CS 361S - Network Security and Privacy
Spring 2014

Homework #1

Due: 11am CST (in class), February 11, 2014

YOUR NAME: _____

Collaboration policy

No collaboration is permitted on this assignment. Any cheating (*e.g.*, submitting another person's work as your own, or permitting your work to be copied) will automatically result in a failing grade. The Deparment of Computer Science code of conduct can be found at `http://www.cs.utexas.edu/undergraduate-program/code-conduct`.

Late submission policy

This homework is due at the **beginning of class** on **February 11**. All late submissions will be subject to the following policy.

You start the semester with a credit of 3 late days. For the purpose of counting late days, a "day" is 24 hours starting at 11am on the assignment's due date. Partial days are rounded up to the next full day. You are free to divide your late days among the take-home assignments (3 homeworks and 2 projects) any way you want: submit three assignments 1 day late, submit one assignment 3 days late, *etc.* After your 3 days are used up, no late submissions will be accepted and you will automatically receive 0 points for each late assignment.

You may submit late assignments to Vitaly Shmatikov or Oliver Jensen. **If you are submitting late, please indicate how many late days you are using.**

Write the number of late days you are using: _____

Homework #1: A Trip to Molvanîa (50 points)

Molvanîa is a small, land-locked republic in Eastern Europe famous for its phishers, spamlords, botmasters—and computer security researchers. It also produces 83% of the world's b33tr00t. Most people get to Molvanîa either by air or by accident, but in this homework, we travel there virtually.

Problem 1 (5 points)

Molvanîan PCs still run Windows 98. Therefore, passwords in Molvanîa are hashed using Microsoft's LAN Manager (LM) hash, which works as follows:

- The password is converted into upper case, null-padded to 14 characters, and split into two 7-character halves.
- Each half is separately converted into a DES key. This key is used to encrypt the ASCII string "KGS!@#$", producing an 8-byte value.
- The two 8-byte values are concatenated, resulting in a 16-byte hash.

Suppose the attacker obtains a file with n hashed passwords. How much work would he need to do to crack these passwords by brute-force search? Show your calculations.

Problem 2 (5 points)

MMACs (Molvanîan Message Authentication Codes) are intended to provide authentication and integrity for email messages between Molvanîan diplomatic missions. All missions share the secret key K. Each message M sent by one mission to another is accompanied by a MMAC, which is constructed as $MH(K, M)$.

MH is a hash function with 320-bit output invented by Molvanîan cryptologists. They took SHA-1 (which is assumed to be one-way and collision-resistant—at least for the purposes of this problem) and used it as a building block to create MH. They even proved that MH, too, is one-way, collision-resistant, and hides the key.

As it turns out, MMAC is completely broken. Someone eavesdropping on a single MMAC-protected message call can gather enough information to forge valid MMACs in the future,

that is, to send any message they want accompanied by a forged MMAC that will pass verification by any Molvanîan mission.

How is MH constructed? (Important: your construction must be one-way, collision-resistant, hide the key, and still make the above scheme vulnerable to forging.)

Problem 3 (5 points)

Molvanîan Telecom is selling a fancy smartphone model called jPhone. Each jPhone stores a long, randomly generated secret value. The phone service provider keeps all secrets, together with the corresponding cell phone numbers, in its database.

When a jPhone user wishes to buy a new ringtone, the jPhone transmits its phone number followed by the secret (in the clear) to the ringtone server. The server checks in its database whether the secret corresponds to the provided phone number and, if it does, downloads the ringtone to the phone and bills the account of the phone's owner.

This design is vulnerable to a **cloning** attack. Someone eavesdropping on a jPhone transmission can easily intercept a (*phone number-secret*) pair. He can then hack his own jPhone's transmission software to use the intercepted pair, enabling him to download ringtones which are billed to the victim's account.

Design an authentication scheme for jPhone based on a cryptographically secure hash function that prevents passive attackers from exploiting eavesdropped messages between the jPhone and the ringtone server.

Problem 4

To access his account online at the Bank of Molvanîa, a user must install a client program on his Windows 98 PC. The user's password is set up when the account is created and stored on the bank's server.

When the user logs in, the client prompts him for his password p, computes HMAC[1] of p and current time t rounded to a minute, and sends the result to the server. The server recomputes HMAC using p and its own time. If the resulting value is equal to the value received from the client, the server allows access.

Problem 4a (5 points)

Unfortunately, Windows 98 PCs crash a lot and when they crash, the clock resets to midnight, January 1, 1980. Subsequently, the client's timestamps are all wrong and authentication fails.

The Bank of Molvanîa hired George Spelvin, Molvanîa's premier security expert, to fix the problem. Spelvin suggested the following clever modification to this authentication scheme. Instead of the client generating the timestamp t, the server sends t to the client as the challenge. The client's response is computed as before.

Does this modification have any security consequences? (Hint: consider an active man-in-the-middle attacker who controls the network.)

Problem 4b (5 points)

Modify Spelvin's scheme so that it is secure against an active man-in-the-middle attacker, but still does not require the client to generate its own timestamps. Your solution should use only timestamps, passwords, and HMAC.

Problem 5

The Bank of Molvanîa adopted the following defense against phishing. The first time a user comes to the bank's website, she enters her username and password as usual, and is given a choice between several pictures. The association between the username and the chosen

[1]This HMAC uses SHA-1, not LAN Manager hash.

picture is stored in the bank's database. In all subsequent sessions, the user types in her username and expects to be shown a picture. Unless she sees the picture she chose during her first session, she does not type in her password. This helps users avoid giving their passwords to fake websites.

Problem 5a (4 points)

Describe a man-in-the-middle attack that allows a fake website to show the user her chosen picture. (Assume that this is not the user's first session, *i.e.*, she has already chosen the picture.)

Problem 5b (4 points)

Design a cookie-based defense for this anti-phishing scheme that prevents the man-in-the-middle attack you discovered in Problem 5a.

Problem 5c (4 points)

If every user of the bank's website has a cookie identifying her to the bank, does this eliminate the need for passwords? Explain.

Problem 6 (5 points)

`3m41l.mi` is Molvanîa's Web-based email system. After the user authenticates to the system's Web server, the server stores a cookie (called `SessionCookie`) in the user's browser so that all subsequent requests from this user do not require authentication.

Email messages are displayed in the user's Web browser using the following HTML template:

```
<HTML>
<BODY>
--- Headers appear here ---
<DIV ID="msg">
--- Email message is displayed here ---
</DIV>
</BODY>
</HTML>
```

Give an example of an email message that you could send to a user of this Web-based email system and that would allow you to read all of that user's email.

Problem 7

Web browsers' same origin policy (SOP) for DOM access is based on the (protocol, host, port) triple. The SOP for sending cookies to websites involves domain and path, but cookies marked "secure" are sent over HTTPS only.

Richer Molvanîans don't use Windows 98 PCs. They prefer ancient Apple PowerBooks with old versions of the Safari browser. In Safari before version 3.0, the SOP for DOM access was defined using host and port only (*i.e.*, it did not include the protocol).

Problem 7a (4 points)

Explain how a network attacker could steal secure `google.com` cookies. (Hint: consider a Molvanîan user who logs into Gmail using HTTPS, but then receives a `google.com` page served over HTTP.)

Problem 7b (4 points)

Under the same assumptions, is it possible for a Web attacker to steal secure `google.com` cookies? Describe an attack or explain why you believe none exists. Recall that a Web attacker can set up a malicious website (at some domain other than `google.com`) and trick the user into visiting this site, but cannot intercept or forge network packets.