Security specificatie

MLB INC.

Datum: 22-03-2015

Auteurs: Chinji Hoe

Gun Coban

Niek Eichner

Stephan de Jonge

High level specificatie

Over het algemeen zijn we bezig met het realiseren van een pinautomaat. Deze pinautomaat simulatie bestaat uit een lokale client met: paslezer, toetsenbord en printer. Men moet de pinautomaat vanaf daar bedienen. De pinautomaat communiceert via een REST API met een MySQL database server. Ter verduidelijking er wordt één client-computer gebruikt om de pinautomaat haar clientsoftware op te laten draaien en er wordt één server gebruikt om de MySQL database en de REST API te hosten. Bovendien wordt er een microcontroller gebruikt voor de user IO (input/output). De clientsoftware op het lokale apparaat zal de GUI(graphical user interface) en de IO afhandelen zowel de low level(serieel) als high level(REST over HTTPS) IO.

Low level specificatie

Binnen deze sub-sectie bevindt zich een microcontroller die de IO van de RFID reader, keypad en seriële verbinding afhandelt. De RFID reader communiceert via een SPI bus, het keypad via conventionele IO headers en de seriële verbinding gaat over USB(met behulp van een virtuele tty interface voornamelijk COM port 3 of 5 in Microsoft Windows).
Deze lage interfaces geven via de seriële verbinding alle relevante data over aan het hogere niveau van de totale applicatie.

Totale specificatie

De input van de gebruiker zal dus via de randapparatuur van de microcontroller over de seriële verbinding verstuurd worden naar de Java client applicatie. Deze Java applicatie zal de valide input interpreteren en zal wanneer de pincode valide is toegang verschaffen tot de saldo informatie en het zal de mogelijkheid tot opname verschaffen. Het saldo wordt opgehaald via de REST API vanuit de database. Als er geld opgenomen wordt dan zal de Java applicatie het verzoek tot wijzigen van het saldo indienen. Deze applicatie handelt ook het verschaffen van de bon af. De Java applicatie wordt dus de primaire koppeling tussen de user input en de database. De GUI wordt geschreven in javascript en maakt gebruik van Node Webkit (Node JS platform). Alle code voor de microcontroller is in een dialect van C geschreven.

Functionele eisen

Het systeem moet een eindgebruiker zijn of haar pas laten scannen waarna de persoon zijn/haar pincode kan invoeren. Als dit slaagt dan kan de persoon zijn/haar saldo checken en/of geld opnemen. Lukt dit niet dan trekt de automaat één poging van de drie pogingen af. Als er drie pogingen zijn gefaald dan zal het systeem de REST API aanspreken om een blokkade op de pas te zetten. Na het opnemen van geld moet de gebruiker de optie krijgen om een transactie bon uit te printen.

Het is dus van groot belang dat iemand niet zomaar bij zijn of haar gegevens kan. De pincode en de pas dienen samen als autorisatie van de persoon. Daarom is het erg belangrijk dat men hun passen en codes in eigen bezit laten. De 3 pogingen voor de pincode zijn om de integriteit van de autorisatie methode te waarborgen.

Wat de communicatie van de database betreft: De database is alleen aanspreekbaar voor geldige pinautomaten, individuele pinautomaten moeten geen belachelijke hoeveelheden data per seconde

op kunnen vragen. Meer dan bijvoorbeeld 5 opvragingen per seconde is bijvoorbeeld al teveel en zou kunnen betekenen dat een pinautomaat gecompromitteerd is. Ook zal alle data over het HTTPS protocol verstuurd en ontvangen worden om onderschepte data onleesbaar te maken. De pincode wordt niet over het netwerk verstuurd. Alleen wijzigingsverzoeken, saldo informatie en blokkade verzoeken. Eventueel kan er nog meer encryptie toegepast worden op de verbinding. Tot slot tolleren we geen draadloze verbindingen waar ze niet nodig zijn. Denk hierbij aan 802.11(Wi-Fi) de voorkeur gaat uit naar: 802.3(Ethernet), glasvezel, coax en beveiligde mobile breedband verbindingen (GPRS, HDSPA, LTE, LTE+, etc).

De fysieke beveiliging is ook van groot belang, de IO poorten van de pinautomaten moeten fysiek strikt beveiligd zijn om misbruik te kunnen voorkomen. Denk hierbij aan USB poorten en ander soortgelijke seriële en/of parallelle poorten. Elke draadje moet in feite ontoegankelijk zijn voor onbevoegden.

De Java client applicatie is op zich al onveilig. Java heeft een complete historie aan onveiligheden van memory corruption tot buffer overflows. Vandaar dat er niet veel te doen valt aan het beschermen van het Java platform. Echter kunnen we de fysieke toegang tot de machine waar het programma op draait beschermen. Echter is het wel belangrijk om de veiligheid van de geschreven Java code te onderzoeken zodat daar in ieder geval geen relevante beveiligingsrisico's in kunnen zitten. Ook kunnen we de executie ruimte van het Java platform beschermen met behulp van technologieën zoals NX bit. De geheugen ruimte moet eigenlijk ook beschermd worden, niet alleen tegen misbruik maar ook tegen het falen van het geheugen zelf. Het beschermen tegen memory errors is te verwezenlijken met ECC RAM geheugen. Ook zitten we met het probleem dat Windows 7 waar de client applicatie op draait ook nog steeds niet helemaal waterdicht is. Allicht kan een strenge gedrag analyserende watchdog ook geen kwaad. Mag ik hier aan toevoegen dat windows 7 geen aangeraden OS is voor direct IO operations. Wij gebruiken daarom het liefst Windows CE of een UNIX OS met direct IO ondersteuning.

Aannames

Wij gaan er vanuit dat niemand zomaar bij de IO poorten kan. Ook verwachten we dat niet iedereen zomaar toegang tot het netwerk heeft aangezien banken vaak een geconsolideerde mobiele breedband verbinding gebruiken waar men niet zomaar op kan onderzoeken. Uiteraard voor pentests lijkt het ons een goed idee om de beveiliging van het netwerk alsnog te testen. Bovendien verwachten we niet dat iemand de pinpassen zomaar kraakt, uiteraard verwachten we dat de kaarten gekopieerd of uitgelezen kunnen worden maar het ont sleutelen van de gegevens zou bijna onmogelijk moeten zijn.

Doelgroep

Iedereen vanaf 16 jaar met een eigen rekening.

Informatie types

In dit systeem worden de volgende gegevens verwerkt:

- Rekeningnummers
- Pincodes
- Saldo informatie
- ASCII Input
- Decryption key
- Blokkade verzoeken
- Saldo wijzigingsverzoeken

Scope

We zouden graag willen dat de beveiliging van de volgende zaken onderzocht wordt:

- Algemene netwerk (connectie naar/van de server).
- Pinpassen (tot in hoeverre zijn ze exploiteerbaar).
- Encryptie methodes (zijn ze veilig genoeg).
- Encryptie sterkte (Duurt brute-forcing lang genoeg om ongewenst te zijn voor derden).
- Geavanceerde exploitatie (buffer overflows, memory corruption, originele ideeën, etc).
- Microcontroller (tot in hoeverre is de onze exploiteerbaar).

Gelieve niet letten op de volgende zaken:

- Het gekozen client operating system.
- Fysieke beveiliging van de client machine.

Succes!