Informe Código Dañino
CCN-CERT ID-12/21

RansomEXX_Linux ransomware

Septiembre 2021

Edita:

GOBIERNO DE ESPAÑA MINISTERIO DE DEFENSA

CCN centro criptológico nacional

Fecha de Edición: septiembre de 2021

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

ÍNDICE

1. SOBRE CCN-CERT

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

El presente documento recoge el análisis de la muestra de código dañino identificada por la firma MD5 **22f2ed67b2aec55d54c9864f98e0ab85** compilada para plataformas Linux (x64). El binario analizado pertenece a la familia de *ransomware* apodada como EXX o RansomEXX, también conocido como Defray777 o RansomX. Este *ransomware*, vinculado con el actor GOLD DUPONT[1], ha estado involucrado en múltiples incidentes desde 2018[2] y ha afectado a organismos gubernamentales y empresas críticas.

En mayo de 2020 el sistema judicial[3] así como el Departamento de Transporte del estado de Texas (TxDOT)[4] se vio afectado por este *ransomware*. La multinacional japonesa Konica Minolta[5] así como el sistema judicial de Brasil[6] fueron también objetivo de RansomEXX en julio y noviembre, respectivamente, de ese mismo año. La empresa brasileña Embraer, considerada como uno de los mayores fabricantes de aviones civiles de la actualidad, no solo se vio afectada[7] por RansomEXX a finales de 2020, sino que sus datos fueron filtrados después de que el fabricante de aviones se negara a ceder ante la, cada vez más común, doble extorsión[8]. En el último año, determinados grupos de atacantes han utilizado, como vía de entrada, ciertas vulnerabilidades en productos VMWare ESXi para cifrar[9] sus discos duros virtuales mediante EXX.

RansomEXX se caracteriza por utilizar la librería *open-source mbed TLS*[10], tanto en su variante Windows como en Linux, para llevar a cabo las labores de cifrado. Dicha librería es empleada para generar claves AES con las que cifrar los ficheros de la víctima usando el cifrado por bloques AES-256 en modo ECB (*Electronic CodeBook*). Posteriormente, cada clave AES es cifrada con una clave pública RSA-4096, embebida en el cuerpo del binario, que será adjuntada a cada archivo cifrado (cuya extensión, para la muestra actual, es *".31gs1-"* seguido de 8 caracteres en hexadecimal). A diferencia de la variante EXX para entornos Windows[11], la muestra analizada no implementa funcionalidades adicionales más allá que el propio cifrado de los ficheros (no finaliza el espacio libre en disco, no emplea técnicas anti-análisis y no finaliza servicios y procesos).

3. DETALLES GENERALES

El fichero analizado se corresponde con una muestra de *ransomware* para entornos Linux (ELF64) de la familia RansomEXX cuya firma se muestra a continuación:

MD5	22f2ed67b2aec55d54c9864f98e0ab85
SHA1	3f78e2c6fa101d03b338ea9a750c0b4ccfe8bd95
SHA256	63de4a85b9d2cc3670987b284614ed54f3ac6c7d2e09c0fff7b0747d79200281

Cabe destacar que este *sample* había sido remitido, por primera vez, a las plataformas de análisis de malware[12] el 23 de junio de 2021.

El binario presenta un tamaño de 212.264 bytes y ha sido compilado (*dynamically linked*) con GCC para arquitecturas de 64 bits. La siguiente imagen muestra su *ELF Header* así como algunas de las propiedades estáticas del mismo.

```
┌──(root💀CCNLAB)-[~]
└─# file exx
exx: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=
ed2d11c4f89d51d0c86833f66a866b12929ab5e5, for GNU/Linux 3.2.0, not stripped
┌──(root💀CCNLAB)-[~]
└─# ldd exx
        linux-vdso.so.1 (0x00007ffc8056c000)
        libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007f3840726000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f3840561000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f3840790000)
┌──(root💀CCNLAB)-[~]
└─# readelf -h exx
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              DYN (Shared object file)
  Machine:                           Advanced Micro Devices X86-64
  Version:                           0x1
  Entry point address:               0x3320
  Start of program headers:          64 (bytes into file)
  Start of section headers:          210280 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         11
  Size of section headers:           64 (bytes)
  Number of section headers:         31
  Section header string table index: 30
┌──(root💀CCNLAB)-[~]
└─# objdump -s --section .comment exx

exx:     file format elf64-x86-64

Contents of section .comment:
 0000 4743433a 20284465 6269616e 20392e33  GCC: (Debian 9.3
 0010 2e302d31 38292039 2e332e30 00        .0-18) 9.3.0.
┌──(root💀CCNLAB)-[~]
└─#
```

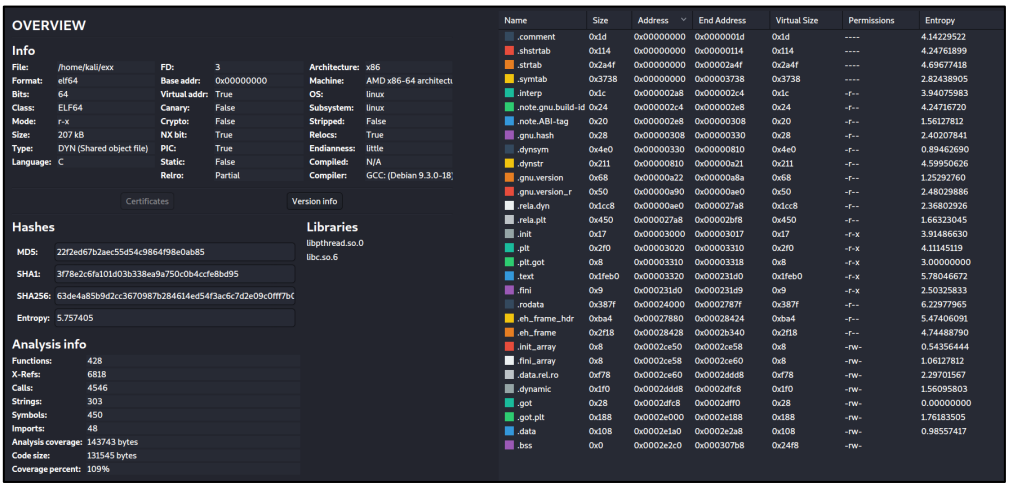


Figura 1. Información estática ELF64

El código dañino ha sido compilado estáticamente con la librería *open-source mbed TLS* y, a diferencia de su versión para Windows, la información del binario que genera el compilador no ha sido eliminada (*stripped)* facilitando en gran medida la ingeniería inversa. Entre la información de *debug* se identifican nombres de funciones, nombres de ficheros, así como variables globales que permiten atribuir fácilmente la muestra a la familia de *ransomware* EXX.

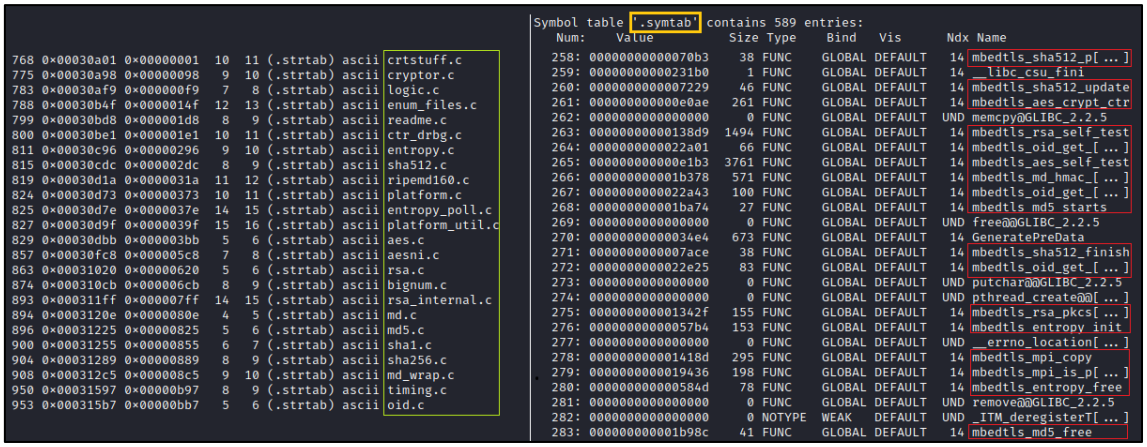


Figura 2. .Symtab (Símbolos de mbed TLS)

La entropía reflejada por los *loadable segments* muestra un binario que no ha sido empaquetado/cifrado.

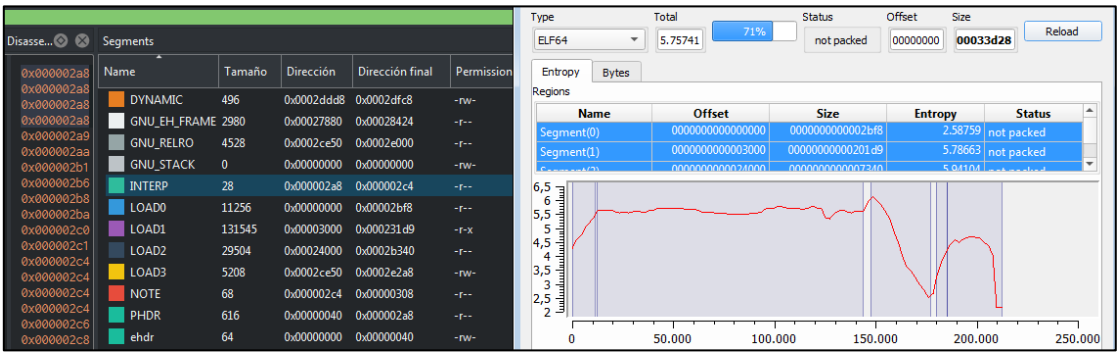


Figura 3. Segmentos / Entropía (LOAD)

El código dañino debe ejecutarse con uno o varios argumentos indicando los directorios que desean cifrarse. Los ficheros resultantes serán renombrados siguiendo el formato "*.31gs1-<8_hex_string>*" donde *<8_hex_string>* indica un valor aleatorio de 8 caracteres en hexadecimal.

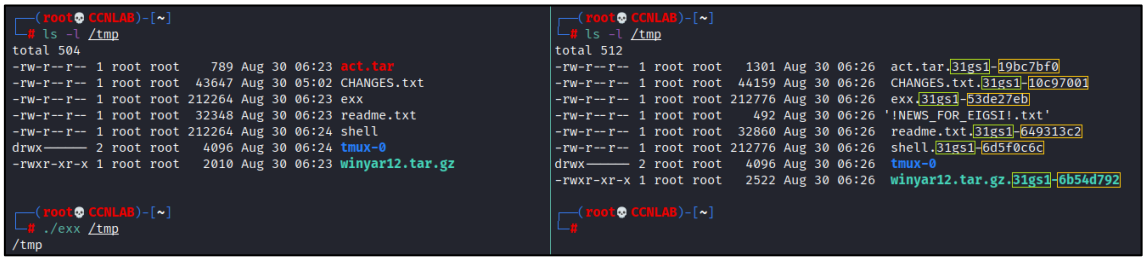


Figura 4. Ficheros cifrados

En cada directorio afectado se creará un fichero con el nombre "*!NEWS_FOR_EIGSI!.txt*" en donde se indican las instrucciones para recuperar los ficheros cifrados. Cabe destacar que el *sample* actual únicamente muestra de manera legible algunas de las cadenas de dicho *txt*. El cuerpo del mensaje, donde se detallan las instrucciones de recuperación, está en su mayoría codificado posiblemente por algún error en el *builder* encargado de gestionar dicha lógica. En la siguiente imagen se muestra, a la izquierda, el mensaje de texto generado por la muestra actual y, en la derecha, el generado por otro *sample*[13], de características muy similares (misma *public key*).

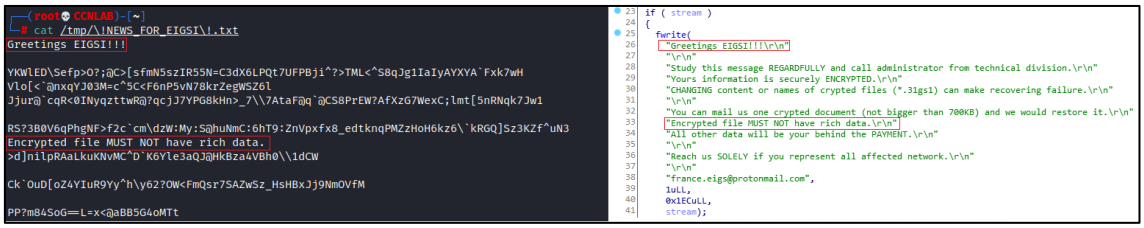


Figura 5. Fichero *!NEWS_FOR_EIGSI!.txt*

Si las víctimas pagan el rescate[14], recibirán una herramienta (*decryptor64*) con la clave privada correspondiente con la que podrán recuperar los ficheros cifrados.

Figura 6. Decryptor64. Fuente: BleedPingComputer

El grupo cibercriminal mantiene, en uno de los dominios *onion* (*http://*rnsm777cdsjrsdlbs4v5qoeppu3px6sb2igmh53jzrx7ipcrbjz5b2ad*[.]onion*), un portal Web donde publican de forma regular los *leaks* vinculados con algunas de las víctimas que han rechazado el pago del rescate. A fecha de realización del presente análisis, los últimos *leaks* publicados estarían relacionados con *Gigabyte Technology* y *American Megatrends International*[15].

4. CARACTERÍSTICAS TÉCNICAS

El binario dañino acepta como parámetros de entrada cada uno de los directorios que desean cifrarse (de forma recursiva). El código no implementa técnicas antianálisis, no trata de finalizar procesos/servicios y no establece comunicaciones con ningún servidor de control. Asimismo, no sobrescribe el *free space* tras finalizar el cifrado tal como ejecuta la variante para plataformas Windows.

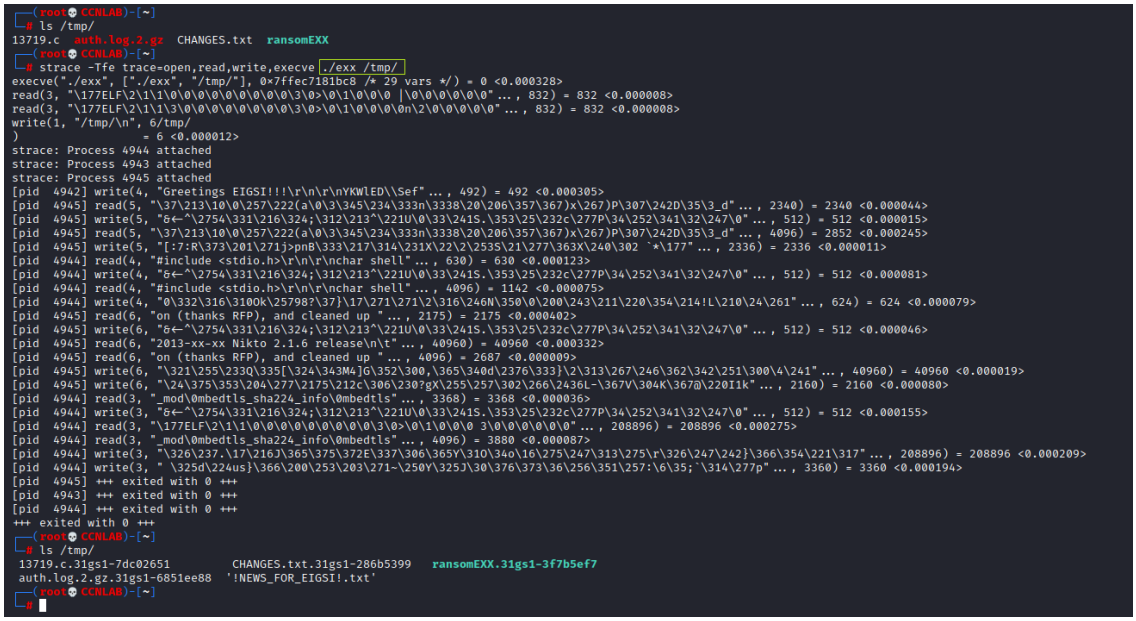


Figura 7. Ejecución RansomEXX

Para ejecutar dichas acciones el código comienza creando un hilo que invocará, en un bucle infinito, a la función *GeneratePreData()* con el objetivo de generar claves AES aleatorias cada 180.000 microsegundos. Cada clave AES será, posteriormente, cifrada con una *public key* RSA embebida en el cuerpo del binario (más información en el punto 5).

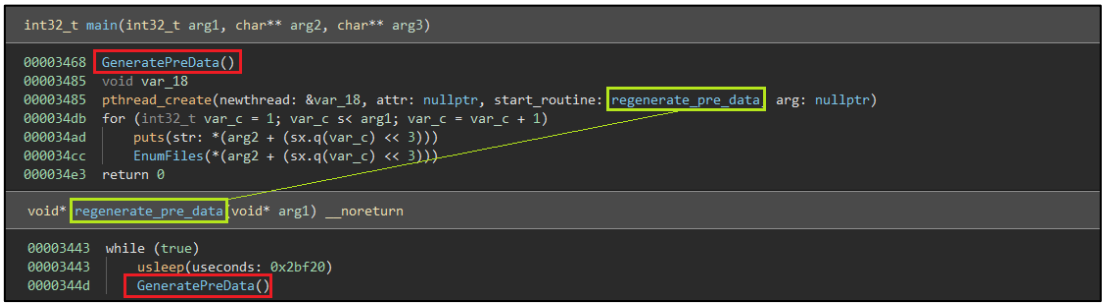


Figura 8. Función *GeneratePreData()*

El espécimen recupera el número de argumentos recibidos (los directorios que se desean cifrar) y, por cada uno, se invocará la función *EnumFiles().* Esta función llamará a *init_workers* en donde se creará un nuevo hilo para ejecutar la función *encrypt_worker* desde donde comenzará el cifrado de los ficheros. La siguiente imagen recoge el *path de* ejecución previamente descrito.

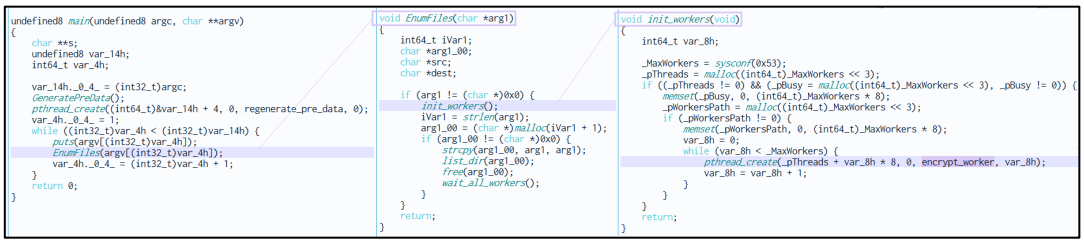


Figura 9. *EnumFiles / init_workers / encrypt_worker*

Desde *EnumFiles* también se invocará la función *list_dir* encargada de crear el fichero "*!NEWS_FOR_EIGSI!.txt*" en cada directorio afectado por medio de la función *ReadMeStoreForDir*.

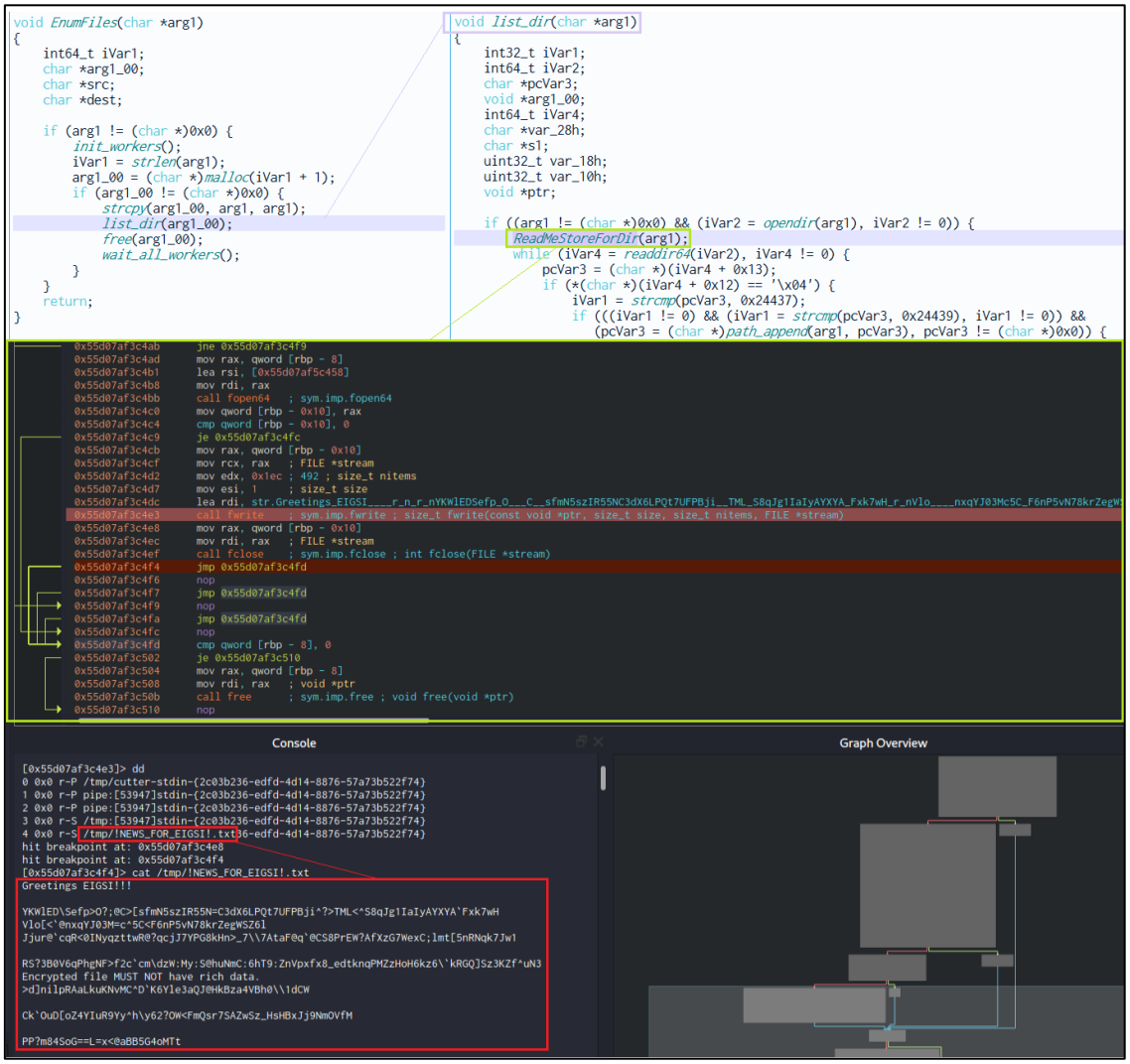


Figura 10. Función *ReadMeStoreForDir*

5. CIFRADO

El binario ha sido compilado estáticamente con la librería *mbed TLS*, una librería *open-source*[¡Error! Marcador no definido.] que implementa funcionalidades criptográficas (manipulación de certificados X.509, protocolos SSL/TLS y DTLS, etc.). A diferencia de la versión para Windows, el binario no ha sido *stripped* facilitando de este modo la ingeniería inversa gracias a los símbolos disponibles en su *symtab*.

Desde la función *GeneratePreData* se hace uso de *mbed TLS* para generar claves AES por medio de la función *mbedtls_crt_drbg_random,* encargada de generar 32 bytes aleatorios.

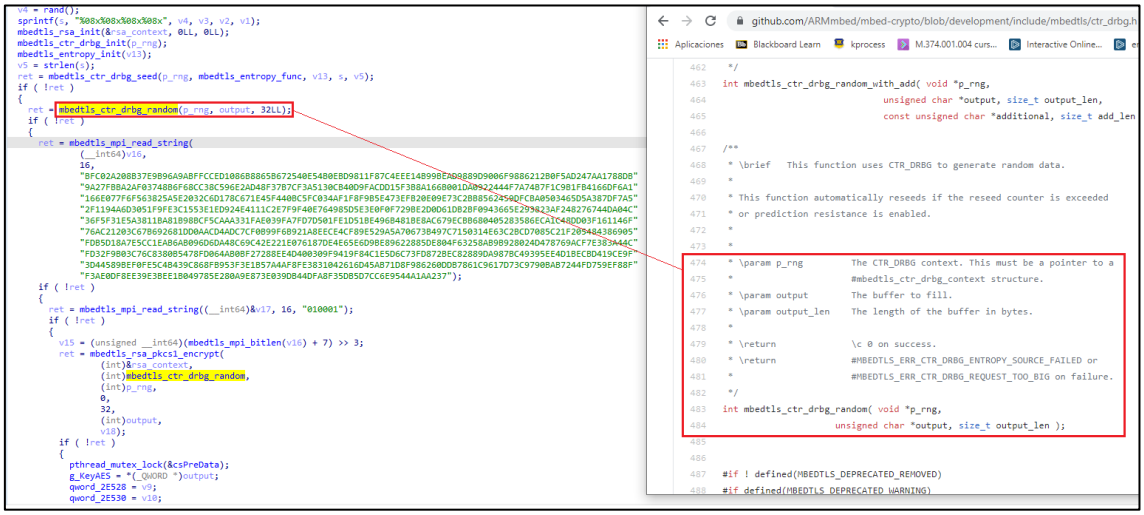
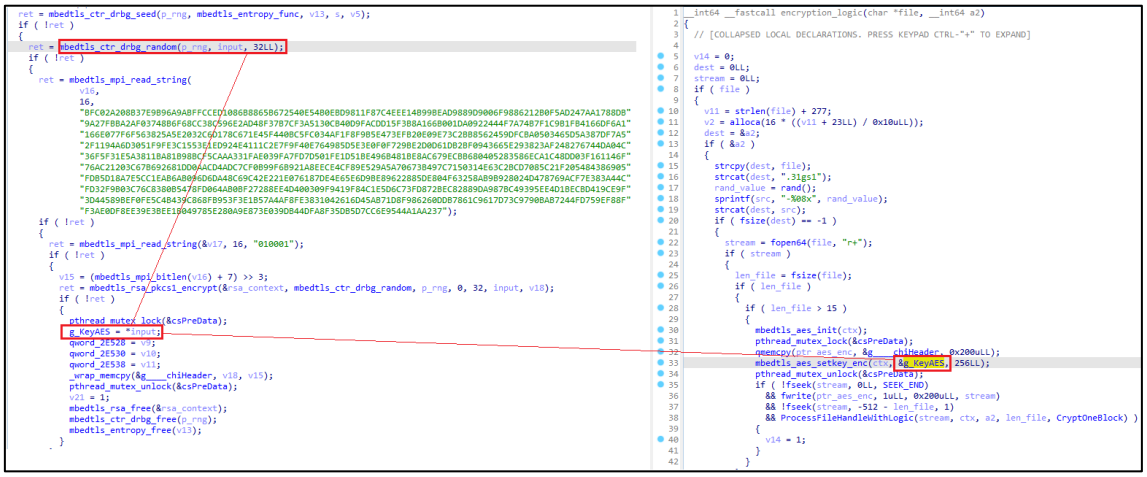


Figura 11. Generación claves AES

Las API *mbedtls_aes_init* y *mbedtls_aes_setkey_enc* se encargarán de inicializar el contexto AES y establecer, el grupo de 32 bytes previamente creado, como Key para cifrar, en modo ECB (Electronic CodeBook), el contenido de los ficheros.



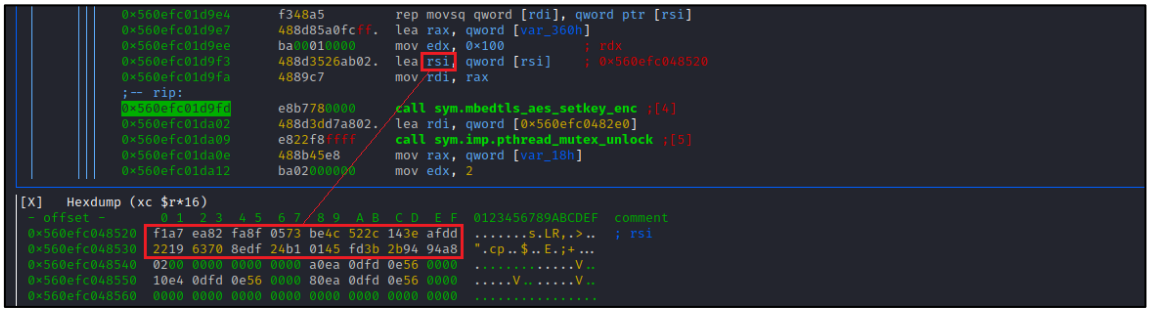


Figura 12. Generación claves AES

La clave AES será cifrada utilizando una clave RSA de 4096 bits (embebida en el código) y el *ciphertext* resultante (clave AES cifrada con RSA) será añadido al final de cada fichero cifrado. La lógica empleada para fijar la clave AES y para copiar dicho *ciphertext* es coordinada por medio de objetos Mutex (*pthread_mutex_lock* y

pthread_mutex_unlock) durante el proceso de cifrado para garantizar la exclusión mutua (véase el recuadro verde en la siguiente imagen).

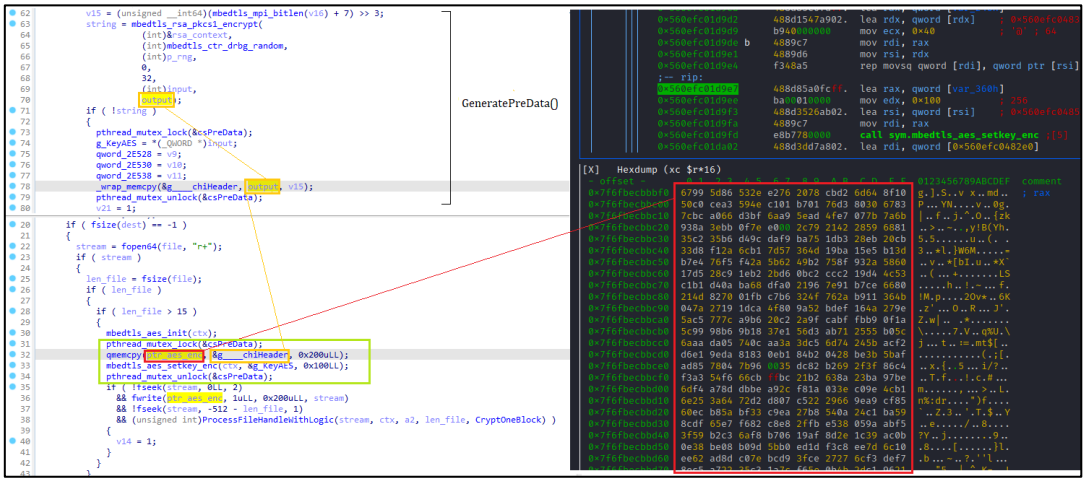


Figura 13. Generación claves AES / Ciphertext (clave AES cifrada con RSA)

Nota: Es posible que los binarios para ambas plataformas (Windows/Linux) compartan el mismo código fuente en aquellas funciones que interactúan con *mbed TLS* dada la similitud existente en la lógica empleada para la gestión y creación de claves.

A cada fichero cifrado se le añadirá la extensión "*.31gs1"* seguido de un valor en hexadecimal de 8 caracteres derivado de la API *rand()* y formateado con *sprintf* ("%08x").

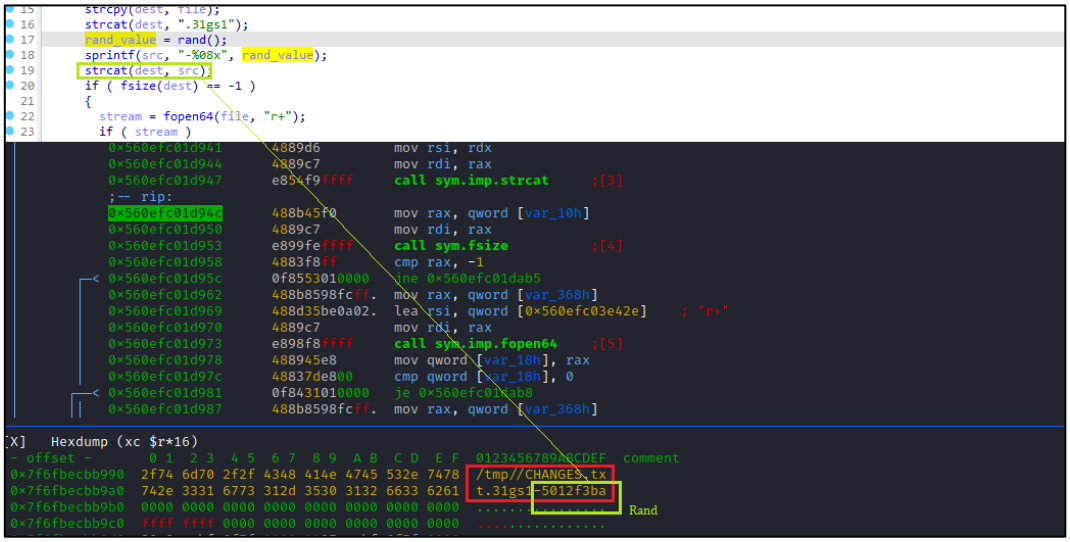


Figura 14. Renombrado de ficheros

ProcessFileHandleWithLogic será la función encargada de cifrar el contenido de cada fichero. Esta función recibe como argumento un puntero a la función *CryptOneBlock* responsable de utilizar la API de *mbedtls_aes_crypt_ecb* (perteneciente a *mbedtls*) para cifrar los bloques de datos de cada fichero. A modo de ejemplo, en la siguiente imagen se muestra el antes y después del cifrado de determinado fichero ordinario.

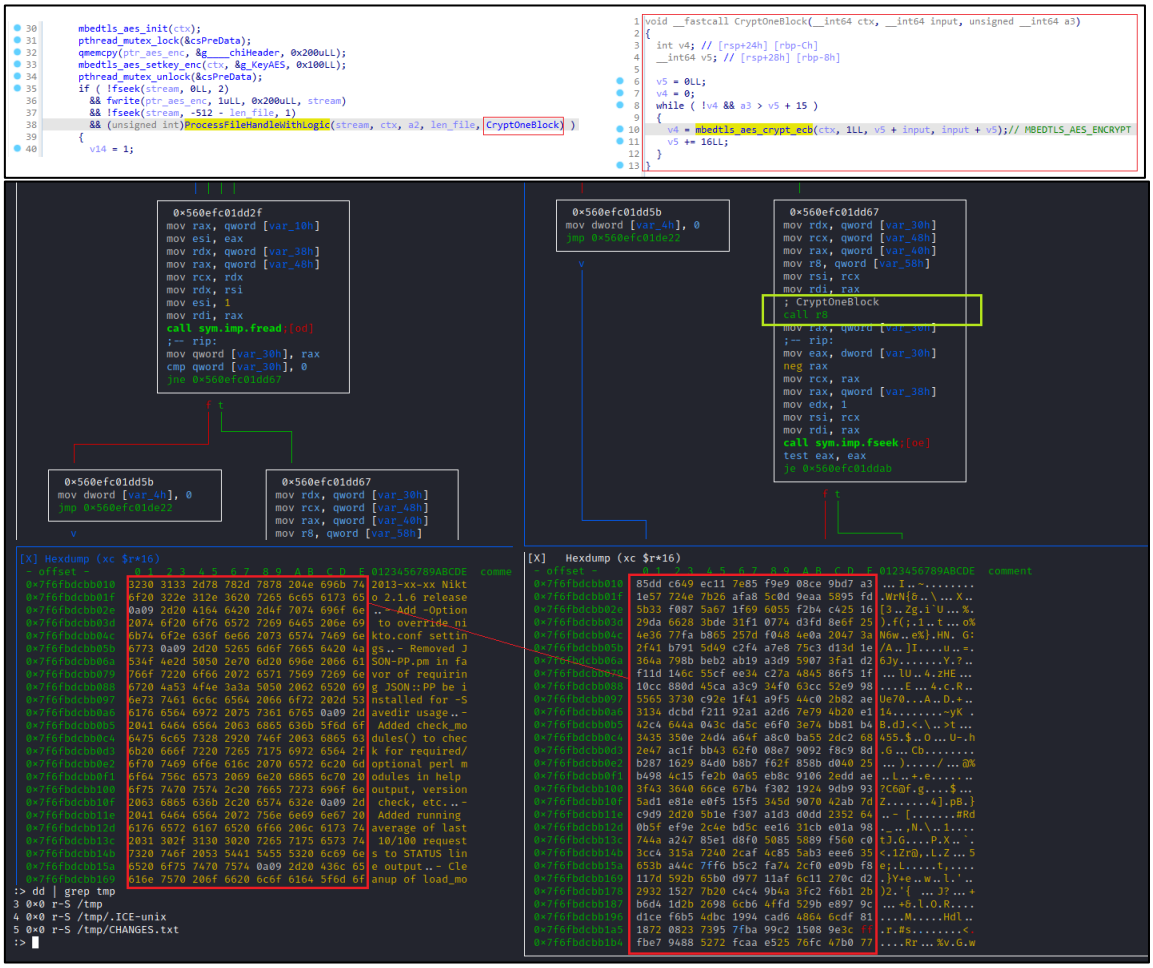


Figura 15. Cifrado de ficheros

A cada fichero se añadirá la clave AES cifrada con RSA al final del mismo garantizando de este modo que el único método factible para descifrar el mismo sea mediante la clave privada RSA (en manos del grupo cibercriminal).

Figura 16. Escritura clave AES (cifrada con RSA)

El siguiente *call graph* muestra el *path* de ejecución previamente descrito y recoge el conjunto de funciones más destacables que intervienen en la operativa del cifrado de ficheros.

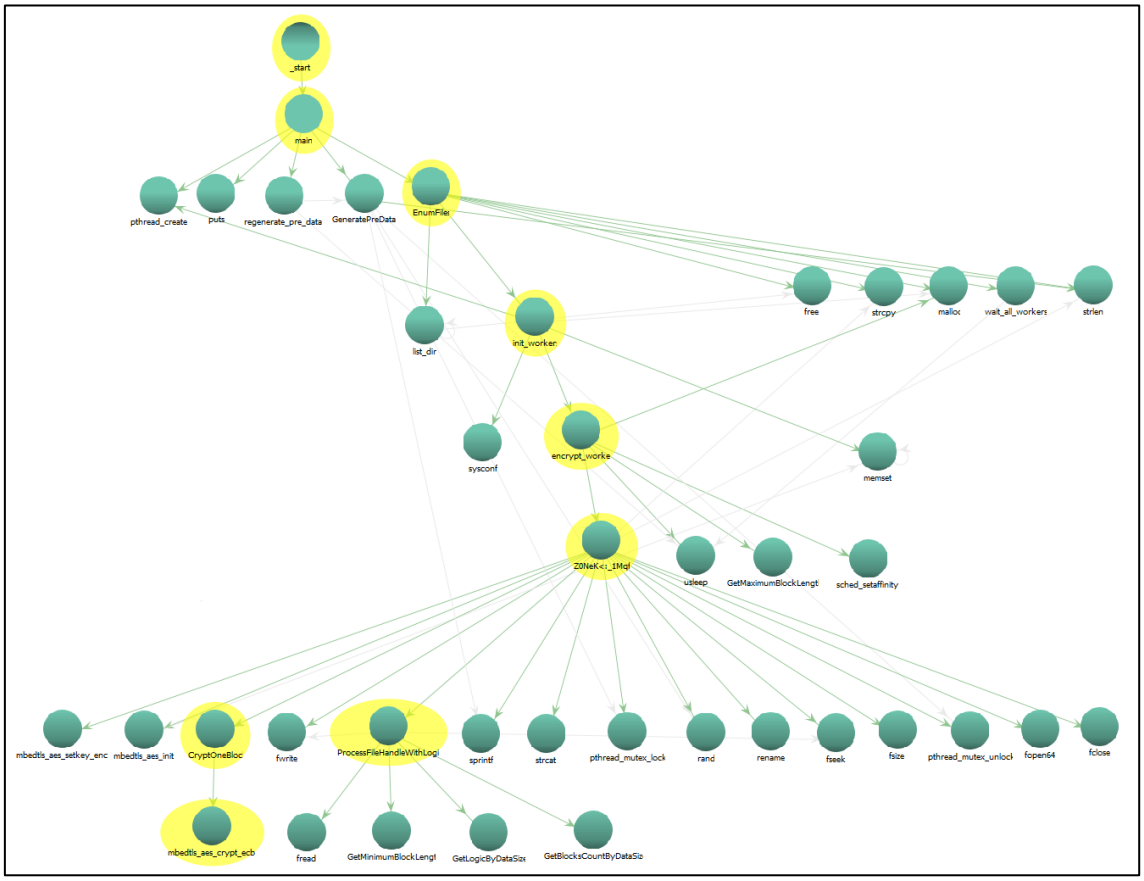


Figura 17. Call Graph

6. DESINFECCIÓN

Dado que la variante analizada no adquiere persistencia, los equipos afectados con RansomEXX no requieren de una rutina de desinfección más allá de reiniciar el sistema o finalizar el proceso vinculado con el *ransomware*. Cabe destacar que, aunque el propio binario dañino no implementa funcionalidades de persistencia, el conjunto de TTP empleadas por los atacantes para proceder con el despliegue del *ransomware* sí que puede involucrar otros componentes o técnicas para garantizar la persistencia del actor; por ejemplo, otros binarios dañinos, cuentas de usuarios a modo de *backdoor*, vulnerabilidades[16] en un servicio expuesto, etc. El proceso de respuesta a incidentes tendrá que investigar y analizar de forma meticulosa el vector de entrada, así como el conjunto de TTP empleado por los atacantes para garantizar que la mitigación ha sido efectiva y para evitar la repetición del ataque.

Respecto al descifrado de los ficheros, el modelo de criptografía utilizado asegura que el único método factible para el descifrado sea mediante el uso de la clave privada RSA.

7. MITIGACIÓN

La manera más estratégica para contener y reducir el impacto del *ransomware* es trabajar de forma paralela en el aislado de redes/VLAN que conforman la entidad

afectada (con el objetivo de contener los segmentos de red con equipos infectados y evitar su expansión) y en el rotado de contraseñas en el dominio. Dicho rotado debe incluir las cuentas locales de administrador (es importante que sean únicas para cada equipo) así como la cuenta KRBTGT del dominio. Téngase en cuenta que estas medidas pueden interferir en el correcto funcionamiento del dominio o redes afectadas y pueden causar diversos imprevistos (por ejemplo, el reinicio de máquinas); no obstante, permitiría contener la amenaza de una manera resolutiva.

8. REGLAS DE DETECCIÓN

8.1. REGLAS DE YARA

```
rule RansomEXX_Lin {
 meta:
   description = "Ransomware ransomEXX (Linux)"
   date = "2021-09-14"
   author = "CCN-CERT"

strings:
 $x1 = "ProcessFileHandleWithLogic" fullword ascii
 $x2 = "mbedtls_aes_crypt_ecb" fullword ascii
 $x3 = "ReadMeStoreForDir" fullword ascii
 $x4 = "!NEWS_FOR" fullword ascii
 $x5 = "enum_files.c" fullword ascii
 $x6 = {45 6E 63 72 79 70 74 65 64 20 66 69 6C 65 20 4D 55 53 54 20 4E 4F 54 20 68 61 76 65 20 72 69 63
68 20 64 61}
 $x7 = "Encrypted file MUST NOT have rich data" fullword ascii
 $x8 = "ReadMeStoreForDir" fullword ascii
 $x9 = { C7 00 2E 33 31 67 66 C7 40 04 73 31 C6 40 06 00 }      // .31gs1 extension
 $x10 = { 42 46 43 30 32 41 32 30 38 42 33 37 45 39 42 }        // Embeded RSA public key}
 $x11 = "regenerate_pre_data" fullword ascii
 $x12 = "ProcessFileHandleWithLogic" fullword ascii

 condition:
   (uint32(0) == 0x464C457F) and ((filesize > 100KB) and (filesize < 350KB)) and (3 of ($x*))
}
```

9. REFERENCIAS

[1] **Malpedia: GOLD DUPONT**

https://malpedia.caad.fkie.fraunhofer.de/actor/gold_dupont

[2] **Last, but Not Least: Defray777**

https://unit42.paloaltonetworks.com/vatet-pyxie-defray777/3/

[3] **Texas Courts hit by ransomware, network disabled to limit spread**

https://www.bleepingcomputer.com/news/security/texas-courts-hit-by-ransomware-network-disabled-to-limit-spread/

[4] **Ransomware attack impacts Texas Department of Transportation**

https://www.bleepingcomputer.com/news/security/ransomware-attack-impacts-texas-department-of-transportation/

[5] **Business technology giant Konica Minolta hit by new ransomware**

https://www.bleepingcomputer.com/news/security/business-technology-giant-konica-minolta-hit-by-new-ransomware/

[6] **Brazil's court system under massive RansomExx ransomware attack**

https://www.bleepingcomputer.com/news/security/brazils-court-system-under-massive-ransomexx-ransomware-attack/

[7] **Hackers leak data from Embraer, world's third-largest airplane maker**

https://www.zdnet.com/article/hackers-leak-data-from-embraer-worlds-third-largest-airplane-maker/

[8] **Double extortion ransomware**

https://www.darktrace.com/en/blog/double-extortion-ransomware/

[9] **Ransomware gangs are abusing VMWare ESXi exploits to encrypt virtual disks**

https://www.zdnet.com/article/ransomware-gangs-are-abusing-vmware-esxi-exploits-to-encrypt-virtual-hard-disks/

[10] **Mbed TLS**

https://github.com/ARMmbed/mbedtls

[11] **CCN-CERT: Informe Código Dañino: RansomEXX (Windows)**

https://www.ccn-cert.cni.es/en/reports/public/6158-ccn-cert-id-11-21-ransomexx-win-ransomware-1/file.html

[12] **RansomEXX Trojan attacks Linux systems**

https://securelist.com/ransomexx-trojan-attacks-linux-systems/99279/

[13] **MalwareBazaar Database: cb408d45762a628872fa782109e8fcfc3a5bf456074b007de21e9331bb3c5849**

https://bazaar.abuse.ch/sample/cb408d45762a628872fa782109e8fcfc3a5bf456074b007de21e9331bb3c5849/

[14] **RansomExx ransomware also encrypts Linux systems:**

https://www.bleepingcomputer.com/news/security/ransomexx-ransomware-also-encrypts-linux-systems/

[15] **Computer hardware giant GIGABYTE hit by RansomEXX ransomware:**

https://www.bleepingcomputer.com/news/security/computer-hardware-giant-gigabyte-hit-by-ransomexx-ransomware/