

TI-Nspire CX II



Achtergrondinformatie

- Een wachtwoord wordt op een computer opgeslagen op een manier die niet leesbaar is voor mensen. Als een hacker je computer scant, zullen ze je platte tekst wachtwoord niet zien. In plaats daarvan zien ze een gecodeerde versie die een hash wordt genoemd.
- Wanneer een account wordt aangemaakt, wordt het platte tekst wachtwoord versleuteld door de hashingfunctie, zoals SHA-256, en opgeslagen in een bestand voor latere vergelijking tijdens de wachtwoordauthenticatie. De hash die door SHA-256 wordt geproduceerd, is 256 bits lang.
- Tijdens de wachtwoordauthenticatie wordt het ingevoerde platte tekst wachtwoord gehasht en vergeleken met de geldige hash die wordt opgehaald van de micro:bit. Als de hashes gelijk zijn, wordt de gebruiker geauthenticeerd en krijgt deze controle over het bijbehorende slot.
- Tijdens de afstandsbediening van het slot wordt de hash verzonden en kan deze kwetsbaar zijn voor afluisteren. Als een hacker de hash van je wachtwoord krijgt, kan hij de hash zoeken in een *rainbow table*. Als deze wordt gevonden, kan het platte tekst wachtwoord dat aan die hash is gekoppeld, worden ontdekt.
- De micro:bit heeft verschillende I/O-pinnen die apparaten zoals een servomotor kunnen besturen. De servo is een speciale motor die niet draait zoals een conventionele motor, maar in plaats daarvan door een boog van 180° beweegt. In deze activiteit beweegt de servo de grendel van de schatkist om de deksel te vergrendelen en ontgrendelen.

Wat is jouw opdracht?

1. Raadpleeg de "Bouw instructies schatkist.pdf" om het elektrisch circuit van de schatkist in elkaar te zetten, de kist in 3D te printen en te monteren.
2. Oefen met het besturen van het servo-circuit.
 - a. Ga naar de pagina met "**servo_test.py**" en voer het programma uit om het servo-besturingscircuit te testen **voordat je de deksel sluit**. Het indrukken van de [enter] toets zal de servo naar de vergrendel- en ontgrendelposities draaien. Een 'X' of een '√' wordt op de micro:bit weergegeven. Druk op de [esc] toets om het programma af te sluiten en koppel de batterij voor de veiligheid los totdat deze weer nodig is. Let op: als de servo zoemt wanneer deze vergrendeld is, raadpleeg dan "Open de schatkist onderdelen en bouw instructies" om de stopposities aan te passen in het "**servo_configuratie.py**" programma.
3. Stel een wachtwoord in op je micro:bit.
 - a. **Kies een van de veul voorkomende wachtwoorden** uit de *rainbow table* van deze activiteit.

Abby	Adam	Beth	Burt	Cora
Abel	Brad	Bret	Carl	Dana

TI-Nspire CX II

- b. Ga naar de pagina met “**stel_wachtwoord_in.py**” en voer het programma uit om het wachtwoord naar keuze in te stellen op je micro:bit.
4. Oefen met het vergrendelen en ontgrendelen van de schatkist
 - a. Ga naar de pagina met “**authenticatie.py**” en voer het programma uit om je wachtwoord en de authenticatieroutine te testen. Dit programma vergelijkt de hash die op de micro:bit is opgeslagen met de hash van het ingevoerde wachtwoord. Als de twee gelijk zijn, krijgt de gebruiker toegang; de servo wordt 90° gedraaid om de grendel van de kist te openen.
 - b. Zodra de kist ontgrendeld is, druk je op [enter] om de servo terug naar de vergrendelde positie te draaien.
 5. Remote login voor de schatkist
 - De **ontvanger**:
 - Ga naar de pagina met “**student_ontvanger.py**” en voer het programma uit. Deel je wachtwoord met de zender. Zij zullen je schatkist op afstand openen.
 - De **zender**
 - Ga naar de pagina met “**student_zender.py**.” Stuur het wachtwoord van de ontvanger, je zal zijn/haar schatkist op afstand ontgrendelen. Voer je programma uit **nadat** de ontvanger en de hacker hun programma's hebben gestart.
 - De **hacker**
 - a) Ga naar de pagina met “**student_hacker_brute.py**” en voer het programma uit. Je zou nu de hash van het wachtwoord van de ontvanger zijn schatkist moeten ontvangen. Gebruik de `brute_force` module en run de functie `brute()` om het wachtwoord corresponderende met de gestolen hash te kraken. Om deze functie te gebruiken, typ **brute(hacked_hash)** in de Python-shell bij de `>>>` prompt op de volgende pagina. Tip: gebruik de [var]-toets op de Nspire om de variabele “`hacked_hash`” te selecteren vanuit het menu.
 - b) Ga naar de pagina met “**student_hacker_rbt.py**” en voer het programma uit. Je zou de verzonden hash van het wachtwoord van de ontvanger zijn schatkist moeten ontvangen. Gebruik de `rainbow table` 'rbt' om het wachtwoord op te zoeken dat overeenkomt met de gestolen hash. Om de tabel te gebruiken, typ je **rbt[hacked_hash]** in de Python-shell bij de `>>>` prompt op de volgende pagina. Tip: gebruik de [var] toets van de Nspire om de variabele “`hacked_hash`” uit een menu te selecteren.
 - Zodra de hacker het wachtwoord van de ontvanger kent, moet de ontvanger zijn programma opnieuw uitvoeren. Vervolgens moet de hacker naar het 'student_zender.py' programma gaan, het uitvoeren en het wachtwoord van de gehackte hash invoeren. Zou de hacker het slot kunnen openen zonder het geheime wachtwoord van de ontvanger te krijgen?
 - Vergelijk het gebruik van de `rainbow table` door de hacker met de brute force-aanval. Welke was de snelste? Welke levert niet altijd een resultaat op? Kun je uitleggen waarom?

TI-Nspire CX II

De code

Zender

```

3.1 4.1 5.1 7 - Cybers...ist RAD 15/15
student_zender.py
# van de ontvanger gebruiken.
kanaal = 1
groep = 1
clear_history()
wachtwoord = input("Geef wachtwoord in: ")
wachtwoord_hash = sha_hash(wachtwoord)
tx(wachtwoord_hash,kanaal,groep)
input("Druk op [enter] om te sluiten")
tx("$Gesloten$",kanaal,groep)
print("De schatkist is vergrendeld!")
    
```

Ontvanger

```

4.1 5.1 5.2 7 - Cybers...ist RAD 11/30
student Ontvanger.py
from microbit_radio import *
from hashing import *
from servo_configuratie import *
# De zender moet het wachtwoord van de
# micro:bit aangesloten op het rekenmachine
# van de ontvanger gebruiken.
kanaal = 1
groep = 1
clear_history()
input("Sluit de kist en druk op [enter] om te vergr
sluit()
    
```

Hacker

```

5.1 5.2 5.3 7 - Cybers...ist RAD 1/14
student_hacker_rbt.py
from microbit_radio import *
from hashing import *
from servo_configuratie import *
from rainbow_table import *

kanaal = 1
groep = 1
clear_history()
print("Man-in-the-middle aanval!")
hacked_hash = rx(kanaal,groep)
print("hash string = {}".format(hacked_hash))
    
```

Extra uitdagingen

- Probeer een andere rol binnen je team.
- Probeer een nieuw wachtwoord dat niet in de regenboogtabel staat.
- Verander de geluiden en LED-displays die worden gebruikt tijdens de authenticatie.
- Verander de servo naar een andere poort; wijzig de 'pin1' code naar het bijbehorende pinnummer.

Samengevat

- De micro:bit heeft invoer- en uitvoerpoorten (I/O) die via software kunnen worden bediend. Wanneer een servo die is aangesloten op een I/O-poort wordt ingesteld op "100", draait de servo tegen de klok in naar de 100° positie. Wanneer de poort wordt ingesteld op "0", draait de servo met de klok mee terug naar de 0° positie.
- Het slot van de kist maakt gebruik van een servo met een aangehechte haakje om het slot te doen sluiten. Wanneer de servo zich in de 5° positie bevindt, grijpt het haakje in het slot, en wanneer de servo in de 100° positie is, komt het haakje los van het slot.
- De I/O-poort die de servo bestuurt is alleen toegankelijk na succesvolle authenticatie.
- Een hash is een unieke 256-bit reeks die een duidelijk platte tekst wachtwoord vertegenwoordigt.
- Authenticatie vergelijkt een opgeslagen geldige wachtwoordhash met een berekende hash van een ingevoerd platte tekst wachtwoord. Als de twee hashes overeenkomen, authentiseert het systeem de gebruiker en verleent toegang.
- Tijdens de remote controle van het slot wordt de wachtwoordhash naar de ontvanger gestuurd, niet het platte tekst wachtwoord.
- Een hacker kan een hash onderscheppen terwijl deze van de zender naar de ontvanger wordt verzonden. De gestolen hash kan vervolgens worden gekraakt met behulp van een *rainbow table* of een brute force-aanval. De *rainbow table* van de hacker kan werken, omdat deze de onderschepte hash bevat. Een brute force-aanval kan echter extreem lange rekentijden vereisen, wat hen kan beletten hun duivelse daad te voltooien.

Tips voor als het misgaat

- Zorg ervoor dat de micro:bit Python-module is geïnstalleerd op het rekenmachine en dat de `ti_runtime.hex` is geïnstalleerd op de micro:bit-kaart. Deze bestanden zijn beschikbaar op education.ti.com/microbit.
- Probeer het wachtwoord opnieuw in te stellen en zorg ervoor dat je 'file written and closed' op het rekenmachine-display krijgt en een '✓' op het micro:bit-display.
- Gebruik de "Bouw instructies schatkist.pdf" om de verbindingen van je schatkist te controleren.
- Zorg ervoor dat de externe USB-batterij is opgeladen en AAN is.
- Controleer of de servo is aangesloten op P1 op het Grove-expansieschild.