

CS 378 - Network Security and Privacy

Spring 2012

Homework #1

Due: 2pm CST (in class), February 16, 2012

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 Computer Sciences department 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 16**. 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 2pm 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 (CSA 1.114—slide under the door if the office is locked). **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 Molvania (50 points)

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

Problem 1 (6 points)

SHA-1 looks way too complicated. Molvanian Institute for Advanced Cryptology—which, incidentally, has old editions of Knuth's books in its library—developed a much simpler hash function. Here is its pseudocode:

```
unsigned int hash(bytearray[] msg)
{
    unsigned int hash = 0xDECAFBAD;
    for(i = 0; i < msg.length(); i++)
    {
        hash = ((hash << 5) XOR (hash >> 27)) XOR msg[i];
    }
    return (hash BITWISE-AND 0x7FFFFFFF);
}
```

Is this a good cryptographic hash function? Explain.

Problem 2 (6 points)

MMACs (Molvanian Message Authentication Codes) are intended to provide authentication and integrity for email messages between Molvanian 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 Molvanian 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 Molvanian 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 (6 points)

Molvanian mainframes use an ancient operating system called T3N3X. In T3N3X, a user can store his data on a sequence of virtual memory pages, but if a process tries to read across a page and the second page is not in physical memory, a page fault occurs. Recovering from a page fault requires swapping a page into memory and thus takes much longer than reading a page that is already in physical memory.

T3N3X provides password protection on individual directories. The password comparison algorithm works as follows (`passwordArgument` is the string provided by the user trying to gain access; he can store it anywhere in memory).

```
for (i = 0; i <= strlen (directoryPassword); i++)
    if (directoryPassword[i] != passwordArgument[i])
        return DENY_ACCESS;
}
return GRANT_ACCESS;
```

Explain how one can exploit this system to guess passwords in time linear in password length.

Problem 4 (6 points)

Molvaniã Telecom is (finally!) 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 5

The Bank of Molvaniã 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 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 (6 points)

3m411.com is Molvania'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, while the SOP for sending cookies to websites involves domain and path. Cookies marked "secure" are sent over HTTPS only. In modern browsers, reading `document.cookie` in an HTTP context does not reveal secure cookies.

Molvania is Eastern Europe's biggest market for old Apple computers and Molvanians exclusively use old Safari browsers. In Safari before version 3.0, the SOP for DOM access is defined using host and port only (*i.e.*, it does not include the protocol).

Problem 7a (4 points)

Explain how a network attacker could steal secure `3m411.com` cookies. (Hint: consider a Molvanian user who logs into her mailbox using HTTPS, but then receives a `3m411.com` page served over HTTP.)

Problem 7b (4 points)

Under the same assumptions, is it possible for a Web attacker to steal secure `3m411.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 `3m411.com`) and trick the user into visiting this site, but cannot intercept or forge network packets.