UNIVERSITY OF MARYLAND
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ENEE 457
Computer Systems Security
Instructor: Charalampos Papamanthou

Homework 5

Out: 11/18/18 Due: 11/28/18

Instructions

1. Strictly adhere to the University of Maryland Code of Academic Integrity.
2. **Type** and submit your solutions as a pdf document at Canvas. Include your full name in the solutions document. Name the solutions document as x-hw5.pdf, where x is your last name.
3. No extensions will be given. Everything will be due at 11.59pm on 12/09/16.

Problem 1 Computer Networking (25 points)

A switch is a device that allows multiple computers to connect to each other in such a way that each computer only receives data intended specifically for it. Suppose we have a switch that works as follows. When a computer is connected to one of its ports, it adds a tuple (x, y) to a data structure called the Server Source Address Table, where x is the MAC address of the certain computer and y is the certain port that is assigned to the computer. This new entry will overwrite any other entry with the same MAC address. Assume an attacker can choose her own MAC address. Describe an attack that the network connected by this switch is vulnerable to.

Problem 2 Domain Name System (DNS) (25 points)

1. Describe the main purpose of DNS.
2. Suppose the transaction ID for DNS queries can take values from 1 to $65,536$ and is randomly chosen for each DNS request. If an attacker sends $1,024$ false replies per request, how many requests should he trigger to compromise the DNS cache of the victim with probability 99%?

Problem 3 Secure Storage (Merkle Hash Trees) (25 points)

In this assignment you are going to write a program (in the programming language of your preference) that implements Merkle hash trees (originally introduced by Ralph Merkle in this paper), as we discussed in class. Suppose n is a power of 2. Your program should implement the following functions:

1. $T \leftarrow \mathsf{setup}(f_1, f_2, \ldots, f_n)$. On input n files $f_1, f_2, \ldots, f_n$ (1 KB each) it outputs the Merkle hash tree T. Recall that a Merkle hash tree is a binary tree such that for each internal node v (that has children u and w) we store a hash $h_v = \mathsf{HASH}(h_u, h_w)$. If v is a leaf corresponding to file f_i we set $h_v = f_i$. You can use an existing implementation of SHA-2 for the implementation of HASH.

2. $\pi_i \leftarrow \mathsf{prove}(i, T)$. On input an index i and the Merkle hash tree T, it outputs a proof π_i for file f_i. Recall the proof is a collection of hashes along the path from i to the root r.
3. $\{\texttt{ACCEPT}, \texttt{REJECT}\} \leftarrow \mathsf{verify}(f_i, \pi_i, h_r)$. On input a file f_i, a proof π_i and the hash of the root h_r, it outputs either `ACCEPT` or `REJECT`.
4. $T' \leftarrow \mathsf{update}(f_i', T)$. On input an updated file f_i', it outputs the updated Merkle hash tree T'.

Run experiments and plot the time it takes to execute the algorithms $\mathsf{setup}()$, $\mathsf{prove}()$, $\mathsf{verify}()$ and $\mathsf{update}()$ for $n = 2^{10}, 2^{11}, 2^{12}, \ldots, 2^{20}$.

Problem 4 SQL Injection Attack (25 points)

1. Explain the bug in this PHP code. How would you exploit it? Write what you would need to do to delete all of the tables in the database.

   ```
   $query = "SELECT name FROM users WHERE uid = $UID";
   // Then execute the query.
   ```

   (Here, $UID represents a URL parameter named UID supplied in the HTTP request. The actual representation of such a value in PHP is a bit messier than we have shown here.)
2. What is the best way to fix this bug?