Secure Programming Excersises Lesson 5

B.L.Schopman

30 November 2019

Contents

1 Introduction

Dit zijn de oefeningen voor de *vijfde* week van **Secure Programming**

2 Login form

2.1 Het probleem

```
if request.method == 'POST':
    try:
        hash = hashlib.sha256(request.form.get('password').encode()).hexdigest()
        print(hash, file=sys.stderr)
        if hash == pwhash:
            print("equal", file=sys.stderr)
            return render_template('loggedin.html')
        else:
            return render_template('loginform.html', result = 'Invalid login')
```

Het probleem is de hash niet goed is. Volgens Medium [1] moet je volgens deze redenen niet SHA256 gebruiken:

- het is erg traag. Het duurt 50% langer dan met SHA-512 om te berekenen
- Het is erg traag in op een onhandige manier.

  Het maakt je code niet simpeler

  Er zijn veel betere opties zoals SHA-512

  Het is geen goede excuus als je een kortere *hash* wilt gebruiken
- *Truncation* kan ervoor zorgen dat je je kwetsbaar bent voor *length extension attacks* bij SHA-256 en SHA-512. Daarom kan je het beste de volgende algoritmes gebruiken: SHA-384, SHA-512/224, en SHA-512/256

Daarnaast wordt er niet gecontroleerd op de gebruikersnaam. Dat is een probleem , omdat

- je dan alleen het wachtwoord hoeft te weten
- het makkelijker te kraken is als

2.2 Bewijzen

De bewijzen voor een verkeerde hash is deze code, omdat daar in staat dat de hash SHA256 is.

```
hash = hashlib.sha256(request.form.get('password').encode()).hexdigest()
```

Daarnaast is in de code geen gebruikersnaam te vinden

```
if request.method == 'POST':
    try:
        hash = hashlib.sha256(request.form.get('password').encode()).hexdigest()
        print(hash, file=sys.stderr)
        if hash == pwhash:
            print("equal", file=sys.stderr)
            return render_template('loggedin.html')
        else:
            return render_template('loginform.html', result = 'Invalid login')
```

2.3 Redemption

De redemption daar voor is:

- je moet SHA-384, SHA-512/224, en SHA-512/256 gebruiken
- Daarnaast moet de gebruikersnaam worden toegevoegd aan de code

3 Cryptanalysis

3.1 Het probleem

3.2 Bewijzen

3.3 Redemption

References

[1] D.T. Strauss. Stop Using SHA-256. https://medium.com/@davidtstrauss/stop-using-sha-256-6adbb55c608, 2017. [Online; accessed 16-December-2019].