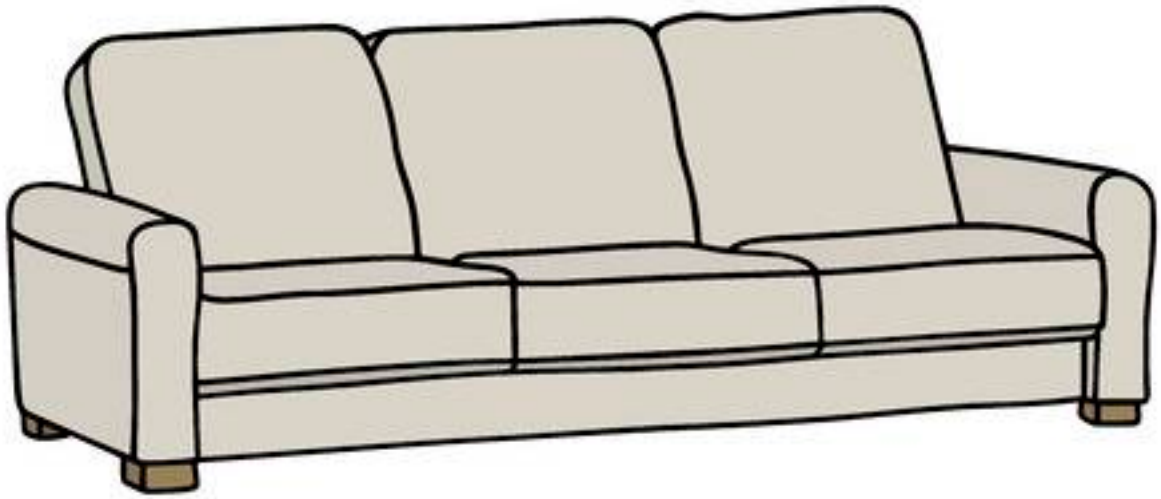
De Bank

beveiligingsplan



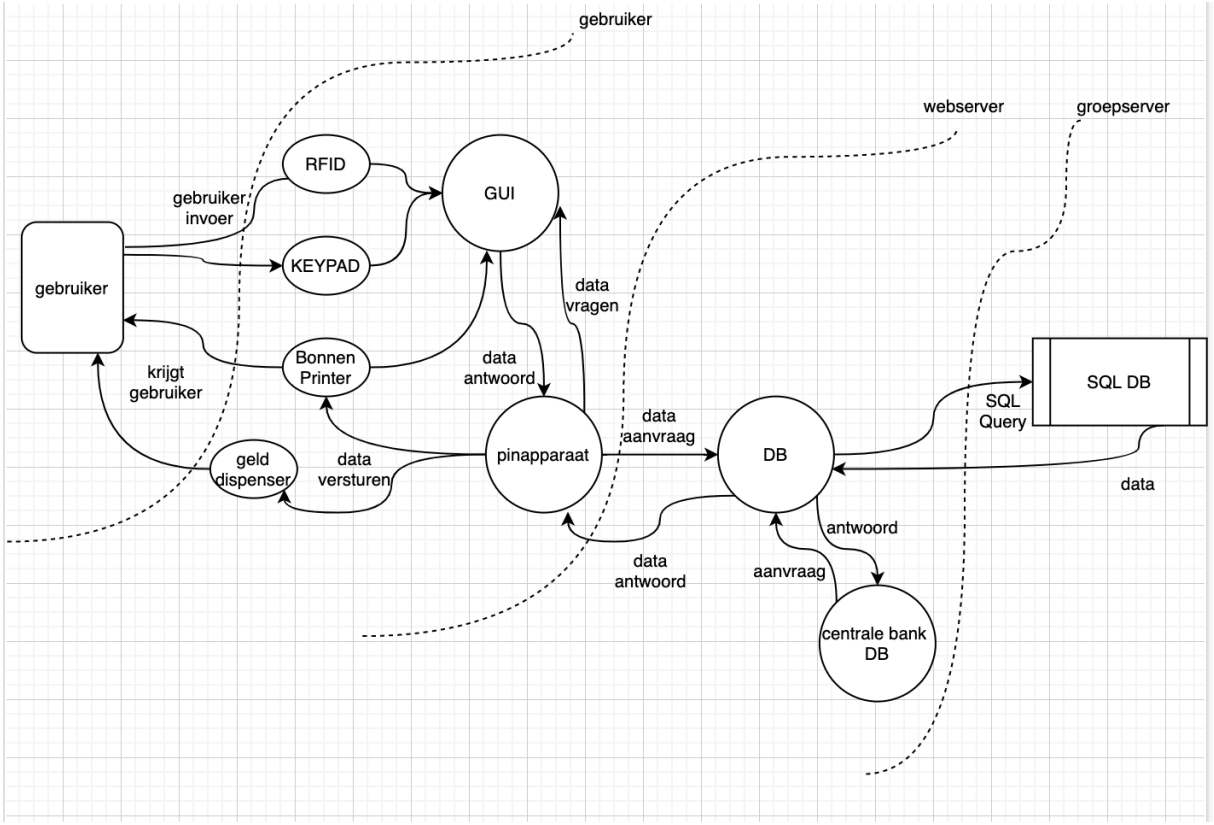
Chenyen Wang | 0988936
Emad Bouzalmad | 0986231
Storm Hoogstrate | 0984053
Tala Hammed | 0995285
Pim van den Toorn | 0984004

Inleiding
Een beveiligingsplan is een rapport waar de plannen voor het beveiligen van een computer of informatiesysteem in staan. Het geeft een systematisch overzicht van het systeem en de technieken voor het verdedigen van een computer voor onbevoegde gebruikers, virussen en ook andere ongelukken dat het systeem in gevaar kan brengen.

Hoofdvraag: Op welke manier wordt de bank het best beveiligd?

Deelvragen: - Op welke manieren kan de pinautomaat aangevallen worden?
- op welke manier worden de gebruikers account het best beveiligd tegen aanvallen?
- hoe kan de database het best beveiligd worden tegen aanvallen?

Data Flow Diagram



Rogue keyboard access:
Dit is een van de simpelere aanvalssoorten. Het is namelijk alleen een fysiek toetsenbord koppelen aan de pinautomaat en vervolgens (door middel van commando's) een stukje code executeren of een SQL-injectie waardoor de pinautomaat gecontroleerd kan worden door de aanvallers.

Het meest lastige van deze aanval is bij de USB of ps/2 komen. Niet alleen omdat deze vaak verborgen zijn, maar de niet gebruikte (of standaard) poorten zijn negen van de tien keer geblokkeerd. Deze aanval is daarom ook een van de redenen waarom die poorten geblokkeerd zijn.

Als de poorten niet geblokkeerd zijn, dan moeten ze wel bereikbaar zijn. De poorten zijn namelijk altijd opgeborgen achter een slot of paneel.

De meest voorkomende manieren om deze open te maken zijn (Als het op die manier beveiligd is):

- Jiggle keys (sleutels die in elk slot passen en door te 'jiggelen' het slot open maakt).
- Een gat boren in het paneel.
- Schroeven uit het paneel halen.

Omdat dit een van de simpelste aanvalssoort is, is het ook gelijk een van de simpelste om te detecteren en op te lossen. Naast de eerdergenoemde methode blokkeren is een andere methode alleen karakters accepteren van de toetsen die er worden gebruikt (Bijv. 0-9, * en #). Verder kan ook het programma stopgezet worden als er een nieuw apparaat gedetecteerd wordt.

Man In The Middle attack (MITM):
Een man in the middle attack is een aanval waarbij informatie tussen twee communicerende partijen, in dit geval de pinautomaat en de klant, onderschept wordt zonder dat de partijen daarvan af weten. Een goed voorbeeld van zo'n MITM-attack is dat Bijv. de postbode jouw brieven leest voordat hij ze bezorgt. Zo kan hij/zij zeer gevoelige informatie bemachtigen en soms zelfs aanpassen.

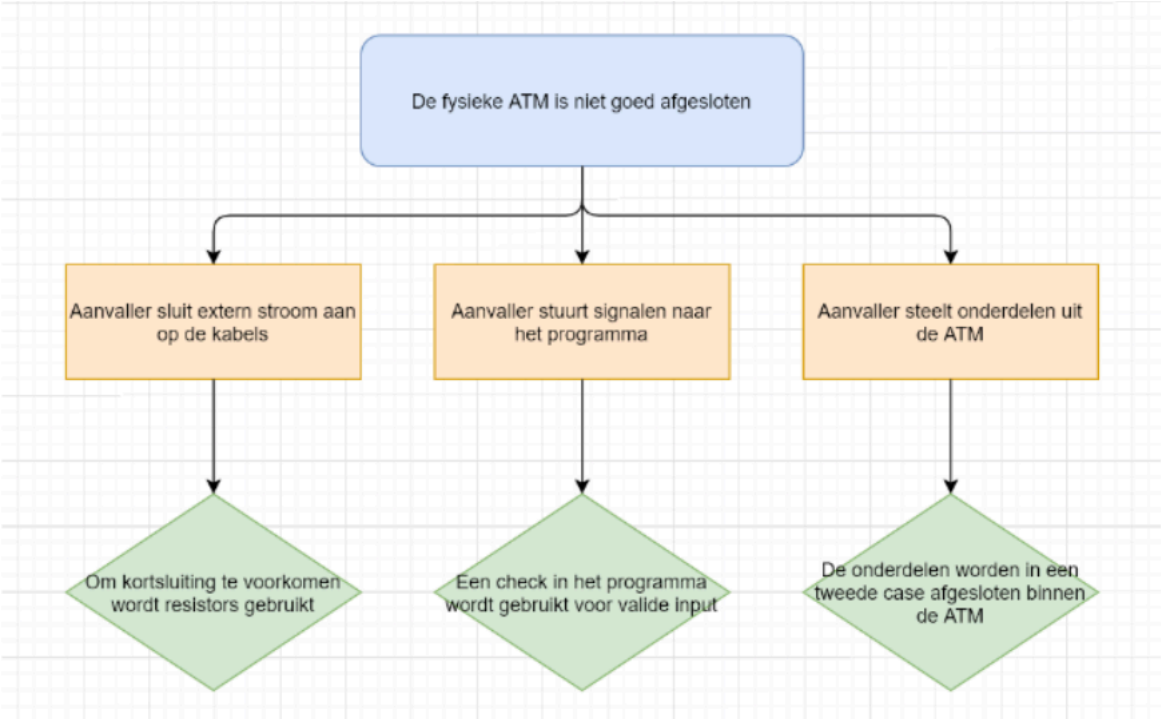
Bij een pinautomaat zijn de twee meest voorkomende MITM-attacks:

- Skimming (RFID)
- Keyloggers (Keypad)

Skimming:
Skimming is een methode om de gegevens van een pinpas te dupliceren. Dit kan volkomen onopvallend gebeuren. Een pinautomaat wordt zo aangepast dat het lijkt alsof het een hedendaagse pinautomaat is, maar eigenlijk is hij geboobytrapt.
Er zijn ook manieren om skimmers helemaal te elimineren, een elektromagnetische puls is er een van. Dit is een korte schok van elektromagnetische energie. Deze schok verstoort (of beschadig vanaf een specifiek niveau) de skimmers, waardoor ze niet meer werken en dus geen data meer kunnen lezen.

Keylogging:
Keylogging is een andere methode om de gegeven van een pinpas te bemachtigen, in dit geval de gebruiker zijn pincode. Er zijn twee soorten keyloggers, de software en- de hardware matige. De softwarematige wordt meer gebruikt dan de hardware matige. De reden hiervoor is dat er bij de hardware matige een keypad replica moet worden toe gevoegd, terwijl er bij de softwarematige alleen een stukje software toe gevoegd moet worden.

Om zo'n keylogger te detecteren is wel een stuk lastiger. De reden hiervoor is dat een keylogger niet altijd een hardware matige keypad is, maar het kan de laatste tijd ook een USB-stick zijn (of flash drive).
Dus de originele manier van, trek aan het toetsenbord is niet altijd valide. Wel een oplossing is net als bij de rogue keyboard access de niet gebruikte porten te blokkeren. Op die manier kan er ook geen USB-stick gebruikt worden als keylogger.



Op welke manieren zouden de aanvallers in kunnen loggen op de accounts van gebruikers en hoe kan je dit voorkomen?

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:

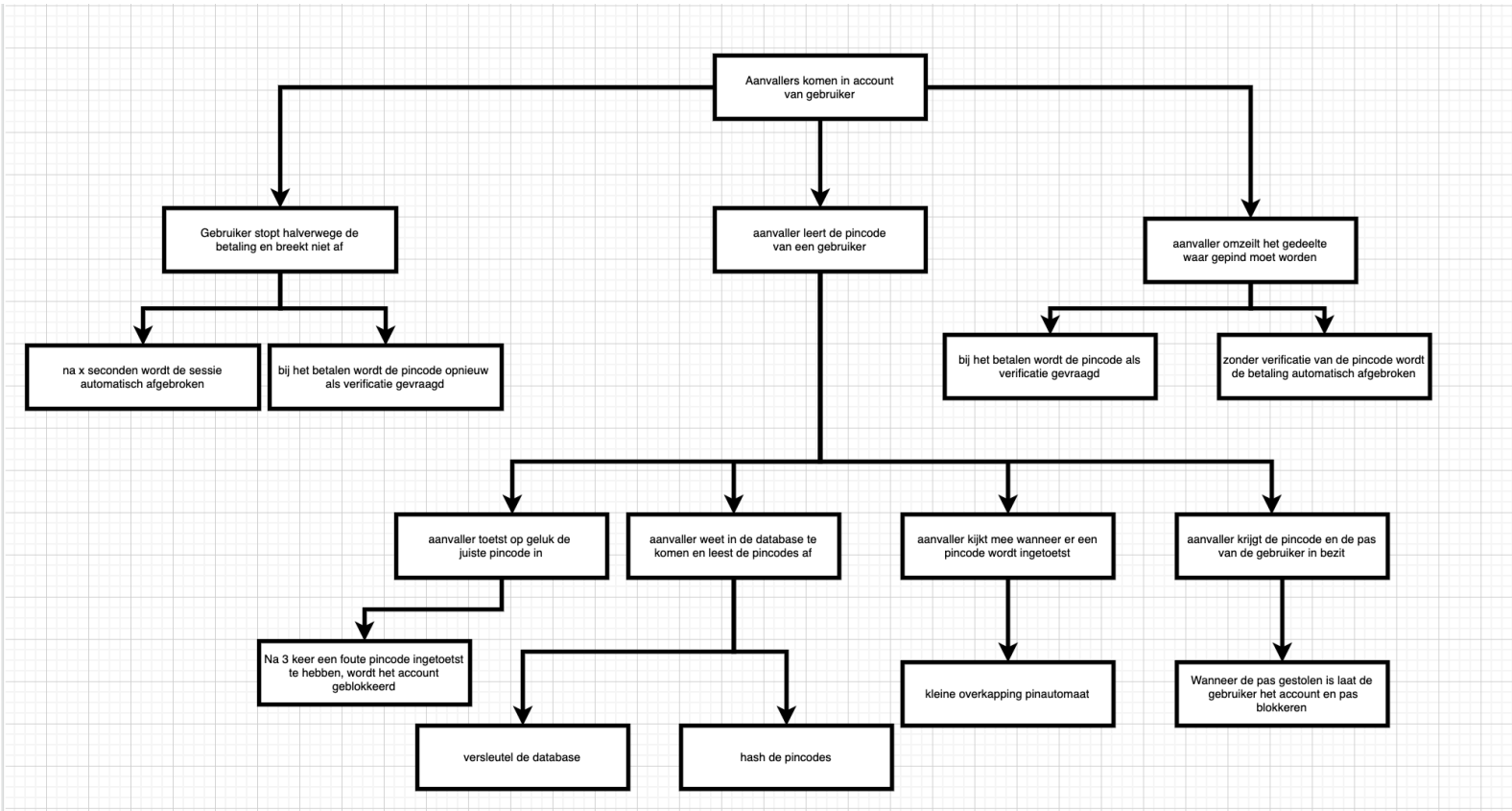
- Meekijken bij de gebruiker en de pincode zien te achterhalen.
- De gebruiker dwingen de pincode af te geven.
- Op goed geluk de juiste pincode intoetsen.
- Via de database lezen welke pincode bij de gebruiker hoort.
- Het pinproces omzeilen en met enkel de pas geld opnemen.

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

Om te voorkomen dat de aanvaller gemakkelijk een account van de gebruiker kan gebruiken, kan er een functie worden geïmplementeerd die na ongeveer 15 seconden zonder activiteit de betaling automatisch afbreekt. 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. 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 crashen door de aanvallers en hoe kan je dit voorkomen?
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.

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.

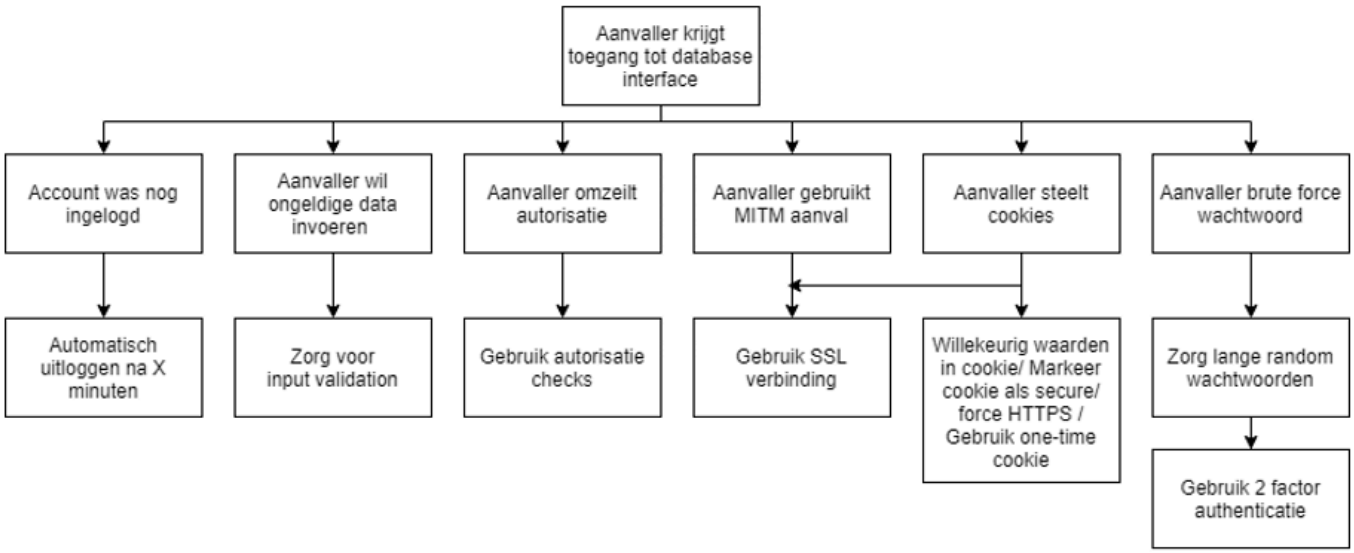

Aanvallers komen in account van gebruiker
Gebruiker stopt halverwege de betaling en breekt niet af
aanvaller leert de pincode van een gebruiker
aanvaller omzeilt het gedeelte waar gepind moet worden
na x seconden wordt de sessie automatisch afgebroken
bij het betalen wordt de pincode opnieuw als verificatie gevraagd
bij het betalen wordt de pincode als verificatie gevraagd
zonder verificatie van de pincode wordt de betaling automatisch afgebroken
aanvaller toetst op geluk de juiste pincode in
aanvaller weet in de database te komen en leest de pincodes af
aanvaller kijkt mee wanneer er een pincode wordt ingetoetst
aanvaller krijgt de pincode en de pas van de gebruiker in bezit
Na 3 keer een foute pincode ingetoetst te hebben, wordt het account geblokkeerd
versleutel de database
hash de pincodes
kleine overkapping pinautomaat
Wanneer de pas gestolen is laat de gebruiker het account en pas blokkeren

Op welke manier zouden aanvallers de 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.

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.

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.



Conclusie

Al met al, is na dit onderzoek gebleken dat het beveiligen van de bank niet zo gemakkelijk is als er gedacht werd. Er valt over veel na te denken, denk aan: Access control fouten, DDoS, SQL-injecties en MITM attack.
Doordat er proactief gekeken wordt naar mogelijke aanvalsplekken kunnen deze worden beveiligd. Door alles te benaderen vanuit het oogpunt van de aanvaller worden de zwakheden snel en effectief aangepakt. Ook nemen wij niet alleen de softwarematige beveiliging in acht, maar ook de hardware matige. Het is namelijk belangrijk dat het hele systeem zo robuust mogelijk is, want een systeem is maar zo sterk als zijn zwakste schakel. Door deze zwakke schakels weg te nemen wordt er een banksysteem door ons gecreëerd dat uiterst veilig is.

Literatuurlijst

https://mens-en-samenleving.infonu.nl/diversen/22626-hoe-skimmers-skimmen.html

https://www.pandasecurity.com/mediacenter/security/keyloggers-be-careful-what-you-type/

https://www.tomsguide.com/us/atm-hack-attack,news-28531.html

https://www.finder.com.au/how-to-protect-yourself-from-card-skimming

https://www.avg.com/en/signal/man-in-the-middle-attack

https://docs.microsoft.com/en-us/sql/relational-databases/security/authentication-access/database-level-roles?view=sql-server-2017