Red vs Blue event

Internetbankieren

Bedrijf: Fontys ICT

Date: 26.10.2020

Colofon

Label	Gegevens
Klant	Marc van Bommel
Auteur	Thomas van Heel
Status	Afgerond
Classificatie	Vertrouwelijk

Versiegeschiedenis

Versie	Datum	Aanpassingen
1.0	28-10-2020	Setup en vullen met inhoud

Hoofdstuk 1

Management samenvatting

Fontys ICT heeft gevraagd, aansluitend op de Red - vs Blue team, een tijd-gelimiteerde pentest uit te voeren op de Internetbankieren applicatie. Dit is gebeurt om de testomgeving in Seclab.

Mijn impressie van de veiligheid van de applicatie is verdeeld. De maker van de applicatie heeft tijdens het maken van de applicatie, de '*hacker mindset*' niet in gedachte genomen. Zo kan je als gebruiker van de applicatie negatieve bedragen overmaken, wat dus inhoud dat er geld op je eigen rekening bijkomt. Met andere woorden, ik heb een aantal kwetsbaarheden/aandachtspunten gevonden in de applicatie.

In de webapplicatie is er de mogelijkheid om geld van een andere rekening af te halen en op je eigen rekening te zetten. In deze applicatie gaat het om erg gevoelige informatie, dus kan dit ernstige gevolgen hebben.

Ondanks dat, wil ik aangeven dat de maker van de applicatie wel een goede toepassing heeft omtrent encryptie, decryptie en filtering van de (gevoelige) informatie. Echter, moet er wel voor gezorgd worden dat de gebruiker van de applicatie "feedback" krijgt als diegene verkeerde data invoert.

In dit rapport vind je alle details van het onderzoek en bevindingen, dit bevat ook het technische advies.

1.1 Overzicht bevindingen

De bevindingen zijn gesorteerd op basis van de classificatie. Indien er meer informatie benodigd is kan Bijlage A ingezien worden.

Bevinding	**Risico**
Invoer filtering - Elke gebruiker van de applicatie kan geld van een willekeurig bankrekening overmaken naar zijn/haar rekening.	
Zwakke authenticatie - ter bevestiging bij het overboeken.	
Slechte error afhandeling/logging van gegevens	

Inhoudsopgave

Hoofdstuk 2

Aanvraag

Vandaag de dag is internet een belangrijk aspect in de maatschappij, zo ook het internetbankieren. Deze applicatie vervuld deze functionaliteit, hiermee kan je gemakkelijk je financiële zaken regelen zoals overboeken.

De stakeholder van dit project zijn Marc van Bommel, Fontys ICT en Thomas van Heel.

2.1 Onderzoeksvraag

De applicatie die getest getest gaat worden is de internetbankieren app(10.10.1.110:4200) van Marc van Bommel, deze staat op de Seclab omgeving van Fontys. Het doel is om de data encryptie en de invoer filtering te testen.

2.2 Scope

De scope van het project is om de data encryptie en de invoer filtering in de web applicatie te testen, gehost op Seclab.

Hoofdstuk 3

Bevindingen

3.1 Aanpak

Eerst is er een connectie gemaakt met de VPN voor de Seclab omgeving, zo kon er getest worden. De cyber killchain is gevolg namelijk:

1. Reconnaissance
2. Intrusion
3. Exploitation

Er zijn meer stappen, maar die zijn nu niet van toepassing, omdat ik niet verder kan gaan dan exploitatie. Met deze stappen, was ik bereid om de zwakheden in de applicatie te achterhalen.

3.1.1 Algemeen

Om alles te testen wat in de scope zit, is er eerst een vorm van scannen gedaan.

3.1.2 Web

Testen is gedaan via Seclab, hier werd de applicatie op gehost. De tools die gebruikt zijn bij de pentest zijn voornamelijk de tools die Kali Linux bevat en die ik gebruik voor alle pentesten. De tools zijn:

- Burp Suite
- Nmap

3.2 Analyse

Onderstaand is een beschrijving gegeven van elk probleem. De technische uitleg is in de bijlage te vinden om beter te begrijpen wat verkeerd ging en wat beter zou moeten.

1. **(Zwakke) authenticatie**
   Indien een gebruiker geld wil overmaken naar een andere rekening kan dit gedaan worden zonder enige vorm van authenticatie.

2. **Slechte error afhandeling**
   Als er een error voorkomt, heeft het bericht van de error te veel informatie in zich. Delen van de code of andere informatie is zichtbaar voor gebruikers. Dit komt, omdat errors slecht zijn afgehandeld. Hackers kunnen hier zijn voordeel mee doen.

De gebruiker krijgt ook logging te zien van zowel errors als zijn/haar huidige bankrekening. Er wordt wel encryptie gebruikt, maar dit is niet wat je wil aangezien hackers hier voordeel uit kunnen halen.

3. Invoer filtering

Een hacker kan geld overmaken naar van een willekeurige rekening(in de database), naar zichzelf. Dit komt door het gebrek aan invoer filtering, want anders zou dit afgevangen kunnen worden.

Bovendien zitten er ook geen checks/filtering in het daadwerkelijke bedrag wat je wil overmaken. Zo kan ik een bedrag van -10,0 euro naar iemand overmaken, wat dus inhoudt dat je 10,0 euro zou krijgen.

Hoofdstuk 4

Conclusie en advies

Concluderend uit de pentest, kan worden vastgesteld dat de applicatie niet geheel veilig is zoals gedacht.

4.1 Conclusie

De applicatie heeft een gebrek aan input filtering, checks en error afhandeling. Bovendien dient de applicatie een extra authenticatie stap te bevatten en moet de logging verwijderd worden uit de applicatie. Dit bij elkaar geeft hackers meer aanvalsvectoren en dat is niet wat je wil, vooral niet in een internetbankieren applicatie.

4.2 Advies

Het beste advies wat gegeven kan worden op basis van de pentest, is om meer te testen voor je de applicatie openbaar maakt. Hiermee voorkom je dat er slordigheidsfouten in de applicatie zitten, zoals error afhandeling, gebrek aan checks en input filtering. Voor nu is het advies om de applicatie te verbeteren op basis van de bevindingen die in dit rapport vermeld staan. In de toekomst dient er een continuïteit omtrent testen te worden toegepast.

Bijlage A

Technische bevindingen

(Zwakke) authenticatie

CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U/C:L/I:L/A:H/RC:C/CR:H/IR:H/AR:M/MAV:L/MAC:L/MPR:L/MUI:R

Reproductie

Er is een kwetsbaarheid gevonden in de authenticatie van het overmaken van geld. Zodra je bent ingelogd controleert het systeem niet meer of jij daadwerkelijk die persoon bent die gekoppeld zit aan die rekening.

Als alle velden ingevuld zijn en je selecteert de optie “overmaken”, volgt er geen authenticatie controle.

NOTITIE: Er wordt gebruik gemaakt van HTTP, dus zodra een hacker een MITM attack uitvoert en de data onderschept heeft hij het account. Hiermee kan hij dan geld overmaken naar zichzelf.

Risico's

Het risico is dat, een hacker geld kan overmaken naar welke rekening dan ook, als hij het account gekraakt heeft op een manier. Het gevolg hiervan is dat je rekening leeg getrokken kan worden en de hacker jouw geld(wellicht via het hoppen naar verschillende rekeningen) naar zijn rekening stuurt.

Preventie

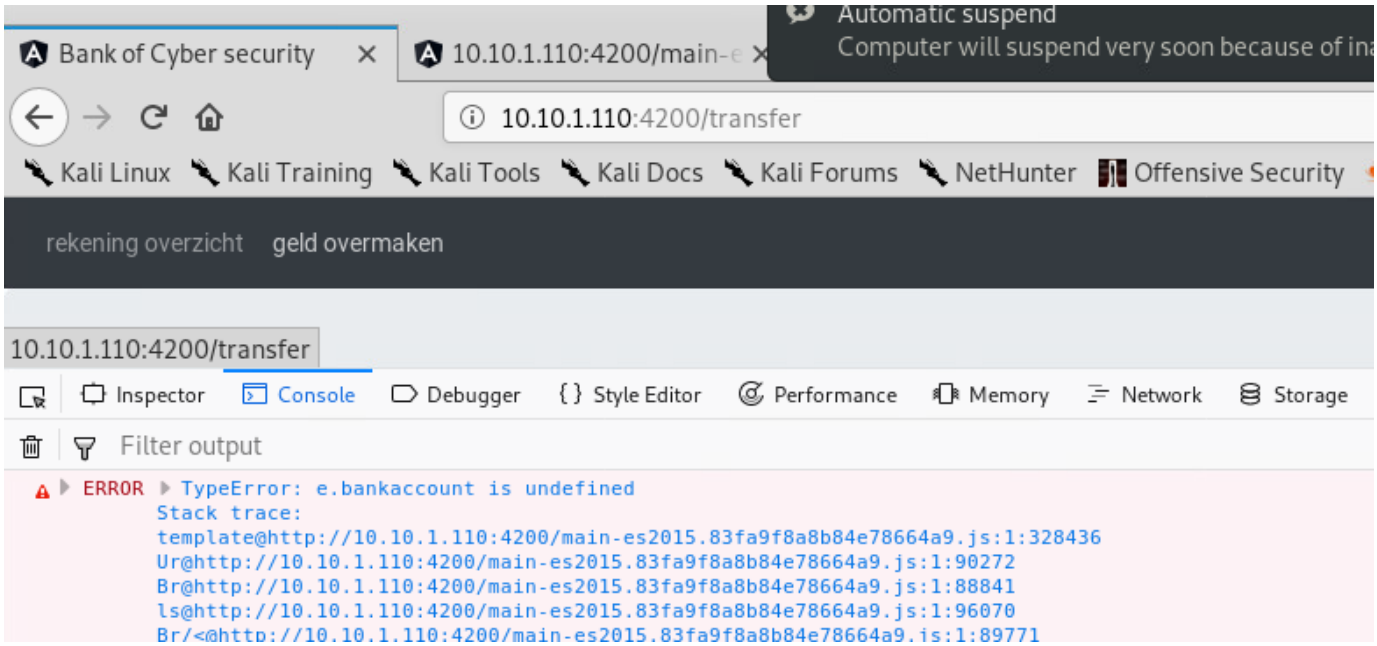
De gebruiker moet wel inloggen op bij zijn/haar rekening te komen (authenticatie), maar toch is het verstandig een extra authenticatie stap toe te voegen bij het overmaken van geld. Denk hierbij aan een pincode, patroon of iets dergelijks. Dit weet dan echt alleen de gebruiker zelf, hierdoor kan er niet ongewild geld overgemaakt worden.

Slechte error afhandeling

0 (INFO)

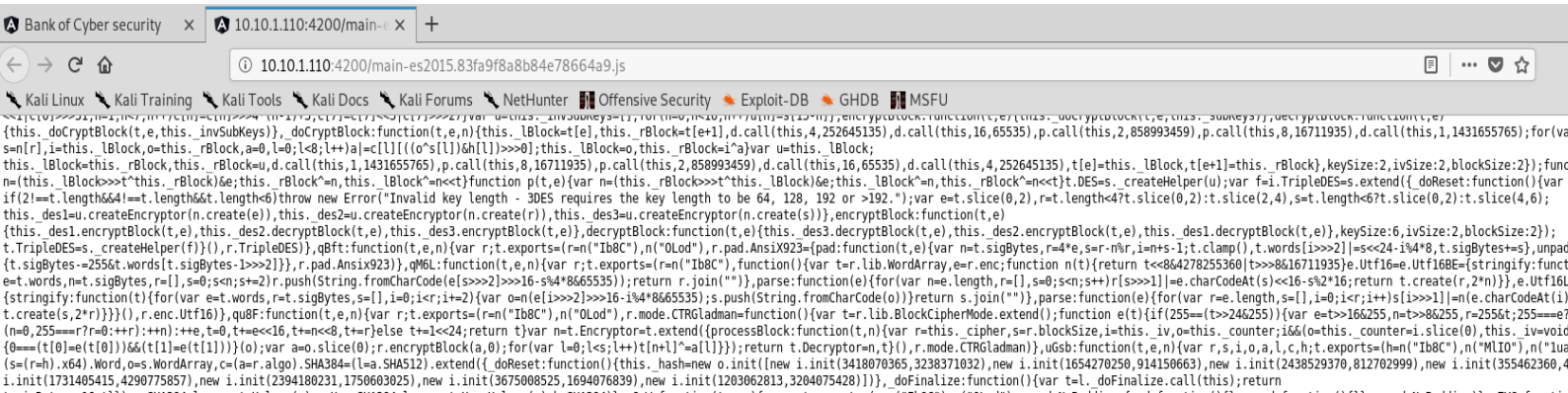
Reproductie

Errors moeten goed afgehandeld worden. Wanneer er een error voorkomt, wil je niet dat de gebruiker dit te zien krijgt of erger nog dat je applicatie crasht. In plaats hiervan, moet het programma intern de error afhandelen en verder gaan op een of andere manier zonder te veel data weg te geven aan de gebruiker.



In de afbeelding hierboven zie je een voorbeeld van een error die wordt weergegeven in de console. Dit geeft weer dat de error afhandeling in de applicatie niet helemaal clean is. Dit leidt tot het laten zien van informatie die niet zichtbaar zou moeten zijn voor de gebruiker.

Hieronder is te zien dat ik via de errors in de console bij de *main.js* kan komen. Hierin staat delen van code, waarin ik ook heb achterhaald dat er AES wordt gebruikt



Ook kan ik de gegevens in de console zien omtrent mijn bankrekening:

Hoewel dit gedecrypt is, zou ik dit niet weergeven in de console. Dit is wellicht iets kleins, maar als dit grootschalig zou gebeuren biedt het de hacker meer aanvalsvectoren.

Preventie

Ben er zeker van dat alle mogelijke scenario's afgevangen zijn. Als er toch een voorkomt, zorg ervoor dat het probleem duidelijk is zonder informatie weg te geven. Laat de gebruiker een errorpagina zien met wat er fout ging.

Risico's

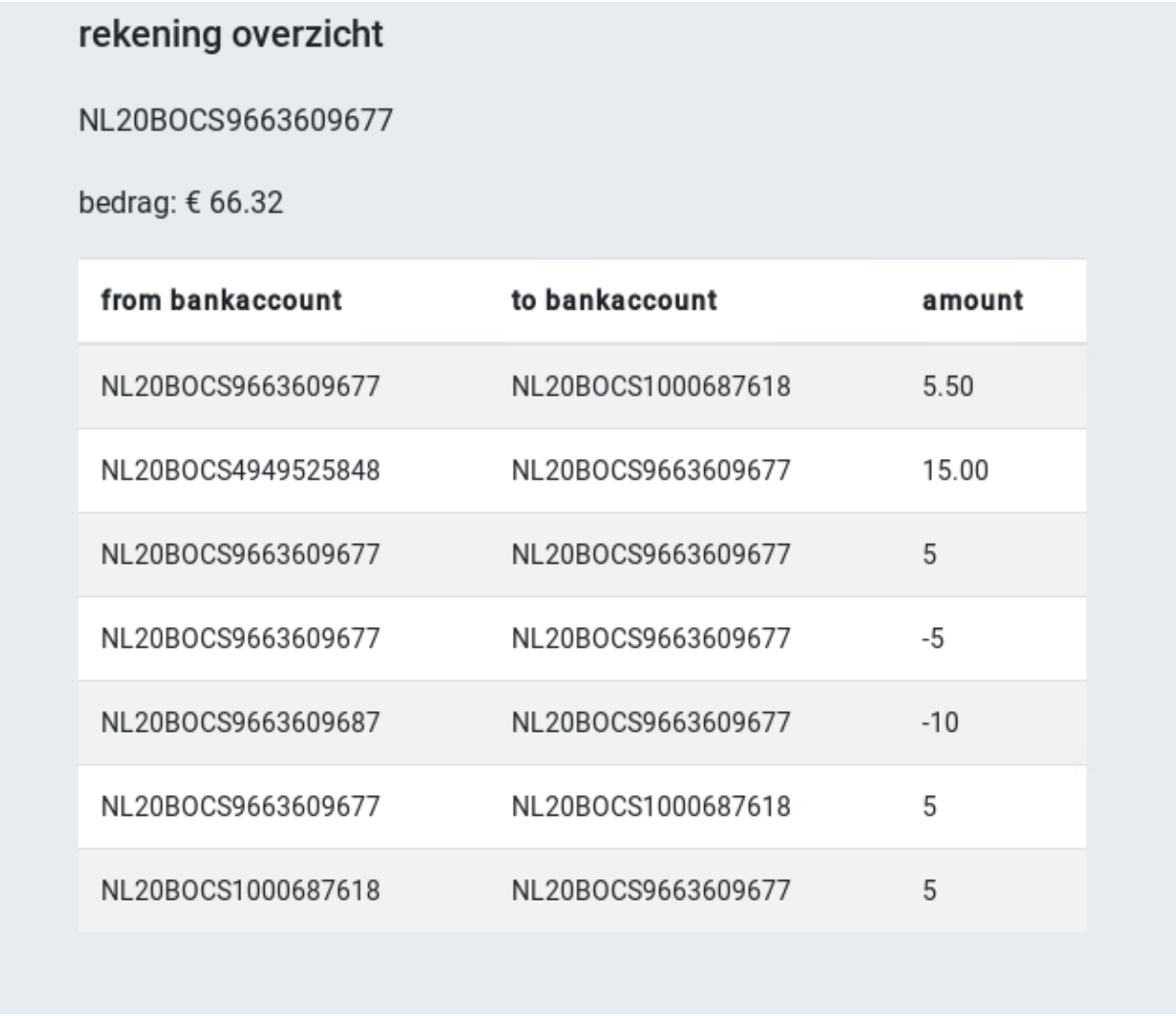
Slechte error afhandeling geeft de hackers meer informatie die hen kunnen helpen een aanval uit te voeren.. The risks are usually low since the information leaked only adds to the overall picture for an attacker, but since it adds value that could be crucial to perform a successful attack it is still mentioned. This is also very inconvenient for the developers or users, since you cannot see what went wrong.

Invoer filtering

CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U/C:L/I:L/A:L/E:P/RC:R/CR:H/IR:H/AR:H

Reproductie

1. Ga naar de pagina om geld over te boeken
2. Vul als bedrag een negatief getal in
3. Selecteer de optie “overmaken”



rekening overzicht

NL20BOCS9663609677

bedrag: € 66.32

from bankaccount	to bankaccount	amount
NL20BOCS9663609677	NL20BOCS1000687618	5.50
NL20BOCS4949525848	NL20BOCS9663609677	15.00
NL20BOCS9663609677	NL20BOCS9663609677	5
NL20BOCS9663609677	NL20BOCS9663609677	-5
NL20BOCS9663609687	NL20BOCS9663609677	-10
NL20BOCS9663609677	NL20BOCS1000687618	5
NL20BOCS1000687618	NL20BOCS9663609677	5

Risico’s

Als er geen invoer filtering is kan je letterlijk alles naar de backend sturen wat je maar wil. In dit geval kan ik negatieve bedragen sturen, waardoor ik in werkelijkheid geld krijg van de andere rekening. Bovendien kan ik geld naar me eigen boeken, of geld van een andermans rekening afhalen.

Preventie

Ter preventie zou elke mogelijke invoer afgevangen moeten worden en zou het invoerveld van “eigen rekening” een placeholder moeten zijn. Hier wil je niet een rekening kunnen invoeren, dan kan ik geld van een andermans rekening halen!