# Beveiligingsrapport

Groepsopdracht

Hiu Tung Tai, Mart-Jan Koedam en Max van der Heijden

**Inhoud:**



**Inleiding**

Als bank wil je de gegevens van een klant koste wat het kost beschermen. Ook wil je dat niemand met de gegevens en geldbedragen gaat schommelen, zo kan een aanvaller bijvoorbeeld oneindig veel geld op zijn eigen rekening zetten zodra hij in de database komt. Dat willen wij natuurlijk voorkomen. We gaan onderzoek doen naar de manieren die gebruikt worden om een database aan te vallen en we onderzoeken op welke manier we de database het beste kunnen beveiligen.

**Hoofdvraag:** Hoe kan de database het best beveiligd worden tegen aanvallen?
**Deelvraag 1:** Op welke manier zouden aanvallers onze database kunnen aanvallen?
**Deelvraag 2:** Op welke manieren kan men de database beveiligen tegen zulke aanvallen?

**Deelvraag 1:** Op welke manier zouden aanvallers onze database kunnen aanvallen?

Een database kan op verschillende manieren worden aangevallen. De eerste manier om gegevens uit de database te halen is door SQL injection. SQL-injectie is een kwetsbaarheid voor internetbeveiliging waarmee een aanvaller gegevens uit de database kan krijgen door commandos in te voeren. Een aanvaller kan dan gegevens bekijken die ze normaal niet kunnen achterhalen. Een voorbeeld van SQL injectie: `SELECT * FROM Users WHERE Username='$username' AND Password='$password'.`

Een logische manier om een database te hacken is om simpelweg het goede wachtwoord te achterhalen van de server, op die manier kan de aanvaller makkelijk bij alle gegevens komen die in de database staan.
Een andere manier om in de database te komen is via het Java programma zelf, als de aanvaller bij de source code terecht komt kan hij met die code de database bekijken en bewerken.

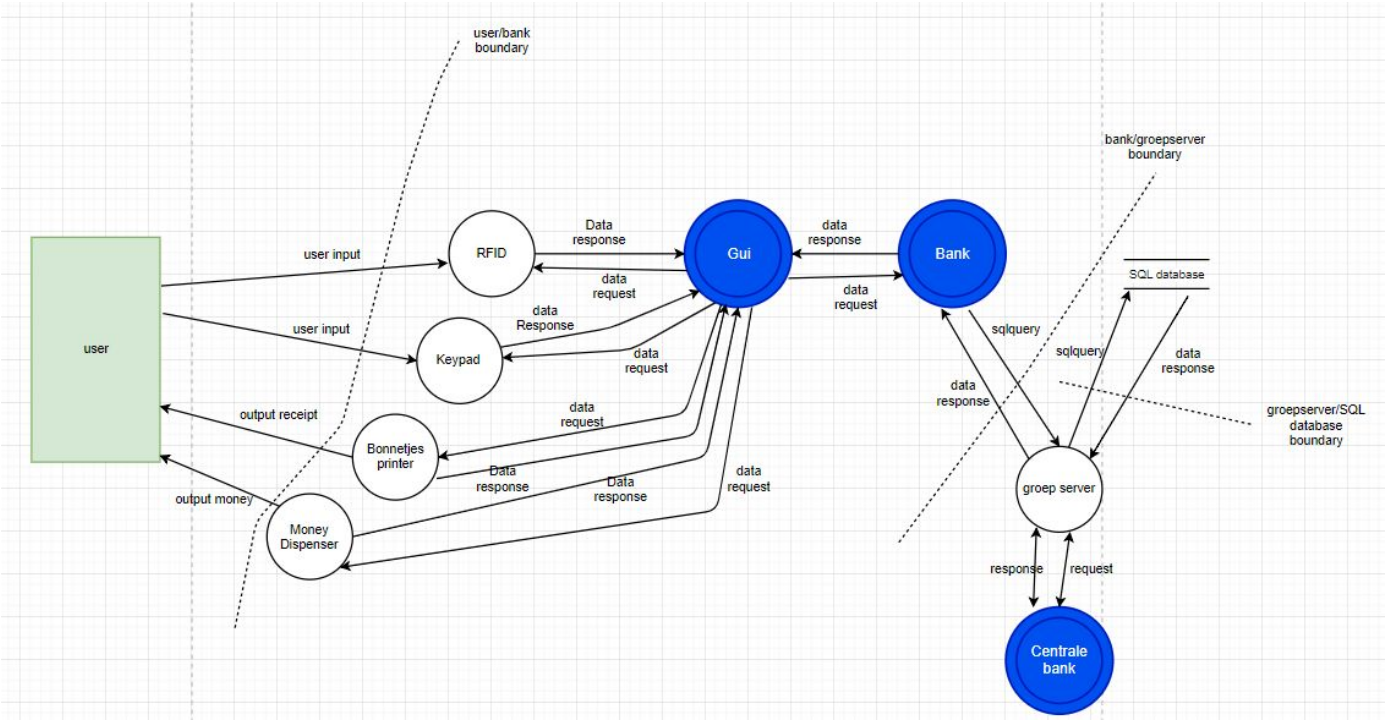
**Deelvraag 2:** Op welke manieren kan men de database beveiligen tegen zulke aanvallen?

Een manier om SQL injectie tegen te gaan is door prepared statements. Prepared statements zorgen ervoor dat SQL commando's niet direct in de database komen. De SQL commando's worden herschreven zodat de database niet verstoord wordt. Ook kan men SQL injectie verminderen door de systeem up to date te houden of door accounts te hebben met gelimiteerde toegang.
Om te voorkomen dat het wachtwoord van de database wordt gehackt kun je natuurlijk een sterk wachtwoord instellen, van bijvoorbeeld meer dan 20 karakters en een variatie van hoofdletters en cijfers.
Ook is het handig om meerdere gebruikers in te stellen voor de database, elke gebruiker heeft zijn eigen privileges en er zijn dus meer verschillende wachtwoorden beschikbaar, als de hacker bijvoorbeeld het account van een gebruiker kraakt met een lage functie en dus weinig privileges, kan de hacker de gegevens niet wijzigen. Een andere manier om dit tegen te gaan is om alle gegevens te encrypten, als de aanvaller de database weet te kraken en bij de database komt kan hij de gegevens niet uitlezen. Om de code te beschermen zodat aanvallers niet via de code in de database kunnen komen is door obfuscator. Met obfuscator kan je de source code veranderen zodat het niet meer leesbaar is. En als het niet leesbaar is, dan kunnen aanvallers hier niks mee.

**Conclusie:**
De database kan het best beveiligd worden tegen aanvallen door prepared statements te gebruiken tegen de SQL-injections.
Ook de database up-to-date houden zorgt ervoor dat hackers niet de kans krijgen om oude hack-manieren te gebruiken op een outdated database.
Een sterk wachtwoord, verschillende gebruikers en privileges zorgen ervoor dat het wachtwoord niet snel wordt gekraakt, en als het wordt gekraakt zijn de gegevens encrypted zodat de aanvaller er nog steeds niks mee kan. Tot slot om de code te beveiligen kan er een obfuscator op de code worden gezet, op die manier is de code niet leesbaar.

**Bronnen:**
https://www.wikihow.com/Hack-a-Database
https://portswigger.net/web-security/sql-injection
https://kb.bodhost.com/secure-database-server/
http://go-database-sql.org/prepared.html
https://www.preemptive.com/obfuscation
https://security.berkeley.edu/resources/best-practices-how-articles/system-application-security/how-protect-against-sql-injection

## Data Flow Diagram



# *Beveiligingsrapport Max*

**Hoofdvraag:** Hoe moet de hardware van de automaat beschermd worden?

**Deelvragen:** **1.** Uit welke onderdelen bestaat het systeem?
**2.** Wat zijn de zwakke punten in het systeem?

**1. Uit welke onderdelen bestaat het fysieke systeem?**
De hardware van de automaat bestaat uit 2 delen.
Het 1e deel bestaat uit een Arduino uno, RFID & keypad en is het deel waar de interactie tussen de gebruiker en automaat plaatsvindt.

Het 2e deel bestaat uit de bonnetjes printer en geld dispenser, ook weer aangestuurd door een aparte arduino uno.

**2. Wat zijn de zwakke punten[1] in het systeem?**
De hardware bestaat uit basic arduino onderdelen welke aangesloten zijn met jumper kabels, die kabels zijn niet erg sterk en kunnen dus gemakkelijk worden losgemaakt.
Ook is het erg makkelijk om kortsluiting te veroorzaken met deze kabels wat voor problemen kan zorgen voor de rest van het systeem.

Een ander zwak punt zijn de kabels waarmee de arduino's verbonden zijn aan de laptop/pc, dit zijn normale usb-c kabels en kunnen makkelijk losgemaakt worden. Wat dit met zich meebrengt is dat dan de fysieke systemen meegenomen kunnen worden.

De keypad is ook een mogelijkheid, zo kan men problemen veroorzaken door grote random getallen in te vullen welke het systeem niet aan kan.

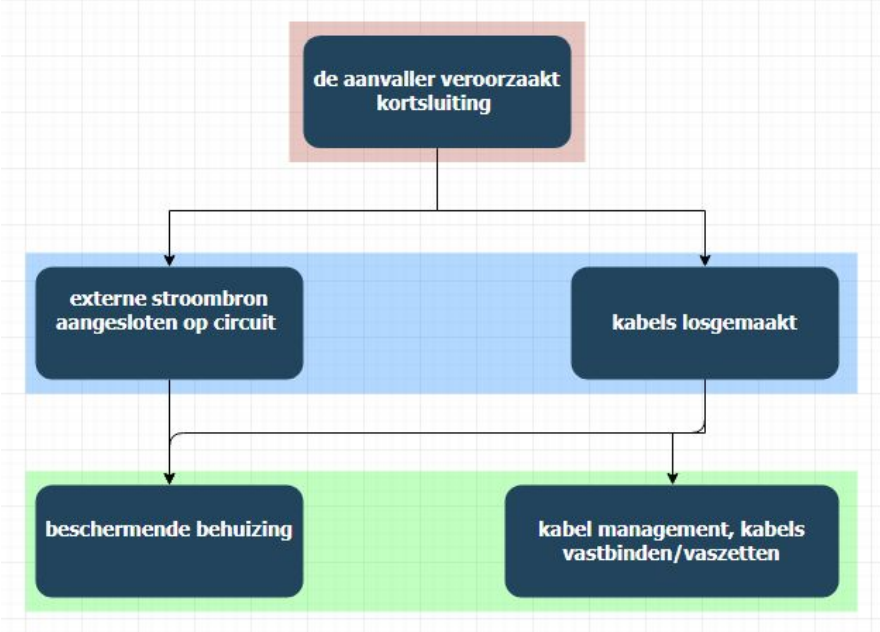
Als de systemen een behuizing hebben kan het ook voorkomen dat men via een spleet of gat het systeem blokkeert of toch nog een aantal kabels weet los te maken.

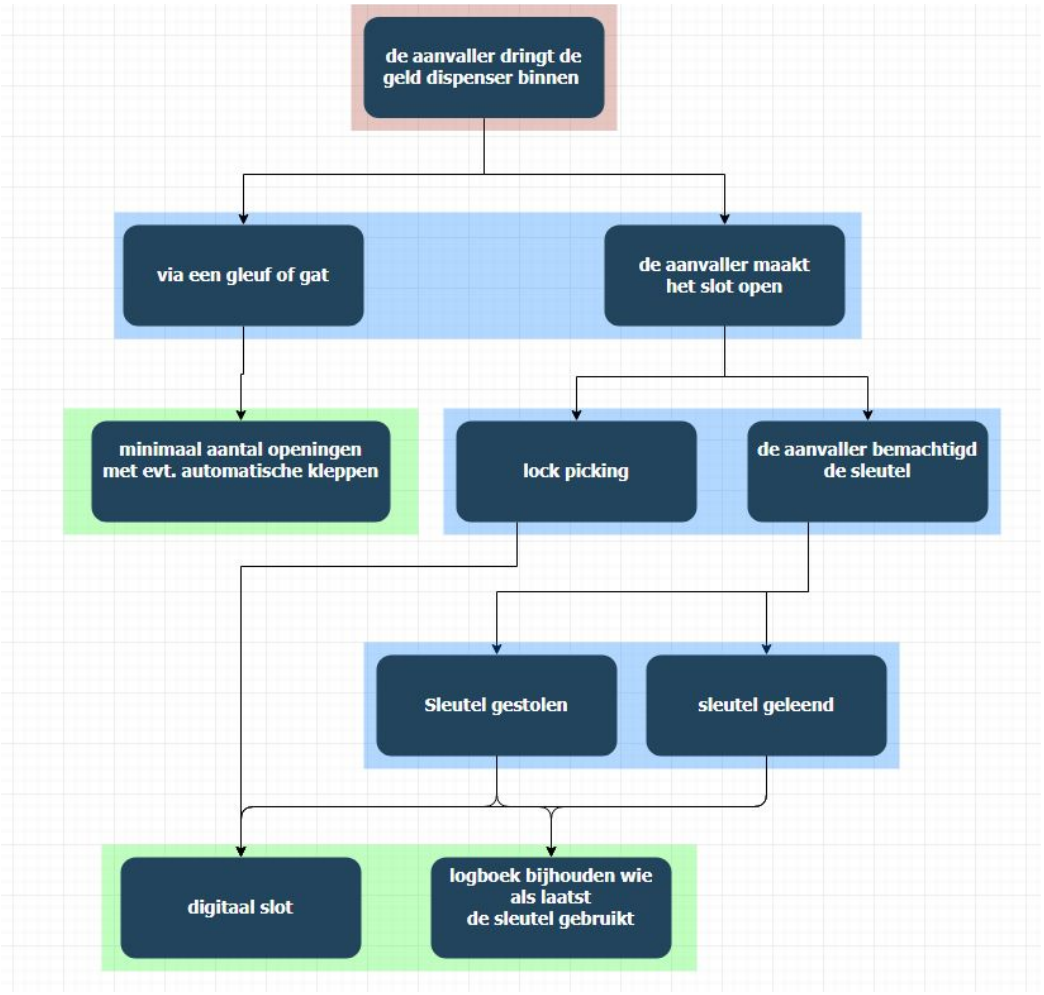
[1] Note: bij dit project wordt er vanuit gegaan dat de automaat niet fysiek gesloopt mag worden.

**Waartegen moet de hardware van de automaat beschermd worden?**
Naar aanleiding van de deelvragen kan er de conclusie getrokken worden dat het grootste gevaar is dat men de kabels losmaakt en zo storingen veroorzaakt.
Dit kan opgelost worden door een behuizing te maken met zo min mogelijk gaten/spleten, en eventueel interne kleppen/schotten welke open en dicht kunnen zodat men niet iets naar binnen kan steken.

Bronnen:
https://www.schneier.com/academic/archives/1999/12/attack_trees.html

Attack trees:




de aanvaller dringt de geld dispenser binnen
via een gleuf of gat
de aanvaller maakt het slot open
minimaal aantal openingen met evt. automatische kleppen
lock picking
de aanvaller bemachtigd de sleutel
Sleutel gestolen
sleutel geleend
digitaal slot
logboek bijhouden wie als laatst de sleutel gebruikt

# Beveiligingsrapport Mart-Jan

Het is belangrijk dat de software van het interface van de bank goed beveiligd is. Je wilt niet dat iemand de software zo kan manipuleren dat die persoon bij accountgegevens kan, of dat die persoon het interface breekt.
Hiervoor heb ik een hoofdvraag en een paar deelvragen opgesteld met betrekking tot mijn interface van de bank:

- **Hoofdvraag:** Hoe kan ik de software van de bank het beste beveiligen?
- *Deelvraag 1:* Op welke manieren zou de software aangevallen kunnen worden?
- *Deelvraag 2:* Op welke manieren zouden we die aanvallen tegen kunnen gaan?

***Deelvraag 1:*** **Op welke manieren zou de software aangevallen kunnen worden?**

*Op welke manieren zouden de aanvallers in kunnen loggen op de accounts van gebruikers?*
Het kan voorkomen dat een gebruiker halverwege de sessie stopt met het proces en vergeet de betaling af te breken, dan blijft de gebruiker ingelogd en zou de aanvaller gemakkelijk geld kunnen pinnen op het account van de gebruiker.
Wanneer de aanvaller een pas van een gebruiker in bezit heeft kan hij mogelijk:

- Meekijken bij de gebruiker en de pincode zien te achterhalen.
- De gebruiker dwingen de pincode af te geven.
- Op goed geluk de juiste pincode intoetsen.
- Via de database lezen welke pincode bij de gebruiker hoort.
- Het pinproces omzeilen en met enkel de pas geld opnemen.

De aanvaller zou de database kunnen hacken en de gegevens daar uitlezen.
De aanvaller weet het pingedeelte te omzeilen en gebruikt alleen de pas om in te loggen

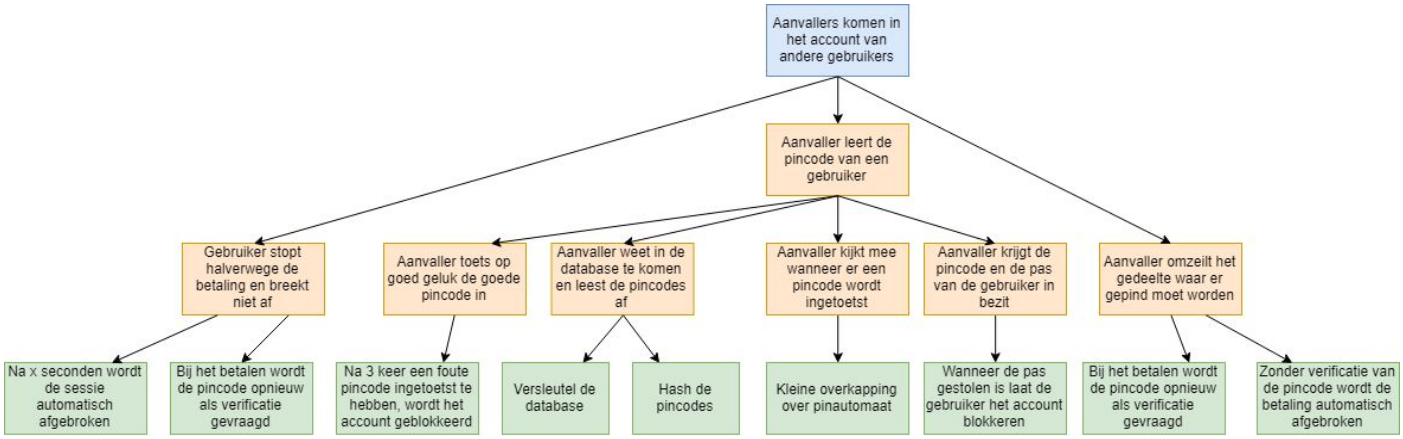
*Op welke manieren zou het programma kunnen breken door de aanvallers?*
De aanvaller gaat proberen om het programma te laten crashen, of om een functie te slopen.
De aanvaller kan dat doen door bijvoorbeeld telkens te switchen tussen verschillende schermen, zo roept het programma telkens een nieuw scherm aan en zal het uiteindelijk vastlopen door de hoeveelheid schermen die draaiend zijn.
De aanvaller kan ook zijn eigen pincode 2 keer verkeerd invoeren, en wanneer er een gebruiker gaat pinnen en per ongeluk de verkeerde pincode invoert (wat niet ongewoon is), dan wordt zijn/haar pas geblokkeerd.
Als de aanvaller de knoppen erg snel na elkaar indrukt worden er meerdere schermen tegelijk geopend en gesloten, hierdoor kan het programma fataal vastlopen.

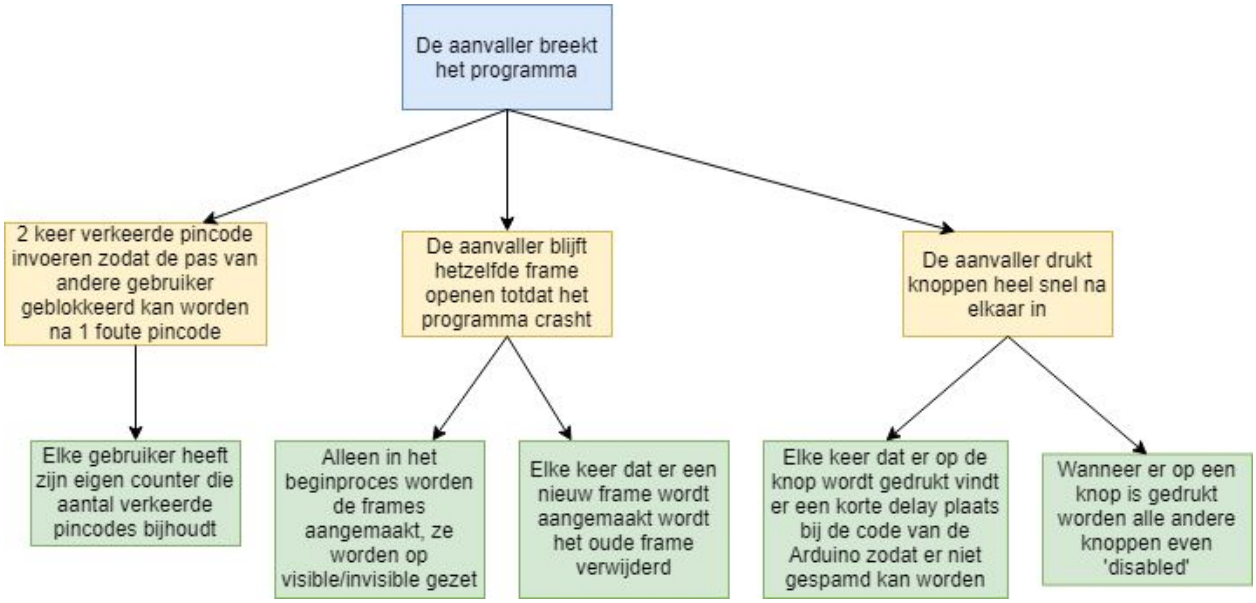
***Deelvraag 2: Op welke manieren zouden we die aanvallen tegen kunnen gaan?***

*Op welke manieren zouden de aanvallers in kunnen loggen op de accounts van gebruikers?*
Om te voorkomen dat de aanvaller gemakkelijk een account van de gebruiker kan gebruiken, kunnen we een stuk code implementeren die na ongeveer 15 seconden zonder activiteit de betaling automatisch afbreekt.
We kunnen niet heel veel doen wanneer de aanvaller de gebruiker dwingt om geld voor ze te pinnen, we zouden een herhalende tekst op het beeldscherm kunnen zetten die de gebruikers waarschuwt voor dieven en een tip om nooit de pincode door te vertellen.
Wat erg handig zou zijn is als de gebruiker zijn pas van buitenaf kan laten blokkeren door bijvoorbeeld te bellen naar de bank, dan kan de aanvaller in de toekomst niet meer met de pas pinnen.
Om te voorkomen dat de aanvaller de database van buitenaf afleest kunnen we de database op verschillende manieren beveiligen. (Zie groepsdeel)
Om te voorkomen dat de aanvaller het pinproces weet te omzeilen kunnen we een extra verificatieprotocol implementeren waar de pincode nogmaals wordt gevalideerd, als er geen pincode is wordt het proces direct afgebroken.

*Op welke manieren zou het programma kunnen breken door de aanvallers?*
Om te voorkomen dat er telkens een nieuw scherm wordt aangemaakt kan je in de code implementeren dat het scherm maar 1 keer wordt aangemaakt aan het begin van de code, en in de huidige sessie wordt het scherm alleen maar zichtbaar of onzichtbaar gemaakt.
Als iemand zijn pincode een keer verkeerd heeft ingevoerd moet dat geen effect hebben op de andere gebruikers, bij elke gebruiker moet apart worden bijgehouden hoe vaak ze de verkeerde pincode hebben ingevoerd. Op die manier kan de aanvaller geen accounts van andere gebruiker blokkeren.
Wanneer er een knop wordt ingedrukt, moeten de knoppen op het scherm direct ontoegankelijk worden, dan kunnen er niet meerdere knoppen tegelijk ingedrukt worden.

***Hoofdvraag: Hoe kan ik de software van de bank het beste beveiligen?***
***Conclusie:*** Na x seconden zonder activiteit wordt het pinproces automatisch afgebroken, op het scherm kan een waarschuwing voor de gebruikers worden getoond om ze alert te maken van de gevaren en dat ze nooit hun pincode mogen vertellen. De pincode wordt meerdere keren gecheckt tijdens de sessie zodat het pingedeelte niet omzeilt kan worden.
Om te voorkomen dat de aanvaller het programma breekt is er een goede code nodig, er moeten geen dingen onnodig aangemaakt worden als ze er al zijn en elke gebruiker heeft zijn eigen counter voor de hoeveelheid keren dat ze de verkeerde pincode hebben ingevoerd.




Bronnen:

https://www.owasp.org/index.php/CRV2_AppThreatModeling
https://www.schneier.com/academic/archives/1999/12/attack_trees.html
https://docs.google.com/presentation/d/1VcouSbjSr_lE4thyZFqY1Bk4h0gMzLD3KxfVwXBF224/edit#slide=id.g52a4644b7c_0_104

# Beveiligingsrapport Hiu Tung

**Hoofdvraag: Hoe kan men ervoor zorgen dat de aanvaller geen toegang krijgt tot iemand anders bank account via de pinautomaat?**

Men wilt graag voorkomen dat een aanvaller toegang krijgt tot iemand anders bankaccount. Om te weten hoe je iets moet voorkomen moet je ook weten wat de manieren zijn om toegang te krijgen op een bankaccount. En natuurlijk wil je oplossingen hebben om dat te voorkomen. Daarom zijn deze twee deelvragen gekozen.

**Deelvragen:**

**1.Wat zijn de manieren om toegang te krijgen op iemand anders account via de pinautomaat?**

**2.Wat zijn de oplossingen om een account te beveiligen?**

1.Er zijn velen manieren om aan iemand anders bankaccount te komen. Een makkelijke manier om binnen de account te komen is bijvoorbeeld iemand is vergeten uit te loggen. Een ander manier is door shouldering. Shouldering betekent over de schouder mee kijken om de pincode te zien.[1,2]

Ook kunnen criminelen aan de inloggegevens komen door te skimmen. Skimmen betekent op onrechtmatige wijze de op bankpassen bemachtigen. Ze doen dat bijvoorbeeld door een apparaat voor de pasgleuf te plaatsen. En als iemand een pinpas in de pasgleuf is steekt, dan kunnen de criminelen makkelijk een kopie van de magneetstrip van de pinpas maken.[3,4] Een ander manier om in te loggen is door de pas te stelen en dan willekeurig pincodes invoeren tot dat de crimineel de goede pincode heeft .

2. Er zijn vele manieren om aan iemand anders bankaccount te komen, maar er zijn ook vele oplossingen om dat te voorkomen. Men kan bijvoorbeeld ervoor zorgen dat de account automatisch uitlogt na een paar minuten. Er zijn al vele banken die dit doen. De asn bank en de SNS bank zijn hier een voorbeeld van.[5,6] Ze gebruiken automatisch uitloggen bij online bankieren, maar dat kan men ook doen bij pinautomaten. Hiermee voorkomt men dat iemand in iemand anders bankaccount komt als de gebruiker vergeten is uit te loggen. Voor shouldering is niet echt een manier dat de banken kunnen gebruiken om het te voorkomen, maar shouldering is alleen het afkijken van de pincode. De criminelen hebben alleen de pincode. Ze kunnen hier niks mee, want om in te loggen heb je ook een pasje nodig. Daarom heb je bij vele banken een pasje en de pincode nodig om in te loggen. Dus criminelen die alleen de pas of de pincode hebben, kunnen niet inloggen. Dit zorgt voor extra veiligheid. Een oplossing om skimmen te verminderen is door een opzetstukje. Het voorkomt helaas geen skimmen, maar het maakt het wel lastiger om te skimmen.[3,4] En als laatste kan men de pas blokkeren als iemand drie keer een incorrecte pin invoert.

**Hoofdvraag: Hoe kan men ervoor zorgen dat de aanvaller geen toegang krijgt tot iemand anders bank account via de pinautomaat?**

Kortom aanvallers kunnen in iemand anders bank account komen doordat iemand vergeten is uit te loggen. Ook kunnen ze gebruik maken van shouldering en skimmen of door de pinpas te stelen en dan willekeurige pincodes in te vullen totdat je ingelogt bent. Deze aanvallen kan men voorkomen door automatisch uitloggen na een paar minuten, opzetstukje om skimmen moeilijker te maken, pas en pincode nodig om in te loggen. En na 3 incorrecte pincodes de pas blokkeren. Deze oplossingen kunnen de aanvallers tegenhouden zodat ze geen toegang krijgen tot iemand anders bank account via de pinautomaat.

[1] https://www.encyclo.nl/begrip/Shouldering
[2] https://www.veiligbankieren.nl/overige-fraude/shouldering/
[3] https://www.banken.nl/bibliotheek/skimming
[4] https://www.consumentenbond.nl/betaalrekening/voorkom_pinpasfraude
[5] https://www.asnbank.nl/online-en-mobiel-bankieren/veilig-bankieren/zo-houden-we-bankieren-veilig.html
[6] https://www.snsbank.nl/particulier/over-sns/veilig-bankieren/zo-houden-we-online-bankieren-veilig.html

Attack tree