# 

# Beveiligingsplan ATM project 3 groep 2.



Andy Vos

0945183

Shivan Rambaran

0973515

Johnny Guo

0976269

Krzysztof Cajler

0951562

Ti2B

[**Beveiligingsplan ATM project 3 groep 2.**](#_6bjsn3no48pw) **1**

[**Voorwoord:**](#_azwvemenaxb3) **3**

[**Threat model information.**](#_137hjl6qpj1l) **4**

[**External dependencies**](#_2dydko1nlt09) **4**

[**Entry Points**](#_12upext2hutq) **5**

[**Assets**](#_mm13vhvlen5o) **6**

[**Trust Levels**](#_ixx3t4c7yyqm) **7**

[**Data Flow Diagram**](#_kwy2xv7ij7wm) **8**

[De overdracht:](#_kyfua5bttayr) 8

[**Determine and Rank Threats**](#_ffqriir4ce9g) **9**

[**Entiteiten**](#_x6at93yexmty) **12**

[**Threat Analysis**](#_t88rxi9uqwte) **14**

[**Shivan:**](#_ht6w7jhbirhi) **14**

[Attack tree 1:](#_7vf7bmvvx7bw) 14

[Attack tree 2:](#_lgdrmrj1fwh) 16

[Conclusie Attack tree 1:](#_blf9xmm95zxa) 18

[Conclusie Attack tree 2:](#_e1ab6fvcztcr) 18

[Bronnen individuele gedeelte:](#_sssrio4cl6v2) 19

[Johnny](#_2u1an03xm13z) 23

[Krzysztof](#_lcsj2361uumu) 26

[**Groepsgedeelte:**](#_6ohgvq49dwxl) **29**

[De (communicatie van en naar de) Linux server](#_oto1pvl32lz) 29

[De (communicatie van en naar de) database](#_opa4l6xyd7y7) 31

[De (communicatie van en naar de) Centrale bank.](#_cedsbc7jg7rv) 34

[Conclusie:](#_y22zcogcylrn) 35

[Bronnen groepsdeel:](#_ynl7d793rci6) 35

[**Bronnen voor het gehele document:**](#_rt8az33kwf60) **36**

# Voorwoord:

In periode 3 is de opdracht gegeven om een Pinautomaat/ATM te programmeren en fysiek in elkaar te zetten. Het programmeren wordt gedaan in project 3 en het fysieke ontwerp wordt ontworpen en in elkaar gezet in project 4. Hier is de beveiliging, zowel software, als fysiek erg belangrijk. Waar we op moeten letten om de pinautomaat zo veilig mogelijk te maken, hoe de aanvallers denken, etc. dit wordt opgenomen in dit document, genaamd het beveiligingsplan. Het document houdt in welke risico’s er zijn en hoe we denken deze risico’s te voorkomen.

Het beveiligingsplan spant over project 3 en 4, in andere woorden gaat dit over het totale eindproduct, lettend dat er geen versie van dit document of applicatie wordt bijgehouden omdat dit document eenmalig wordt opgesteld.

Uitleg van de hoofdstukken:

[**Threat model**](#_137hjl6qpj1l) **information**

De threat model information geeft het informatie over de applicatie weer. Van wie het is, welke versie, etc.

[**External dependencies**](#_2dydko1nlt09)

External dependencies zijn elementen buiten de applicatie die het applicatie kan bedreigen. Dit kan zijn van een server waarmee de applicatie communiceert, naar een arduino dat input verstuurt.

[**Entry Points**](#_dh0tfyytlbia)

Entry Points zijn potentiële punten dat aanvallers kunnen gebruiken om bepaalde onderdelen aan te vallen, denk hierbij bijvoorbeeld aan het inlogscherm op een online website.

[**Assets**](#_mm13vhvlen5o)

Assets zijn onderdelen die aanvallers willen en de developers willen beschermen, denk hierbij aan geld, reputatie, etc.

[**Trust Levels**](#_wmz9glc1qxyd)

Trust levels zijn niveaus van toegang die het rechten binnen een applicatie bepalen, bijvoorbeeld een anonieme gebruiker kan geen balans opvragen, storten of opnemen.

[**Data Flow Diagram**](#_7qsyugzvbxx)

Een data flow diagram is een is diagram die de dataoverdracht tussen de external dependencies laat zien, bijv. Van de ATM naar de webserver.

[**Determine and Rank Threats**](#_ffqriir4ce9g)

Dit is een lijst waar de mogelijke dreigingen worden opgesomd en gecategoriseerd op het model van STRIDE[1]

**Entiteiten:**

Met entiteiten wordt bedoeld wat zijn de mogelijke doelwitten van de aanvaller, dit kan zijn van een gebruiker waar het pasje van wordt gestolen tot de server die kan worden aangevallen.

**Threat Analysis**

In dit hoofdstuk wordt er dieper gekeken naar specifieke data flows of dreigingen op het individuele vlak en het groepsvlak van de bank.

1.STRIDE is een model dat wordt gebruikt bij het threat modeling, zie <https://www.owasp.org/index.php/CRV2_AppThreatModeling> voor meer uitleg.

# **Threat model information.**

|  |  |
| --- | --- |
| **Threat Model informatie:** | |
| Versie: | 1.0 |
| Naam: | The USSR National Bank. |
| Beschrijving: | Een applicatie en fysieke ontwerp van een pinautomaat waar mensen kunnen pinnen. |
| Document Beheerders: | Shivan Rambaran, Krzysztof Cajler,Johnny Guo, Andy Vos. |
| Deelnemers: | Shivan Rambaran, Krzysztof Cajler,Johnny Guo, Andy Vos. |
| Reviewer: | Mustafa Haijan, Daisy Hoffman, Gwen Maas. |

# **External dependencies**

|  |  |
| --- | --- |
| **Externe afhankelijkheden:** | |
| **ID:** | **Beschrijving:** |
| 1 | De applicate vraagt input door middel van hardware elementen die communiceren via arduino. |
| 2 | De applicatie communiceert met een Linux server op het domein van Hogeschool Rotterdam. |
| 3 | De applicatie communiceert met een MYSQL server die draait op een Linux server. |
| 4 | De applicatie communiceert met een centrale bank die draait op python. |

# 

# **Entry Points**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entry Points:** | | | |
| **ID:** | **Naam:** | **Beschrijving:** | **Trust Levels:** |
| 1.1 | Cardreader | Een RFID scanner die een pasje leest. | n.v.t |
| 1.1.1 | Card reader scherm | Het programma scant een pasje en haalt de bijbehorende gegevens uit de database. | (1) Anonieme gebruiker |
| 1.2 | Keypad | Een keypad die het invoer van de gebruiker leest. | n.v.t |
| 1.2.1 | Login scherm | Een scherm waar de gebruiker zijn pincode intypt. | (2) Gescande gebruiker |
| 1.2.2 | Snelkeuze Menu | Een menu dat verschillende opties aan de gebruiker laat zien. | (3) Ingelogde gebruiker |
| 2. | SSH port | De dataverkeer tussen de applicatie en de database gebeurt op basis van een SSH port. | n.v.t |
| 2.1 | Inloggen op de server | De applicatie logt in als gebruiker op de linux server. | (2) Gescande gebruiker  (3) Ingelogde gebruiker  (4) Root Admin  (5) Bank user |
| 3 | MYSQL | De applicatie krijgt toegang tot de database doormiddel door in te loggen op de mysql database | n.v.t |
| 3.1 | Data opvragen en sturen vanuit/ naar de database | De applicatie vraagt data op van de MYSQL database | (2) Gescane gebruiker  (3) Ingelogde gebruiker  (4) Root admin  (5) Bank user |
| 4. | Centrale Bank | De client (ATM) verstuurt een request naar de centrale bank. | (3) ingelogde gebruiker |
| 4.1 | Centrale bank naar de server | De centrale bank stuurt de request van de ATM door naar de Server waar de database op staat. | (3) Ingelogde gebruiker  (4) root admin  (5) Bank user |
| 4.2 | Centrale bank naar de ATM | De centrale bank stuurt de response van de server terug naar de ATM. | (4) root admin  (5) bank user |

# **Assets**

|  |  |  |  |
| --- | --- | --- | --- |
| **Assets:** | | | |
| **ID:** | **Naam:** | **Beschrijving:** | **Trust Levels:** |
| 1 | Gebruikers | Assets die behoren bij de gebruikers. | (2) Gescande gebruiker)  (3) Ingelogde gebruiker |
| 1.1 | Inloggegevens | Inloggegevens die de gebruikers gebruiken om geld te pinnen.  (pincode en UID van de kaart) | 1. (1) Anonieme gebruiker   (2) Gescande gebruiker  (3) Ingelogde gebruiker  (4) root admin  (5) Bank user |
| 1.2 | Persoonlijke data. | Het balans checken van de gebruiker, zowel de rekeningnummer. | (2) Gescande gebruiker  (3) Ingelogde gebruiker  (4) root admin  (5) Bank user |
| 1.3 | Geld | Het geld wat op de rekening van de gebruiker staat | (3) ingelogde gebruiker  (4) root admin  (5) Bank user |
| 2. | Systeem | Assets die behoren bij het systeem | (2) Gescande gebruiker)  (3) Ingelogde gebruiker  (4) root admin |
| 2.1 | Database | Gegevens in de database en de MYSQL database zelf. | (3) Ingelogde gebruiker  (4) root admin  (5) Bank user |
| 2.2 | API | De server/database API die de requests van de client (pinautomaten) handelt | (3) Ingelogde user  (4) root admin  (5) Bank User |

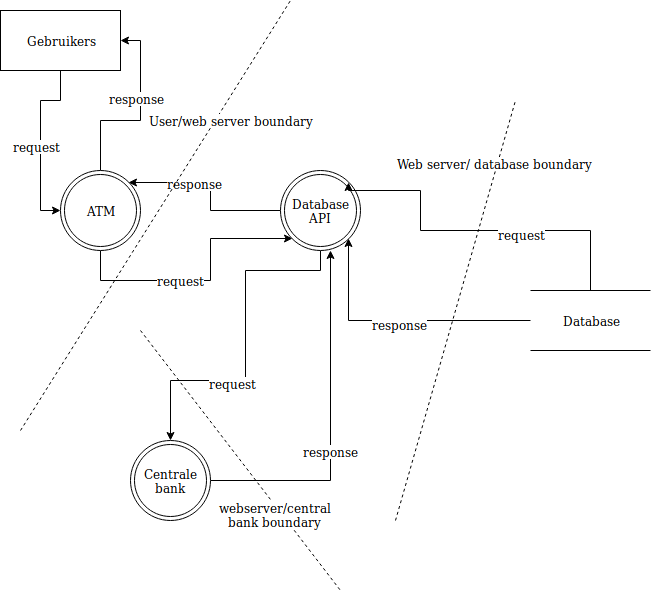
# 

# **Trust Levels**

|  |  |  |
| --- | --- | --- |
| **Trust Levels:** | | |
| **ID:** | **Naam:** | **Beschrijving:** |
| 1 | Anonieme gebruiker | Gebruiker die zijn pasje nog niet heeft laten scannen en nog niet ingelogd is. |
| 2 | Gescande gebruiker | Gebruiker die zijn pasje heeft laten scannen maar zijn pincode nog niet heeft ingetoetst. |
| 3 | Ingelogde Gebruiker | Gebruiker die door het login scherm heen is en zijn kaart heeft gescand |
| 4 | Root admin | Admin die de server beheert. |
| 5 | Bank user | De account waarop de applicatie inlogt op de MYSQL database. |

# 

# **Data Flow Diagram**

****

## De overdracht:

De overdracht van data gaat als volgt; een gebruiker scant zijn kaart door middel van een RFID-Reader, deze data wordt opgeslagen in een string en vervolgens fetcht de ATM via de database API de “ID” wat hier de data van de kaart is van de database, dit vergelijkt de ATM, als dit overeenkomt dan gaat de gebruiker door naar het loginscherm. Bij het loginscherm wordt de pin gevraagd met behulp van een keypad, de pin wordt hierna weer gefetcht uit de database en vergeleken, komt dit overeen dan mag de gebruiker storten/opnemen / (doneren) en het balans checken. Dit wordt ook weer gedaan door requests aan de database API die het balans aanpast als de gebruiker stort/opneemt (of doneert).

## 

# Determine and Rank Threats

De mogelijke gevonden bedreigingen worden gecategoriseerd op het model van STRIDE, dit houdt in dat er bij elke gevonden dreiging er gekeken wordt wat de aanvaller zou doen. Dit heeft betrekking tot **de gehele bank,** in de threat analysis wordt er dieper op ingegaan over hoe bepaalde datastromen van de individuele gedeelten van de bank kunnen worden beveiligd, ook zal er dieper gekeken worden naar de database en server, wat het groepsdeel inhoudt.

|  |  |  |  |
| --- | --- | --- | --- |
| **STRIDE:** | | | |
| **Threat voorbeeld:** | **Wat de aanvaller doet:** | **Toelichting:** | **Entiteiten:** |
| **Spoofing:** |  | Als iemand voordoen, zoals een bestand of als user (als BIll Gates bijv.) |  |
| User Spoofing | Een gebruiker nadoen. | Als klant van een bank vragen om transacties te doen. | (2) ATM |
| IP redirecting (IP-Spoofing) | Creëert een namaak server en redirect de requests naar die server. | De requests komen hierbij nooit aan bij de echte server. | (3) Server |
| Bank spoofing | De aanvaller doet zich voor als een Cliënt en stuurt betaalverzoeken naar klanten, wordt ook wel **Phishing** genoemd. | Met Client wordt hier bedoeld dat de aanvaller hier een bank nadoet. | (1)Gebruiker |
| **Tampering:** |  | Data of code/parameters van code modificeren. |  |
| Data redirecten | Het dataverkeer van de atm en/of server en/of centrale bank omleiden. | Dit kan door middel van het ip-adres waar de client naar toe verwijst te veranderen. | Communicatie tussen de:  (2) ATM  (3) Server  (4) Database  (5) Centrale Bank |
| Verander files | De ATM API veranderen zodat data doorgesluisd wordt.  -de files op de disk drive van de ATM veranderen. | De API zodanig veranderen dat de data verstuurd wordt naar een ander adres i.p.v naar de server waar de database op staat. | (2) ATM |
| Verander de files op de web server | De server API veranderen zodat de data bijv. Doorgestuurd wordt. |  | (3) Server |
| Verander files in de database | De waardes in de database worden verandert | Bijvoorbeeld dat het balans op 1000000000 euro wordt gezet van een gebruiker. | (4) Database |
| Skimming | De pas kopieren om zo meerdere pogingen te verkrijgen om in te loggen. | Dit gebeurt door een extra hardware element dat niet bij de bank hoort op de bank te zetten, hierbij wordt de code van het pasje gekopieerd. | (1)Gebruiker  (2) ATM |
| Data in de dataflow veranderen | De files die worden verstuurd vanaf de ATM of aankomen bij de ATM zodanig veranderen. | Bijvoorbeeld de balans veranderen als de gebruiker geld stort van 100 euro naar 1000 euro in de dataflow. | (4) Database |
| Plofkraak | Een methode om de ATM fysiek te beschadigen om vervolgens het geld uit de ATM te halen | Dit wordt meestal gedaan met grote objecten zoals auto’s | (2) ATM |
| **Repudiation** |  | Claimen iets niet te hebben gedaan. |  |
| Server logs verwijderen | Claimen dat de transactie nooit heeft plaatsgevonden door de server logs te verwijderen. |  | (4) Server |
| Claimen dat een transactie niet voltooid is. | Claimen dat de transactie niet voltooid is en geld terug eisen van de bank. | Hierbij meld de aanvaller dus ook dat hij/zij geld heeft verloren. | (2) ATM  (5) Database |
| Modificeer de dataflow van of naar de server | Data zetten/en of verwijderen. Zodat de transactie bijvoorbeeld nooit heeft plaatsgevonden | Bijv. een extra query hieraan toevoegen of de data blokkeren. | Communicatie tussen  (2) ATM  (3) Server  (4) Database  (5) Centrale bank. |
| **Information Disclosure** |  | Informatie aan iemand geven die het niet mag gebruiken/zien. |  |
| Vrijgeven database waardes | Het vrijgeven van de data in de database. | Data wordt hierbij geleaked, denk bijv. Aan websites zoals WIkiLeaks. | (3) Server  (4) Database |
| Dataflow aftappen/package sniffing | De dataflow afluisteren/aftappen via een programma (bijv. Wireshark). | Hierbij wordt gebruik gemaakt van programma’s zoals wireshark die de dataflow kan zien tussen client en server. | Communicatie tussen  (2) ATM  (3) Server  (4) Database  (5) Centrale bank. |
| **Denial of Service** |  | De service op welke manier dan ook belemmeren (de server laten crashen bijv,) |  |
| Server DDoS | Spamt de server API met requests, meestal SYN/ACK flood attack genoemd. | Hierbij wordt het afhandelen van de requests vertraagd.  Er komen steeds meer requests aan de deur van de server, waardoor het afhandelen langzamer verloopt. | (3) Server |
| De ATM Onklaar maken | De ATM zodanig beschadigen dat het niet meer gebruikt kan worden | Dit gebeurt meestal fysiek of door de ATM te laten crashen. | (2) ATM |
| **Elevation of Privilege** |  | Een gebruikersaccount met bevoegdheden bepaalde applicaties laten runnen. |  |
| Omzeilen van de autorisatie checks | Het omzeilen van het lezen van de keycard en de pincode. | Met omzeilen houdt in dat op een of andere manier de autorisatie checks worden omzeild en er meteen naar het opvragen van het balans wordt gegaan | (2) ATM |
| Een gebruiker logt niet uit | Een gebruiker loopt weg of logt na tijdens de transactie niet uit. |  | (1)Gebruiker |
| Root access/ login via man in the middle | Man in the middle attack waarbij iemand de aanvaller in laat loggen als root. |  | (3) Server  (4) Database |
| Privileges veranderen. | De privileges veranderen om toegang te krijgen tot gevoelige bestanden. |  | (2) ATM  (3)Server  (4) Database |

# **Entiteiten**

|  |  |  |
| --- | --- | --- |
| **Entiteiten:** | | |
| **ID:** | **Naam:** | **Beschrijving:** |
| 1 | Gebruikers | De klanten van de bank |
| 2 | ATM | De pinautomaat die draait op java |
| 3 | Server | De server die draait op linux |
| 4 | Database | De database op de server die draait op mysql |
| 5 | Centrale Bank | De centrale bank die berichten doorschuift op basis van land die draait op python |

# 

# **Threat Analysis**

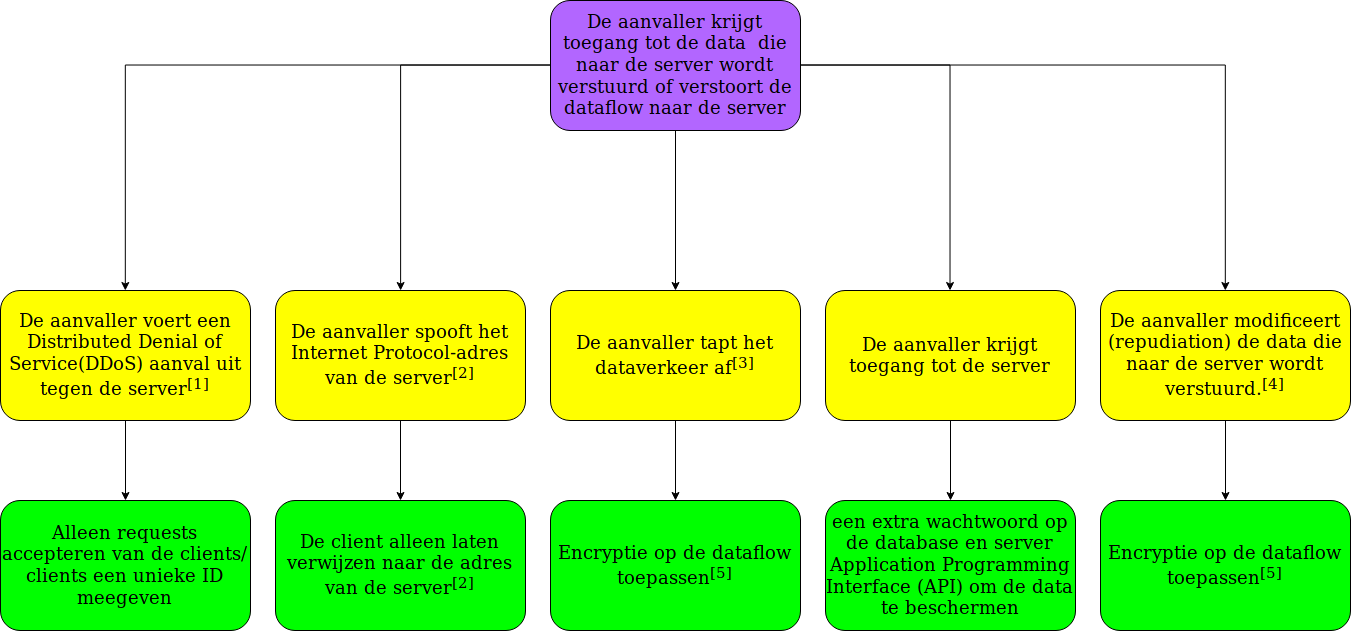
# Shivan:

Bij de threat Analysis wordt er vanuit de punt van de aanvaller gekeken over hoe de bank kan worden aangevallen. Als developer wil je juist dat je bank zo goed mogelijk beveiligt wordt tegen de methodes die de aanvaller gebruikt. Een aanvaller kan namelijk via verschillende hoeken aanvallen, bijvoorbeeld met geweld of via software. Aangezien het verboden is (in dit project) via geweld een bank te beroven blijft er maar 1 mogelijk aanvalscategorie over, namelijk het softwarematig hacken van de bank. De hoofdvraag hierbij luidt;Hoe kan het meest kwetsbare aanvalspunt waar de aanvaller toegang tot heeft beschermd worden als bank? Echter is deze vraag niet in een keer te beantwoorden, omdat er verschillende stappen aan te pas komen (we moeten immers denken als aanvallers). Enkele stappen die de aanvaller in zijn denkwijze zal meenemen zijn:

1. Op welke wijze gaat de aanvaller dit onderdeel aanvallen?
2. Wat wilt de aanvaller hiermee bereiken?
3. Wat zijn de gebruikte tegenmaatregelen die binnen de grenzen van dit project mogelijk zijn?

Deze zogenoemde stappen worden genomen als deelvragen om een duidelijk beeld te schetsen over hoe een aanvaller te werk gaat en hoe dit voorkomen kan worden. Deze deelvragen worden per attack tree behandeld om te onderzoeken welke van de aanval mogelijkheden in een attack tree het gevaarlijkst is voor bank om zo de hoofdvraag te beantwoorden.

### Attack tree 1:



**[1].**[**https://www.researchgate.net/publication/265294758\_A\_Review\_of\_DDOS\_Attack\_and\_its\_Countermeasures\_in\_TCP\_Based\_Networks**](https://www.researchgate.net/publication/265294758_A_Review_of_DDOS_Attack_and_its_Countermeasures_in_TCP_Based_Networks)

**[2].**[**https://www.ionos.com/digitalguide/server/security/ip-spoofing-fundamentals-and-counter-measures/**](https://www.ionos.com/digitalguide/server/security/ip-spoofing-fundamentals-and-counter-measures/)

**[3].**[**https://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets/**](https://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets/)

**[4].**[**https://www.owasp.org/index.php/CRV2\_AppThreatModeling**](https://www.owasp.org/index.php/CRV2_AppThreatModeling)

**[5].**[**https://www.avast.com/c-sniffer**](https://www.avast.com/c-sniffer)

**Op welke wijze valt de aanvaller aan als de aanvaller de data flow naar de server wilt aanvallen/verstoren?**

Ten eerste zal, als de aanvaller probeert de server te verstoren, een hele hoop requests of pakketjes naar de server sturen. Dit heet een DDoS(Distributed Denial of Service) aanval en zorgt dat de rij aan de “deur” van de server steeds langer wordt waardoor het afhandelen van de rij steeds langzamer gaat. Hier is er sprake van websockets, aangezien de verbinding tussen de cliënt en de Server(API) verloopt via websockets. Bij het tweede geval is er sprake van IP spoofing[2], de aanvaller neemt hierbij het ip-adres van de server en redirect de dataflow naar zijn eigen programma waar het IP-Adres aan gekoppeld is. Bij het derde geval tapt de aanvaller de communicatie tussen de ATM en de server af met behulp van een third party application. Denk bijvoorbeeld aan het programma Wireshark die onbeveiligde communicatie kan aflezen. Meestal gebeurt dit als de communicatie tussen apparaten niet beveiligd is[3], denk aan een publieke wifi netwerk dat zonder een wachtwoord is beveiligd. (De NS trein WiFi bijvoorbeeld). Als vierde kan de aanvaller door middel van bijvoorbeeld een man in the middle attack toegang krijgen tot de webserver. Als laatste kan de aanvaller de data zetten in de dataflow naar de server toe, deze data kan inhouden dat de request al is verzonden naar de server of dat de transactie is afgebroken.

**[1}.**<https://www.researchgate.net/publication/265294758_A_Review_of_DDOS_Attack_and_its_Countermeasures_in_TCP_Based_Networks>

**[2]**.<https://www.ionos.com/digitalguide/server/security/ip-spoofing-fundamentals-and-counter-measures/>

**[3]**.<https://www.comparitech.com/blog/information-security/what-is-packet-sniffing/>

**Wat wilt de aanvaller hiermee bereiken?**

Een aanvaller valt zijn doelwit niets voor niets aan, hij heeft altijd een motief om dit te doen. Zo wordt hier bekeken wat de aanvaller met zijn aanval wilt bereiken. Een DDoS aanval brengt vooral financiële beschadiging met zich mee, bijvoorbeeld het niet doen van aankopen bij van een bepaald product omdat de webshop niet bereikbaar is, Dit geldt ook voor de bank, namelijk dat deze financiële- en reputatieschade oploopt omdat klanten niet kunnen pinnen bij de bank en/of het proces van de transactie zeer traag verloopt. Bij het spoofen is er geen schade aan de bank zelf, er wordt niets gedaan tegen de server of de ATM, maar juist de gegevens van een gebruiker wordt hier gestolen. Hierdoor kan de aanvaller misschien inloggen op het account van de gebruiker wiens gegevens gestolen zijn en hiermee de account leegroven. Met deze methode kan de bank ook een flinke reputatieschade lijden, aangezien de data die hier wordt gelekt zeer gevoelig is en hierdoor klanten/gebruikers weglopen van de bank.Als de aanvaller toegang krijgt tot de server kan de server kan ontoegankelijk worden verklaard, omdat de aanvaller zowel de bestanden op de server kan aanpassen, en de database onbruikbaar kan maken/ kan vrijgeven/ leeg kan roven. Bij data injectie (of te wel repudiation genoemd) wordt er geen schade toegebracht aan de server of database zelf, wel zal er misplaatste informatie in de dataflow te vinden zijn waardoor de Server API in de war kan raken en de data alsnog zal verwerken of zal weggooien.

**Wat zijn de gebruikte tegenmaatregelen die binnen de grenzen van dit project mogelijk zijn?**

Tegen een DDoS aanval zijn bepaalde encryptiemethodes zeer gebruikelijk, zoals WPA2. Echter wordt dit meestal in niet publieke WiFi netwerken als voorzien. Ook kan er een rate limit worden ingesteld, dit houdt in dat een server maar naar maximaal aantal sockets tegelijkertijd kan luisteren of maar een aantal requests in een bepaalde tijd kan afhandelen[1].Tegen IP-Spoofing is als enige middel binnen dit project om de clients te laten verwijzen naar de Server address, aangezien een websocket een IP-adres en poortnummer nodig heeft om verbinding te maken met de Server(API). Verder kan er een extra waarde meegegeven worden aan de berichten naar de server die als een handshake dient om zo verdachte pakketten eruit te filteren. Tegen het aftappen (packet sniffing) kan encryptie worden gebruikt op de websockets. Gelukkig bestaat hier al een protocol voor genaamd SSL (Secure Sockets Layer)[2]. Tegen Man in the middle attack is helaas geen goede oplossing te verkrijgen, omdat je mede developers die je bank beveiligen je groepsgenoten zijn, dus een van de groepsgenoten laat de aanvaller inloggen.

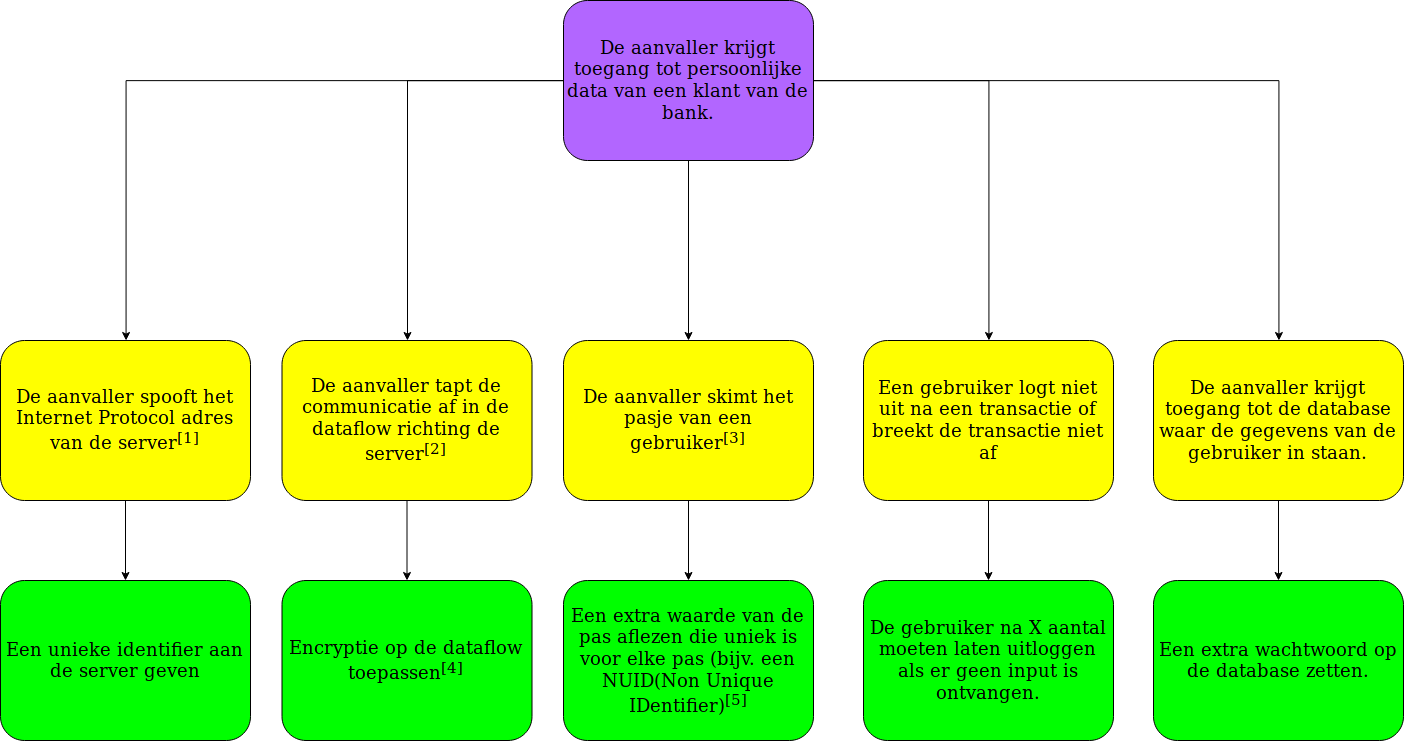
Tegen repudiation (dataflow modificeren) geldt in principe hetzelfde als packet sniffing door een SSL-encryptie op de verbinding toe te passen. Als de verbinding encrypt is, weet de aanvaller niet welke data hij moet veranderen.

**[1}.**[https://www.freecodecamp.org/news/how-to-secure-your-websocket-connections-d0be0996c556](https://www.freecodecamp.org/news/how-to-secure-your-websocket-connections-d0be0996c556/)

**[2]**.<https://www.techopedia.com/definition/24025/secure-sockets-layer-ssl>

**[3]**.<https://www.comparitech.com/blog/information-security/what-is-packet-sniffing/>

### Attack tree 2:



**[1].**<https://www.ionos.com/digitalguide/server/security/ip-spoofing-fundamentals-and-counter-measures/>

**[2].**<https://www.comparitech.com/blog/information-security/what-is-packet-sniffing/>

**[3].**<https://www.vraaghetdepolitie.nl/privacy/gegevensmisbruik/wat-is-skimmen.html>

**[4].**<https://www.freecodecamp.org/news/how-to-secure-your-websocket-connections-d0be0996c556/>

**[5].**<https://www.rfidcard.com/types-of-uid-rfid-card/>

**Op welke wijze valt de aanvaller aan in het geval dat de aanvaller toegang tot persoonlijke gegevens van de gebruiker wilt bemachtigen?**

Als een aanvaller de persoonlijke data van een gebruiker wilt bemachtigen kan dit op een aantal manieren: ten eerste kan de aanvaller het ip-adres van de server spoofen. Oftewel de aanvaller geeft een ip-adres op wat niet bestaat dat lijkt op het ip-adres van de server.Ten tweede kan de aanvaller tapt de dataflow af, dit wordt meestal gedaan door TCP verkeer tussen de server en de cliënt af te luisteren/ af te tappen en naar eigen behoeftes af te lezen[1]. Ten derde kan de aanvaller een pasje skimmen. Bij het skimmen wordt de magnetisch “ID” van een pasje gelezen. Het ID hier kan het UID of NUID zijn[2]. Deze wordt gekopieerd en worden hier meerdere pasjes van gemaakt, sinds het pasje zelf checkt of de pincode correct is en niet de database. Verder heeft de aanvaller vrij spel als de gebruiker niet uitlogt, hij kan namelijk via het account van de gebruiker grote hoeveelheden geld opnemen. Als laatste De aanvaller kan toegang krijgen tot de database als het de gebruikersgegevens van de database in bezit krijgt, dit kan via een man in the middle attack.

**[1]**.<https://www.comparitech.com/blog/information-security/what-is-packet-sniffing/>

**[2].**<https://www.rfidcard.com/types-of-uid-rfid-card/>

**Wat wilt de aanvaller hiermee bereiken?**

De aanvaller wilt natuurlijk bij de persoonlijke gegevens komen van de klant, meestal wilt hij hiermee geld van de rekening van de gedupeerde leegtrekken. Bij IP-spoofing is er alleen schade aan de gebruiker van de ATM door middel van het doorspelen van de data naar een nep ip-adres dat op die van de server lijkt. Er is bij het aftappen van de communicatie , eveneens, alleen schade bij de gebruiker van de ATM door middel van het doorspelen van de data naar de server die intercept wordt door de aanvaller. BIj het skimmen kan gegevens van de pas worden gestolen en worden gedupliceerd om zo door vele pogingen toegang te krijgen tot het account van de slachtoffer. Bij het niet uitloggen van een gebruiker kan het geld van het bankaccount snel worden leeggehaald. Als laatste als de aanvaller toegang krijgt tot de server kan hij De database teniet doen (vernietigd), al het geld op de database kan worden gestolen en persoonlijke gegevens kunnen worden gelekt.

**Wat zijn de gebruikte tegenmaatregelen die binnen de grenzen van dit project mogelijk zijn?**

Tegen IP-Spoofing kan een unieke code voor de ATM worden gebruikt, die de ATM als eerst naar de server stuurt als een soort van “ACK” (bevestigingssleutel als toegang) Tegen het dataverkeer aftappen kan Encryptie toegepast worden op de dataflow van- en naar de server.

Tegen het Skimmen kan je de NUID van de pasjes laten scannen in plaats van de UID, NUIDs zijn niet te kopiëren in vergelijking met UIDs en zijn nog steeds uniek, ook kan je een ontwerp toepassen waardoor het apparaat dat het skimmen doet niet over de kaart heen kan. Tegen het niet uitloggen is er de mogelijkheid om na x aantal minuten de gebruiker uit laten loggen als er geen input van de gebruiker zelf is. Als laatst is er tegen Het inloggen van de database een tweede encryptie laag toepassen/ een databasebeheerder toewijzen of privileges veranderen en root uitsluiten.

### Conclusie Attack tree 1:

De aanvaller valt vanuit verscheidene middelen aan om de data flow naar de server te verstoren. Dit kan zijn van DDoSSen naar de data modificeren. deze zijn te verhelpen door met name een extra wachtwoord of een SSL encryptie toe te passen op de dataflow in plaats van te communiceren door een SSH tunnel.

Door de SSL encryptie kunnen packet sniffers zoals wireshark niet bij de informatie komen, ook zorgt de SSL encryptie dat de ATM-server verbinding als een soort van VPN wordt gezien dat moeilijk te decrypten is.

### Conclusie Attack tree 2:

Bij het beveiligen tegen de aanvaller als deze de persoonlijke gegevens wilt bemachtigen is encryptie en een goed ontwerp van de ATM heel belangrijk, encryptie is om IP-spoofing en het packet sniffing/aftappen tegen te gaan en het ontwerp om het skimmen tegen te gaan.

### 

### Bronnen individuele gedeelte:

Mittal, A., Manoria, M., & Shrivastava, K. (2011, november). A Review of DDOS Attack and its Countermeasures in TCP Based Networks. Geraadpleegd op 10 april 2019, van <https://www.researchgate.net/publication/265294758_A_Review_of_DDOS_Attack_and_its_Countermeasures_in_TCP_Based_Networks>

Rao, R. (z.d.). An Intro To Denial of Service (DOS) Attacks & Countermeasures. Geraadpleegd op 10 april 2019, van <https://securitycommunity.tcs.com/infosecsoapbox/articles/2018/01/18/intro-denial-service-dos-attacks-countermeasures>

Imperva. (z.d.). What is a TCP SYN Flood | DDoS Attack Glossary | Imperva. Geraadpleegd op 12 april 2019, van <https://www.imperva.com/learn/application-security/syn-flood/?utm_campaign=Incapsula-moved>

IONOS. (2017, 30 augustus). IP Spoofing: Simple manipulation of data packets by attackers. Geraadpleegd op 11 april 2019, van <https://www.ionos.com/digitalguide/server/security/ip-spoofing-fundamentals-and-counter-measures/>

<https://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets/>

<https://www.owasp.org/index.php/CRV2_AppThreatModeling>

<https://www.avast.com/c-sniffer>

<https://www.comparitech.com/blog/information-security/what-is-packet-sniffing/>

<https://www.freecodecamp.org/news/how-to-secure-your-websocket-connections-d0be0996c556/>

<https://www.techopedia.com/definition/24025/secure-sockets-layer-ssl>

<https://securitycommunity.tcs.com/infosecsoapbox/articles/2017/04/17/all-you-need-know-about-stride-repudiation-threat>

<https://www.vraaghetdepolitie.nl/privacy/gegevensmisbruik/wat-is-skimmen.html>

<https://www.rfidcard.com/types-of-uid-rfid-card/>

**Andy:**

* Hoe kan de aanvaller het systeem bedreigen?

wat kan er worden aangevallen?

welke data wilt de aanvaller verkrijgen?

hoe kan dit voorkomen worden?

**Attack tree 1:**

## 

## 

**Attack tree 2:**

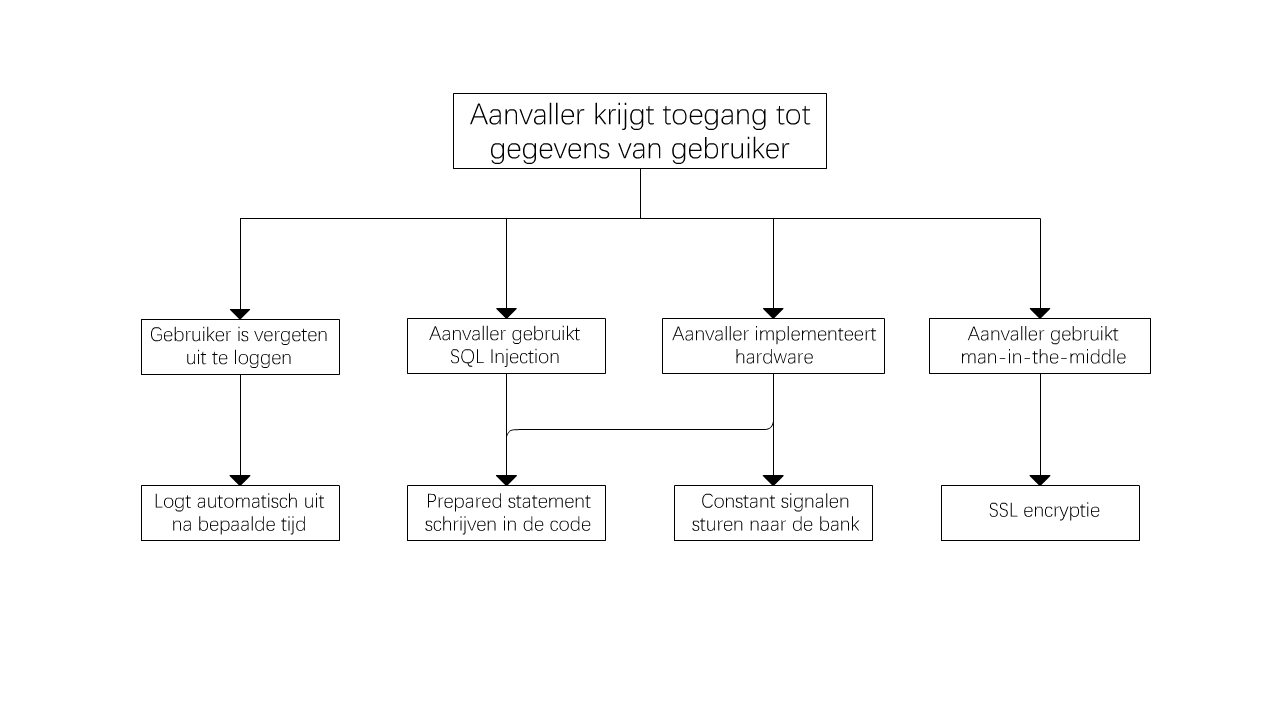
## 

## 

## Johnny

* Welke acties moet je ondernemen om het systeem te beveiligen?
  + Op welke manieren kan het systeem worden aangevallen?
  + Wat zijn de motieven voor deze aanvallen?
  + Hoe los je de kwetsbaarheden op?

Attack tree 1:

****

**Op welke manieren kan het systeem worden aangevallen?**

SQL-injectie oftewel SQL injection is een kwetsbaarheid van applicaties die gebruik maken van een database. Databases slaan informatie op met gebruik van SQL. SQL staat voor Structured Query Language. Het maakt voor de communicatie met het DBMS gebruik van een query. De query is een ASCII-tekenreeks en een opdracht stuurt het door naar het DBMS. Het DBMS voert de opdrachten uit en stuurt eventueel gegevens naar het opdrachtgevende programma. SQL-injectie kan uitgevoerd worden door in het kwetsbare invoerveld tekens invoeren die ervoor zorgen dat een ongewenste SQL-query wordt uitgevoerd. Meestal wordt er gebruik gemaakt van een apostrof. Hierdoor wordt de string die in de invoer wordt gebruikt verbroken, waardoor er vervolgens een SQL-statement kan worden toegepast.

Een bank kan fysiek worden aangevallen, zo kan de inbreker toegang krijgen tot de hardware of de kluis zelf door bijvoorbeeld een boor te gebruiken. De bank kan vanaf de hardware worden aangevallen. Zo kan ervoor worden gezorgd dat er de hardware geen contact meer maakt met de bank, ervoor zorgen dat de data dat naar de hardware wordt gestuurd wordt opgevangen of eigen data naar de bank wordt gestuurd, de aanvaller kan zijn eigen hardware implementeren zodat er data wordt verstuurd naar het systeem wat normaal niet kan, zo kunnen er SQL-statements worden geschreven.

Bij een onveilige verbinding is het mogelijk om tussen twee systemen die met elkaar communiceren af te luisteren. De gegevens van de gebruiker kan dan bekeken of aangepast worden. Dat wordt een man-in-the-middle-aanval genoemd.

**Wat zijn de motieven voor deze aanvallen?**

De aanvaller kan meerdere redenen hebben om verschillende aanvallen te plegen. Een van de redenen is concurrentie. Een bank heeft meerdere concurrenten en probeert zoveel mogelijk klanten te hebben. Maar wanneer blijkt wanneer er informatie is gelekt, daalt de reputatie van de bank met als gevolg dat de klanten overstappen naar hun concurrent.

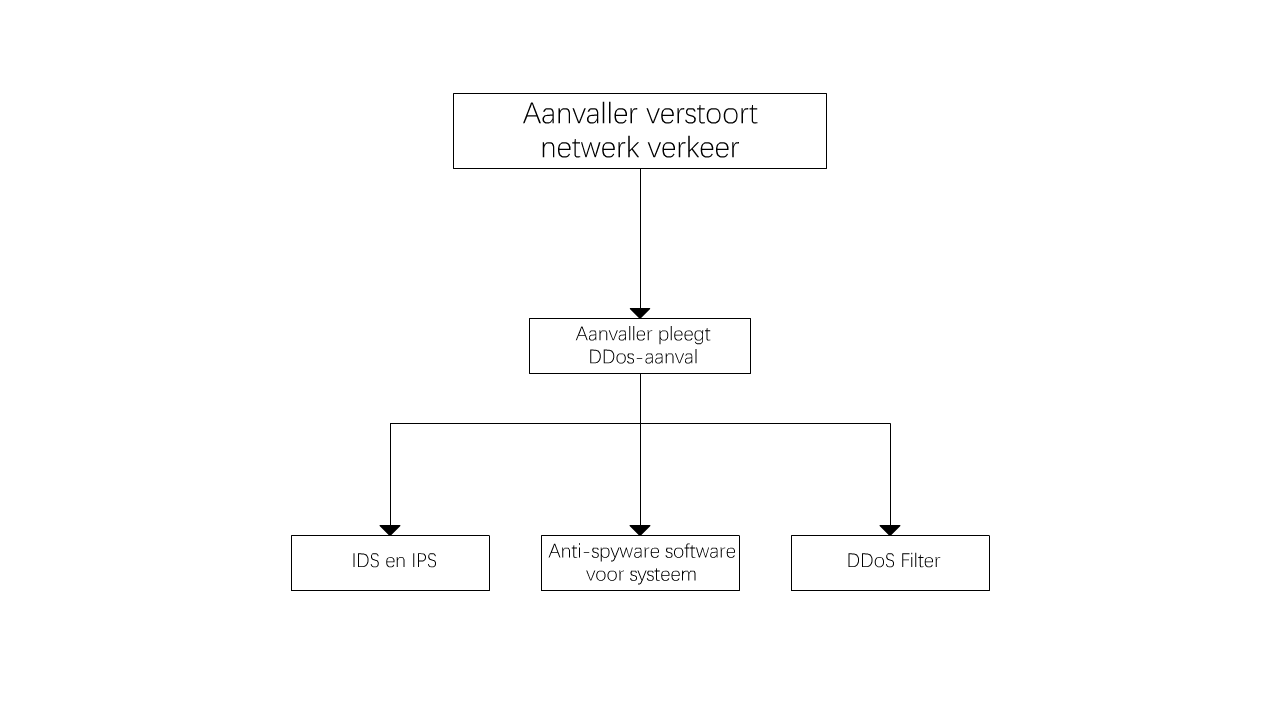
**Hoe los je de kwetsbaarheden op?**

Om SQL- injectie te voorkomen moeten de kwetsbaarheden weg. Dit kan door een “prepared statement” toe te voegen aan de code. Hiermee wordt de apostrof methode onmogelijk toe te passen. Op deze manier wordt de opdracht eruit gefilterd en zorgt dat de invoer een string blijft.

Het schrijven van een SQL-statement is onmogelijk met alleen een keypad met een bepaald aantal toetsen. Hier kan de aanvaller zijn eigen hardware implementeren. Ten eerste moet de hardware moet goed geïsoleerd worden door het in de kluis te stoppen. Ten tweede moet de hardware van de bank constant signalen sturen om ervoor te zorgen dat alles in orde is, wanneer er hardware wordt toegevoegd of wordt verwijderd, dan ontvangt de bank een signaal dat er ergens iets mis is en meteen de ATM uitschakelt.

De man-in-the-middle-aanval kan heel simpel worden opgelost met behulp van een SSL encryptie, zo wordt alleen contact gemaakt tussen de gebruiker en het systeem van de bank, waardoor het onmogelijk is voor de aanvaller ertussen te zitten.

Attack tree 2:



**Op welke manieren kan het systeem worden aangevallen?**

Een DDoS-aanval (distributed-denial-of-service-aanval) is een cyberaanval waarbij ontzettend veel verkeer naar computers, computernetwerken of servers wordt verstuurd. Hierdoor wordt het lastig voor een normale gebruiker gebruik te maken van de servers. Bij DDoS-aanvallen wordt er meestal gebruik gemaakt van een botnet. Een botnet is een netwerk van heel veel computers die besmet zijn met een malware. De hacker die de malware heeft verspreid kan nu vanaf een centrale punt zorgen dat de computers in de botnet opdrachten kan uitvoeren. Met alle computers een website bezoeken waardoor de server overbelast raakt en onbereikbaar wordt voor gebruikers. Hierdoor is het mogelijk om hele IT-infrastructuren plat te leggen.

**Wat zijn de motieven voor deze aanvallen?**

De motieven van de aanvaller kan van alle kanten op komen, maar de bekendste motieven om een DDoS-aanval te plegen zijn: revenge, concurrentie, politiek, omkoping en een afleiding.

**Hoe los je de kwetsbaarheden op?**

Voor DDoS-aanvallen kunnen niet voorkomen worden, want de botnet blijven met bestaan en kan op elk moment worden ingeschakeld, maar er zijn maatregelen die genomen kunnen worden om de gevolgen te verminderen. Door het beschikken van IDS (Intrusion Detection Systems) en IPS (Intrusion Protection Systems), een sterk anti-spyware software voor alle systemen met internetconnectie en filters toepassen wanneer de DDos-aanval plaatsvindt, zodat het niet gewilde netwerkverkeer weg wordt gefilterd.

**Conclusie**

Het systeem moet immuun zijn voor de volgende gevaren: SQL-injectie, DDoS-aanvallen, hardware implementatie en man-in-the-middle-aanvallen. Voor SQL-injectie moet er een prepared statement worden geschreven waardoor string niet wordt gebroken. Door het beschikken van een IDS, IPS, anti- spyware software en filters worden de gevolgen minder erg. De hardware van de bank moet constant signalen sturen naar het systeem om ervoor te zorgen of alles in orde is en de hardware intact is. Er moet een SSL encryptie aanwezig zijn tussen de verbindingen van de klant en het banksysteem.

Bronnen individuele deel:

<https://nl.wikipedia.org/wiki/Distributed_denial_of_service>

<https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.md>

<https://realhosting.nl/helpdesk/wat-is-een-ddos-aanval/>

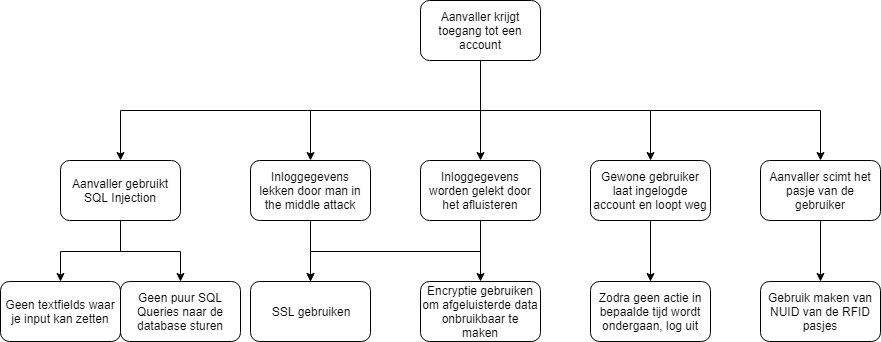
<https://www.bcs.org/content/ConWebDoc/8852>

<https://sites.google.com/a/pccare.vn/it/security-pages/dos-attacks-and-countermeasures>

<https://www.tripwire.com/state-of-security/security-data-protection/backdoors-hardware-attacks-prevention-detection-5/>

## Krzysztof

Attack Tree 1:



**Op welke manieren kan de systeem aangevallen worden?**

Een aanvaller heeft veel mogelijkheden om een systeem aan te vallen. In deze gedeelte zal ik me richten op de scenario dat een aanvaller toegang krijgt tot een account van een gebruiker. Er zijn meerdere manieren om dit te doen. Een daarvan is SQL injection. Wat ermee bedoeld wordt is dat je in een input balk een SQL Query invoert en dat die op de SQL server wordt uitgevoerd. Dat is heel gevaarlijk, omdat je zo alle informatie kan zien, aanpassen, maar ook verwijderen.

Een andere mogelijkheid om toegang tot een account te komen is het lekken van data die tussen client en server loopt. Dit kan door twee oorzaken komen. De ene is het afluisteren van data die niet versleuteld is en de andere is de man in the middle attack.

Het afluisteren van data kan heel makkelijk worden gedaan, het enige wat er nodig is, is de software zoals Wireshark. Daarmee kunnen pakketten worden ontvangen die door de netwerk gaan.

Man in the middle attack is net iets lastiger. Een aanvaller doet zich voor als een andere persoon. Tegen de server kan die zeggen dat hij de client is en tegen de client zegt die dat die de server is. Zo gaat alle data tussen de aanvaller. Daarop zal de encryptie niet genoeg zijn.

De 4e manier is heel simpel te voorkomen, maar moet wel geïmplementeerd worden, om dit te kunnen voorkomen. Het gaan namelijk om dat de gebruiker vergeet uit te loggen(transactie afbreken) en dus de aanvaller heeft toegang tot de account.

De laatste mogelijkheid is dat het pasje van de gebruiker wordt geskimt. Daarmee wordt bedoeld dat er data van zijn pasje kan worden gelezen en dan door de aanvaller gekloond.

**Wat zal aanvaller er aan hebben om een aanval te plegen?**

Een aanvaller kan diverse redenen hebben om een aanval te plegen. Zo kan de aanvaller geld willen stelen, de persoonlijke gegevens van de klanten of de reputatie van een organisatie te ondergaan. Bij deze aanval zal de aanvaller geld willen stelen. Daarnaast zal zo’n aanval de reputatie van de bank ondergaan, omdat de bank niet meer vertrouwbaar zal beschouwd worden.

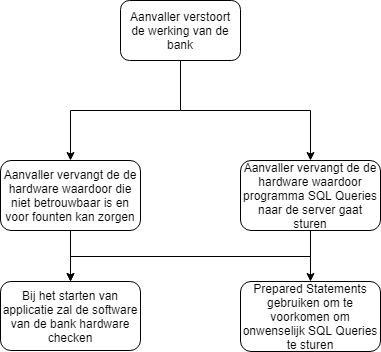
**Wat zijn praktische maatregelen die genomen kunnen worden om de aanval te voorkomen?**

Voor alle mogelijkheden voor dit soort aanval zijn er oplossingen die de applicatie veel veiliger zullen maken, maar ook betrouwbaarheid van de applicatie zal groter zijn. Tegen de SQL Injection wil ik twee maatregelen ondernemen, die de SQL Injection kunnen voorkomen. De eerste is dat in de programma geen Textfields zullen zijn(balken voor de invoer). Dat voorkomt dat er input van de toetsenbord kan komen. De enige input komt dan uit de keypad en die heeft alleen beperkte tekens die niet geschikt zijn om een SQL Injection aanval te doen. Op zich zal dit genoeg zijn, maar voor de zekerheid zullen er ook geen puur SQL Queries naar de server gestuurd zijn. De GUI zal alleen soort van opdrachten sturen en de programma op de server zal deze opdrachten vertalen naar SQL queries en SQL server laten uitvoeren.

Daarnaast kan de gebruiker vergeten om uit te loggen. Daarvoor is de oplossing bedacht om gebruiker vanzelf uit te loggen als die actie ondergaat in bepaalde tijd. Er zal wel genoeg tijd moeten zijn voor gebruiker om een actie onder te kunnen gaan.

De laatste is dat het pasje van de gebruiker wordt geskimt. Mijn applicatie gebruikt RFID wat niet erg veilig is, maar het kan wel veilig gemaakt worden door slim te zijn en alleen veilige data van RFID kaarten te gebruiken. Daarmee wordt bedoeld dat er NUID zullen gebruikt worden omdat die uniek zijn en die ook niet gekloond worden.

Attack Tree 2:



**Op welke manieren kan de systeem aangevallen worden?**

De pinautomaat kan op twee manieren worden aangevallen om de werking te verstoren. Beide manieren gaan om de hardware. Wat de aanvaller doet is dat die de hardware omwisselt. Zijn hardware zal bijvoorbeeld andere getallen opvangen als er op de knopjes wordt gedrukt. Dat zorgt dat de pincodes niet meer zullen kloppen. De andere manier is dat de microcontroller SQL Queries zal sturen en programma zou het onbedoeld naar SQL server sturen wat niet de bedoeling is.

**Wat zal aanvaller er aan hebben om een aanval te plegen?**

Dat hangt af wat de aanvaller zal doen. Zodra hij/zij bijvoorbeeld de nummers van de keypad verandert en GUI zal dus verkeerde getallen opvangen, kan dit leiden tot gefrustreerde klanten die hun eigen rekeningen blokkeren. In tegenstelling de andere manier is veel gevaarlijker, omdat SQL Queries kunnen gestuurd worden. Dat kan leiden tot het laten zien, manipuleren of verwijderen van data.

**Wat zijn praktische maatregelen die genomen kunnen worden om de aanval te voorkomen?**

Om hardware wisseling te voorkomen, kan er een check worden gedaan bij het starten van de GUI, die zal bepaalde phrase of een bericht van de microcontroller krijgen, anders zal die niet werken. Tegen de onwenselijk Queries

**Conclusie**

Om de GUI te beveiligen is wel encryptie met SSL nodig om nog man in the middle attacks te voorkomen. Verder is het heel handig om prepared statements gebruiken of een API gebruiken die berichten in SQL Queries gaat omzetten. Verder is er een bescherming nodig om gebruikers uit te loggen zodra ze geen actie ondernemen. Tegen skimmen kan er NUID gebruikt worden en tegen aanvallers hardware kunnen check worden gedaan om zijn hardware niet gebruikelijk te maken.

# 

# Groepsgedeelte:

Het groepsgedeelte bestaat uit het beveiligen van de gebruiker en de ATM, de database, server en de communicatie van en naar de centrale bank. Deze 4 onderdelen worden apart van elkaar uitgewerkt en per onderdeel wordt er gekeken naar:

* Hoe efficiënt mogelijk kan tegen deze dreigingen worden beveiligd?
  + Op welke manieren kan de aanvaller dit onderdeel aanvallen? (denkend aan STRIDE)
  + Wat kan de aanvaller als hij/zij eenmaal binnengedrongen/de aanval succesvol is?
  + Hoe groot is de risico op de aanvallen en wat zijn de tegenmaatregelen voor dit onderdeel?

## De (communicatie van en naar de) Linux server

De linux server draait op het domein van Hogeschool Rotterdam, deze server is te benaderen door een SSH port die beveiligd is met een gebruikersnaam en wachtwoord om ongewenst gebruikers buiten te houden.

|  |  |  |
| --- | --- | --- |
| **Op welk manier kan de aanvaller de Linux server aanvallen?** | | |
| Soort aanval: | Beschrijving: | Extra toelichting: |
| 1.Server DDoS | De server laten overspoelen met SYN/ACK requests. | De server kan maar een request tegelijkertijd handelen, overspoelen met requests betekent een langere wachtrij. |
| 2.IP of Client Spoofing | De aanvaller heeft hierbij een nep IP-adres wat hetzelfde lijkt op de IP van een client waarmee de server verbonden mee is. |  |
| 3.Login via man in the middle | Iemand laat de aanvaller inloggen op de server. |  |

|  |  |  |
| --- | --- | --- |
| **Wat kan de aanvaller als hij/zij binnengedrongen is/de aanval succesvol is op de Linux server?** | | |
| Soort aanval: | Beschrijving: | Extra toelichting: |
| 1.Server DDoS | De aanvaller krijgt hier niets materieels, alleen waardering voor zijn aanval en reputatieschade aan de bank. |  |
| 2.IP of Client Spoofing | De aanvaller kan hiermee waardevolle gegevens krijgen, zoals pincodes/ UIDs etc. |  |
| 3.Login via man in the middle | De aanvaller kan alle files veranderen op de server of zelfs de database uitzetten als hij/zij als root ingelogd is. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Wat is het risico en wat zijn de tegenmaatregelen om de Linux server te beschermen tegen deze aanvallen?** | | | |
| Soort aanval: | Risico | Tegenmaatregel: | Beschrijving |
| 1.Server DDoS | MEDIUM | 1. Encryptie mechanismes gebruiken zoals WPA2, echter wordt dit door de meeste wifi netwerken al door voorzien. 2. Beveiligen tegen TCP SYN/ACK floods door middel van cookies: 3. **SSL connectie met behulp van certificates** | De database API zal verbinding maken met de linux server met een SSL verbinding, dit is een encryptiemethode waarbij de certificaten |
| 2.IP of Client Spoofing | HIGH | Een unieke waarde meegeven aan elke cliënt of de server dat werkt als een soort van herkenningsteken. | Door deze unieke waarde kan de server Andere clients negeren die deze sleutel niet hebben. |
| 3.Login via man in the middle | LOW | Hiervoor is geen oplossing bekend, aangezien een van de developers zelf meewerkt aan de aanval | De developer zelf heeft toegang tot de database en server. |

## De (communicatie van en naar de) database

De database draait op een MYSQL server die draait op een linux server, om toegang hiertoe te krijgen heeft de gebruiker van de database het gebruikersnaam en wachtwoord nodig die voor de database is ingesteld, hierdoor is ongewenste toegang uitgesloten.

|  |  |  |
| --- | --- | --- |
| **Op welk manier kan de aanvaller de database aanvallen?** | | |
| Soort aanval: | Beschrijving: | Extra toelichting: |
| 1.SQL Injectie | De aanvaller manipuleert een SQL query die wordt verzonden naar de database | Voornamelijk gedaan via text fields. |
| 2.Login via man in the middle | De database is beveiligd met een wachtwoord, echter als iemand die deze wachtwoord weet de aanvaller hiermee laat inloggen kan de aanvaller tot de gegevens komen. |  |
| 3.Package sniffing | De aanvaller gebruikt een programma zoals wireshark om de dataflow te zien en te onderscheppen van en naar de bank | Wireshark is een programma waarmee een dataflow tussen client en server kan worden gelezen. |

|  |  |  |
| --- | --- | --- |
| **Wat kan de aanvaller als hij/zij binnengedrongen is/de aanval succesvol is op de database?** | | |
| Soort aanval: | Beschrijving: | Extra toelichting: |
| 1.SQL Injectie | Als de SQL injectie succesvol is, kan de aanvaller met de statement UNION meerdere queries op de database uitvoeren om vervolgens |  |
| 2.Login via man in the middle | De aanvaller verkrijgt toegang tot de data op de database waarmee de hij/zij kan doen met de database wat hij of zij wilt |  |
| 3.Package sniffing | Er komen hier meerdere dingen aan de orde:   1. De packets worden gelezen en vervolgens geleaked naar buiten. 2. De packets worden gemodificeerd (tampering). 3. Aan de hand van de packets kan data worden afgelezen waarmee de aanvaller andere doelwitten kan aanvallen. | 3. Met andere doelwitten wordt bedoelt dat de aanvaller dan informatie beschilkt om de database of centrale bank aan te vallen. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Wat is het risico en wat zijn de tegenmaatregelen om de database te beschermen tegen deze aanvallen?** | | | |
| Soort aanval: | Risico | Tegenmaatregel: | Beschrijving |
| 1.SQL Injectie | MEDIUM | 1. Filters zetten op de query. 2. Geen input vragen vanuit text velden in de cliënt. | SQL injectie vindt vooral plaats als er ruimte is om een een query uit te voeren, bijv. In een tekstveld. |
| 2.Login via man in the middle | LOW | Hiervoor is geen oplossing bekend, aangezien een van de developers zelf meewerkt aan de aanval | De developer zelf heeft toegang tot de database en server. |
| 3.Package sniffing | HIGH | De verbinding beveiligen met SSL. | Door de encryptie van SSL verbindingen kunnen programma’s zoals wireshark de dataflow niet aftappen. |

## 

## De (communicatie van en naar de) Centrale bank.

De Centrale bank is een API geschreven om berichten door te sturen naar de juiste bank waar de cliënt bij hoort. De bank doet verder niets, alleen berichten doorsturen.

|  |  |  |
| --- | --- | --- |
| **Op welke manier kan de aanvaller de communicatie/datastroom van en naar de centrale bank aanvallen/hacken?** | | |
| Sport aanval: | Beschrijving: | Extra toelichting: |
| 1.Package sniffing | De aanvaller gebruikt een programma zoals wireshark om de dataflow te zien en te onderscheppen van en naar de bank | Wireshark is een programma waarmee een dataflow tussen client en server kan worden gelezen. |
| 2.IP Spoofing | De aanvaller Spooft het IP van de centrale bank. |  |

|  |  |  |
| --- | --- | --- |
| **Wat kan de aanvaller als hij/zij binnengedrongen is/de aanval succesvol is met betrekking tot de communicatie tot de centrale bank?** | | |
| Soort aanval: | Beschrijving: | Extra toelichting: |
| 1.Package sniffing | Er komen hier meerdere dingen aan de orde:   1. De packets worden gelezen en vervolgens geleaked naar buiten. 2. De packets worden gemodificeerd (tampering). 3. Aan de hand van de packets kan data worden afgelezen waarmee de aanvaller andere doelwitten kan aanvallen. |  |
| 2.IP Spoofing | De aanvaller krijgt een packet waarin de UID/pincode of gegevens voor het balans in wordt verandert. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Wat is het risico en wat zijn de tegenmaatregelen om de datastroom van en naar de centrale bank te beschermen tegen deze aanvallen?** | | | |
| Soort aanval: | Risico | Tegenmaatregel: | Beschrijving |
| 1.Package sniffing | HIGH | De verbinding beveiligen met SSL. | Door de encryptie van SSL verbindingen kunnen programma’s zoals wireshark de dataflow niet aftappen. |
| 2.IP Spoofing | HIGH | Een unieke waarde meegeven aan elke client of de server dat werkt als een soort van herkenningsteken. | Door deze unieke waarde kan de server Andere clients negeren die deze sleutel niet hebben. |

## Conclusie:

Tot slot blijkt er dat de meeste gevaren liggen bij het packet sniffing/ spoofing en DDoSSing, dit kan allemaal voorkomen worden door, in plaats van een SSH tunnel te gebruiken, te gaan voor SSL encryptie. SSL lost al gauw heel veel problemen op met het betrekking tot de communicatie, dit maakt de data flow ontoegangkelijk voor packet sniffers en spoofing. Voor het DDoSSen bestaan al verschillende soorten software die filteren SYN/ACK flood attacks (waar DDoSSen uit bestaat).

## Bronnen groepsdeel:

<https://www.certifiedsecure.com> (cheat sheets/ instructie videos)

Digicert. (z.d.). What Is SSL (Secure Sockets Layer)? | DigiCert.com. Geraadpleegd op 12 april 2019, van <https://www.digicert.com/ssl/>

Facts About Port Scanning. (z.d.). Geraadpleegd op 12 april 2019, van <https://whatismyipaddress.com/port-scan>

Hacking Linux OS: Complete Tutorial with Ubuntu Example. (2019, 2 april). Geraadpleegd op 12 april 2019, van <https://www.guru99.com/hacking-linux-systems.html>

# Bronnen voor het gehele document:

OWASP. (z.d.). CRV2 AppThreatModeling - OWASP. Geraadpleegd op 10 april 2019, van <https://www.owasp.org/index.php/CRV2_AppThreatModeling>

Peeples, K. (2015, 2 december). STRIDE Threat Model. Geraadpleegd op 11 april 2019, van <https://dzone.com/articles/stride-threat-model>

Shostack, A. (2014). Threat Modeling with STRIDE [Powerpoint presentatie]. Geraadpleegd op 11 april 2019, van <https://users.encs.concordia.ca/~clark/courses/1601-6150/scribe/L04c.pdf>

Wagenseil ·, P. (2018, 14 november). Hacking an ATM Is Shockingly Easy. Geraadpleegd op 12 april 2019, van <https://www.tomsguide.com/us/atm-hack-attack,news-28531.html>

Wat is skimmen? (z.d.). Geraadpleegd op 12 april 2019, van <https://www.vraaghetdepolitie.nl/privacy/gegevensmisbruik/wat-is-skimmen.html>