Avanced websecurity - HA4 B2

Nora Wernerson, dat14nwe
Emma Asklund, dat14eas

December 12, 2017

1 Introduction

The security of a web page for giving students grade is not the best it could be. Messages sent to the server needs a signature calculated as $s = HMAC(name||grade, k)$ and truncated to 10 characters. With this knowledge we made a program to find the right signature with a variant of brute-force.

2 Description of the algorithm

As mentioned is the introduction, out program is using brute-force, but testing all combination of possible characters in the signature would take very long time and would not be efficient. Therefor we exploited the fact that when the signature is verified it checks one character at a time and returns 0 when finding the first character that is wrong. This means that if the first character of the signature is wrong, zero is returned quite fast. But if for instance the first five character of the signature is right and the sixth is wrong it would take longer time before zero is returned. With this knowledge, we can analyze the time it takes before zero is returned to find out if each character at a time is correct.

In our program we look at the character in the signature one at the time where each character can be any of the hexadecimal characters 0-f. We starting with the first character, by analyzing how long time the GET request takes we can see for which hexadecimal number the request took the longest time and then we know that this is the right hexadecimal number for the first position of the signature. Now this hexa character is saved. When doing the same procedure for the second position of the signature we first add the correct hexa character for the first position and then try the different hexa character of the second position to find the right one. This procedure is repeated until the entire signature is found. The algorithm in python code can be found in section 3.

2.1 Dependency

This algorithm is depending on the internet connection. When trying our program on for example Enduroam it gets the wrong answer a lot of time, but if we used it at home with a cable connected to the computer, instead of using wireless, it get the right answer 6/10. This flaw is not anything the program is taking in to consideration.

3 Code algorithm

```
for y in range(20):
        bestTime = 0
        for i in range(16):
            temp = signString + intToHex(i)
            payload={'name' : name, 'grade': grade,  'signature':
temp}
            r = requests.get('https://eitn41.eit.lth.se:3119/ha4/
addgrade.php', params=payload, verify = False)
            time = r.elapsed.total_seconds()
            if time > bestTime:
                bestTime = time
                foundRightChar[y] = intToHex(i)
        signString = signString + foundRightChar[y]
    signature = signString
```