1. We know of many situations where cross-domain queries are allowed per domain/query. What can go wrong when it is not restricted? Please construct a scenario where cross-domain data access occurs (that would otherwise be denied by crossdomain checks) and some information leak happens.

2. One of the solutions against CSRF was to use the Referer HTTP header to validate form submissions. On the other hand, we know that HTTP headers can be faked by the client. How come Referer checking can still be useful and in which situations could it break?

3. Find all potential vulnerabilities in this C function:

```c
int url_encode_and_validate(void) {
    char input[100], encoded[200];
    int ret = 0;
    int i, j;

    gets(input);
    if (strlen(input) > 100) {
        return 0;
    }
    j = 0;
    for (i=0; i<strlen(input); i++) {
        encoded[j++] = '%';
        encoded[j++] = hex_char((input[i] >> 4) & 15);
        encoded[j++] = hex_char(input[i] & 15);
    }
    encoded[j] = 0;
    if (validate(encoded)) ret = 1;
    return ret;
}
```

4. Find all potential vulnerabilities in this JSP snippet:

```jsp
<% String title = request.getParameter("title");
    // regexp replace
    title = title.replace('<script>','<div>');
    title = title.replace('<object>','<div>');
    title = title.replace('<applet>','<div>');
    title = title.replace('<embed>','<div>'); %>
<p>Title <b>'<%= title %>'</b> added to your shopping cart.</p>
```

5. What possible vulnerabilities could the following URL exploit?

```plaintext
http://a.com/a.php/b.html?sessid=10203040506070809&user='id'&pass=.&name=x'union%20select%20'z&message=%3ci%4dg+s%52c%3d%22j%61v%61S%43r%69pt%3a%61l%65rt(%27%2c%27)%3b%22%3e
```

Send response by e-mail to mroos@ut.ee no later than 12.00. You can answer either in English or in Estonian.