Secure Programming Techniques — exam 31.05.2019

1. How does a setuid program know the actual user who is running it?
2. Can we use CORS headers to fight CSRF? If yes, how? If no, why?
3. Find all potential vulnerabilities in this C function:

```
void store_socket_data_to_file(int fd)
{
    int i, len, f;
    char *buf;

    /* Read length of filename */
    if (read(fd, &len, 4) < 0) return;
    /* Allocate room for string */
    buf = malloc(len);
    /* Read filename */
    if (read(fd, buf, len) < 0) return;

    /* Open file for writing */
    if (f = open(buf, O_RDWR | O_CREAT) < 0) return;
    /* Read from socket into the file, in chunks */
    while (i = read(fd, buf, 1024) > 0 ) {
        write(f, buf, 1024);
    }
    close (f);
}
```

4. Find all potential vulnerabilities in this PHP snippet:

```
<?php
  $sql="SELECT item,price FROM store WHERE ID=".
    addslashes($_REQUEST['id']);
  $result = mysql_query($sql) || die($sql);
  while($row = mysql_fetch_array($result)) {
    echo $row['item'] . ", " . row['price'] . "<br>";
  }
  $fn = escapeshellcmd($_REQUEST['user']);
  $file = fopen("/var/www/avatars/".$fn, "r");
  fpassthru($file);
  fclose($file);

  echo "You are using " . $_SERVER['HTTP_USER_AGENT'] . "\n\n";
?>
```

5. What possible vulnerabilities could the following URL exploit?

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