Family Name:.........................

First Name:...........................

Section:...............................

Security Protocols and Applications
(Part 1/2: Javascript Worms)

Final Exam

July 1st, 2008

Duration: 2 hours (for the 2 parts)

This document consists of 7 pages.

Instructions

Electronic devices are *not* allowed.

All printed documents are permitted.

Answers must be written on the exercises sheet.

This exam contains 2 *independent* parts.

This part contains 4 exercises.

Answers can be either in French or English.

Questions of any kind will certainly *not* be answered. Potential errors in these sheets are part of the exam.

You have to put your full name on the first page of each document and have all pages *stapled*.

1 Same Origin Policy

1. On a web page located at the following URL

   `https://webmail.epfl.ch/horde/imp/mailbox.php?mailbox=INBOX`

   which of the following instruction will be blocked by the SOP?

   a) `<script src="https://www.hacker.com/hack.js">`

   b)
   ```
   <script>
   req = new XMLHttpRequest();
   req.open("GET","https://www.hacker.com/hack.js",false);
   </script>
   ```

   Circle the one(s) that will be blocked.

2. Can you explain the motivation of this behaviour?

2 The Journal du Net

3. The “journal du net” has a search page that is vulnerable to cross site scripting. Typing the word “hello” into the search form generates the following URL:

   ```
   http://telechargement.journaldunet.com/cgi/rech.php?key=hello
   ```

   the word “hello” appears in the following part of the resulting web page:

   ```
   JE CHERCHE UN LOGICIEL<input name="key" value="Hello" type="text">
   ```

   What do you have to type in the search form to make the following code appear in the page: `<H1>hacked!</H1>`

4. Interestingly, classical cross site scripting attacks that include java script (e.g. `<script>alert("hacked!")</script>` do not work. However, replacing `<H1>hacked!</H1>` from the previous question with the following code, yields pop-up with the message “100”:

   ```
   <img src=0 onerror=alert(100)>
   ```

   Can you explain this behaviour?

3 The Nasdaq

The web site of the Nasdaq suffers from cross site scripting on the pages that display quotes. The vulnerable URL is the following:

```
http://quotes.nasdaq.com/quote.dll?mode=stock&symbol=test&page=quick&selected=MSFT
```

5. The string that is assigned to the parameter `symbol` ("test" in the above example) appears as text in different places of the resulting web page. It is thus easy to insert any HTML code into the page. However all characters are changed to upper case (in the above example, the page displays "TEST" and not "test"). Javascript is case sensitive and all interesting commands include lowercase letter. Can you still find a way to insert working javacode into the page?

6. Explain how the developers at Nasdaq could have successfully prevented cross site scripting attacks.

4 Writing a Worm

So far we have only looked at how to insert javascript into a web page. This is only one part of a worm. Imagine that you want to design a fully functional worm that propagates automatically. The target system is a web based e-mail system like `webmail.epfl.ch` or `owa.epfl.ch`. You want to exploit the following vulnerability: when a list of messages is displayed, for example when the user is looking at his inbox, any code that is included in the subjects of the messages appears as-is in the web page. For example, if the subject of a mail included `<b>test</b>` the word test would appear in bold in the list of message.

7. Without writing any actual javascript code explain

   a) what functions your worm would have to include, and how these functions would be implemented,

   b) how exactly the worm would propagate,

   c) whether or not your worm would be impacted by the same origin policy,

   d) and finally, if the subject of a message could only hold a limited amount of code, how you could work around this limitation.

**Any attempt to look at
the content of these pages
before the signal
will be severly punished.**

Please be patient.