd en uitgevoerd, wat de kans op XSS en soortgelijke aanvallen vermindert.
* Een Content Security Policy moet worden ingesteld via HTTP-headers om de bronnen te beperken vanwaar scripts geladen mogen worden. Dit voorkomt de uitvoering van ongeautoriseerde scripts. CSP helpt bij het stoppen van XSS-aanvallen door de uitvoerbare scripts te beperken tot vertrouwde bronnen.
* Gevoelige gegevens moeten worden versleuteld, zowel in rust (bijvoorbeeld door disk encryption) als tijdens transport (door gebruik van TLS/SSL). Encryptie beschermt gegevens tegen ongeautoriseerde toegang en manipulatie, zelfs als aanvallers toegang krijgen tot de gegevens.
* Alle API-verkeer moet via HTTPS verlopen, en certificaat pinning moet worden geïmplementeerd om te voorkomen dat aanvallers de gegevens kunnen onderscheppen en wijzigen.
* Stel limieten in voor het aantal verzoeken per tijdseenheid per gebruiker of IP-adres. Dit kan worden gedaan met behulp van API-gateways of specifieke rate limiting bibliotheken. Rate limiting voorkomt overbelasting van de API en beschermt tegen DoS-aanvallen, waardoor de beschikbaarheid van de diensten behouden blijft.
* Bij de RFID lezer kan beveiligd worden met input-validatie zodat alleen correcte pinpassen in het systeem binnen kunnen komen.

**Verbindingen tussen de verschillende banken**

* Gebruik end-to-end encryptie (zoals TLS) voor alle communicatie tussen systemen om ervoor te zorgen dat data niet kan worden onderschept en gelezen tijdens het transport. Deze maatregel voorkomt dat gegevens onderweg kunnen worden onderschept en gelezen, wat de integriteit en vertrouwelijkheid van de gegevens waarborgt.
* Dwing het gebruik van sterke cryptografische protocollen zoals TLS 1.3 af en schakel oudere kwetsbare protocollen uit. Het gebruik van sterke protocollen vermindert de kans op protocol downgrade-aanvallen en zorgt voor veilige communicatie.
* Voeg tijdstempels en niet-herbruikbare tokens toe aan berichten om te voorkomen dat geldige berichten opnieuw worden verzonden en misbruikt. Anti-replay mechanismen voorkomen dat aanvallers geldige dataoverdrachten herhalen, wat de beveiliging van de communicatie verhoogt.

**De fysieke hardware**

* Gebruik beveiligde behuizingen voor apparatuur en implementeer sensoren die manipulatie detecteren. Beveiligde behuizingen en detectiesystemen beschermen apparatuur tegen sabotage en diefstal en zorgen voor snelle detectie van ongeoorloofde toegangspogingen.
* Implementeer een zero-trust model en strikte toegangscontrole voor medewerkers waarbij elke toegangspoging wordt gelogd en gemonitord. Deze maatregel beperkt de mogelijkheden van medewerkers om ongeautoriseerde handelingen uit te voeren en zorgt voor gedetailleerde logging van toegangspogingen.

**Conclusie**

Er zijn veel risico’s in het hele systeem. Bij deze pinautomaat zijn de volgende maatregelen genomen:

* RFID is beveiligd met input-validatie zodat er geen foute pinpas in het systeem kan komen.
* De GUI is beveiligd met HTTPS waardoor alle gegevens ge-encrypt worden.

**Bronvermelding**

1. OWASP. (n.d.). OWASP ESAPI Project
2. Mozilla. (n.d.). [Content Security Policy (CSP)](https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP)
3. SSL.com. (2020). What is SSL/TLS?
4. OWASP. (n.d.). Certificate and Public Key Pinning
5. OWASP. (n.d.). Rate Limiting
6. OWASP. (n.d.). Transport Layer Protection Cheat Sheet
7. NIST. (2020). [Guide to Physical Security](https://csrc.nist.gov/publications/detail/sp/800-116a/final)
8. Microsoft. (2021). [Zero Trust Deployment Guide](https://docs.microsoft.com/en-us/security/zero-trust/)