ANÁLISIS DE CRIPTOGRAFÍA USADA POR RANSOMWARE MEDIANTE INSTRUMENTACIÓN DINÁMICA DE BINARIOS

Autor: Ignacio Samuel Crespo Martínez

Tutor: Ricardo J. Rodríguez Fernández

Contenido

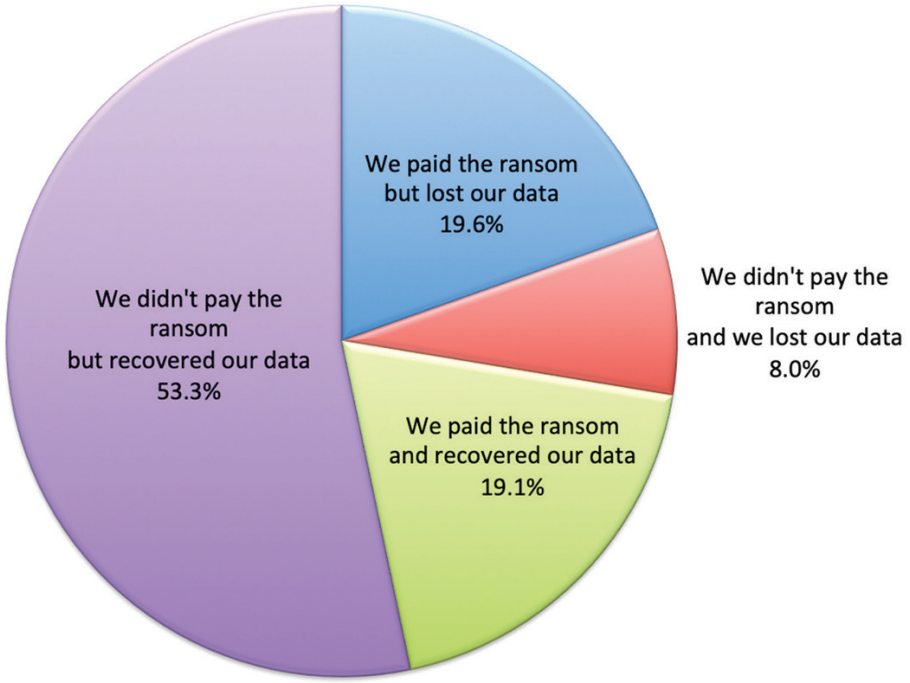
- Introducción
- Conceptos previos
- Sistema desarrollado
- Experimentos y evaluación
- Estado del arte
- Conclusiones y trabajo futuro

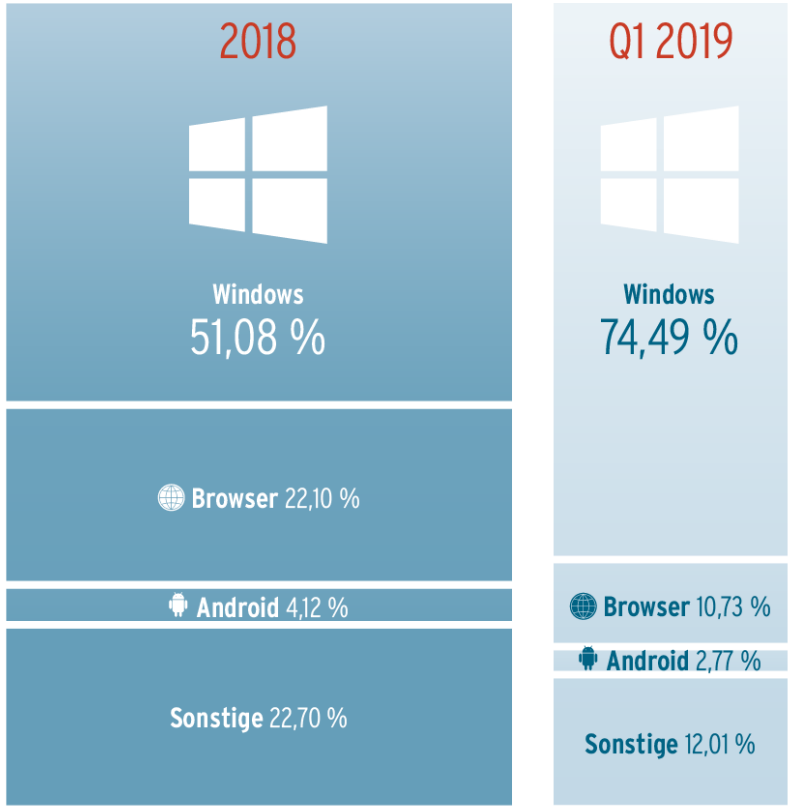
Introducción

Ransomware

- Es un tipo malware diferente al tradicional
  - Hacer imposible el acceso a los datos
  - No es sigiloso, notifica a la victima
  - Fácil de desarrollar
- Tipos de ransomware: *crypto y locker*
- Vectores de ataque
  - Spam/ingeniería social
  - Publicidad maliciosa
  - Herramientas de instalación de malware o botnets

Introducción



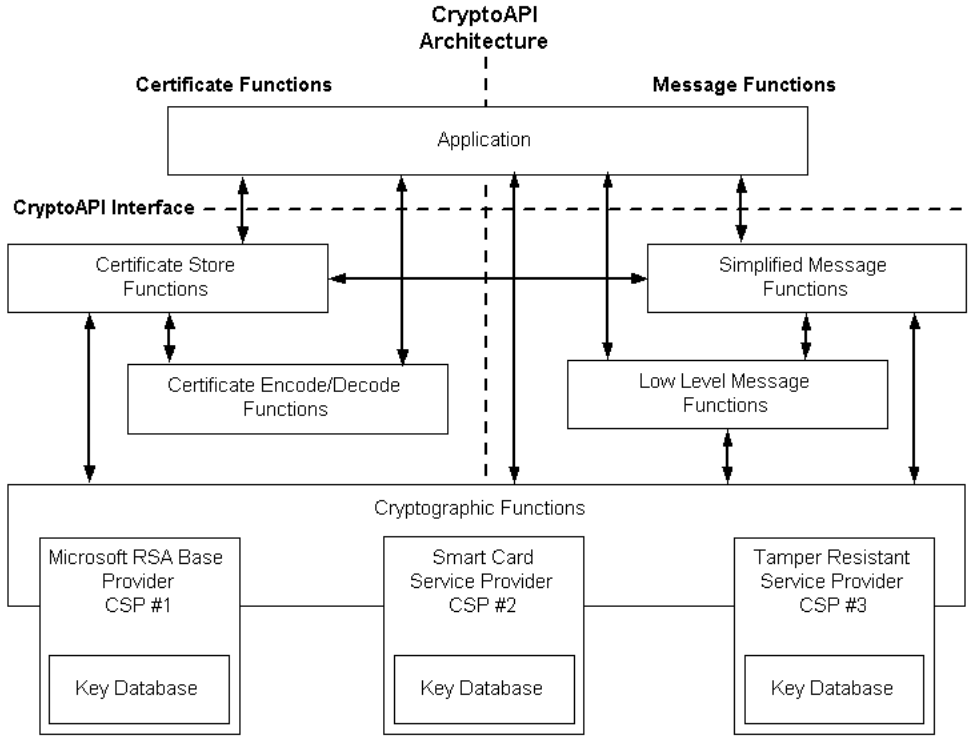


Introducción

- Objetivos del proyecto
  - Conocer las características de la criptografía usada por el ransomware
  - Estudio de las familias de ransomware a lo largo de los últimos años (de 2014 a 2018)
  - Uso de técnicas de instrumentación dinámica de binarios
    - Monitorización de funciones de sistema relacionadas con la criptografía

Conceptos previos

- CryptoAPI
  - Librería que proporciona servicios criptográficos a las aplicaciones de los usuarios en Windows
  - Formado por cinco áreas principales:
    - Funciones de codificación/decodificación de certificados
    - Funciones para almacenar certificados
    - Funciones de mensaje simplificadas
    - Funciones de mensaje de bajo nivel
    - Funciones criptográficas base



Conceptos previos

- Proveedor de servicios criptográficos (CSP)
  - Ninguna operación criptográfica específica de un algoritmo de cifrado es parte de la CryptoAPI
  - Contiene todas las implementaciones de estándares y algoritmos criptográficos
  - Compuesto por una librería de sistema
- Algoritmo de cifrado
  - Parámetro ALG_ID
  - Utilizado para generación de claves y para cifrar
  - Nuevo CSP puede definir nuevos valores

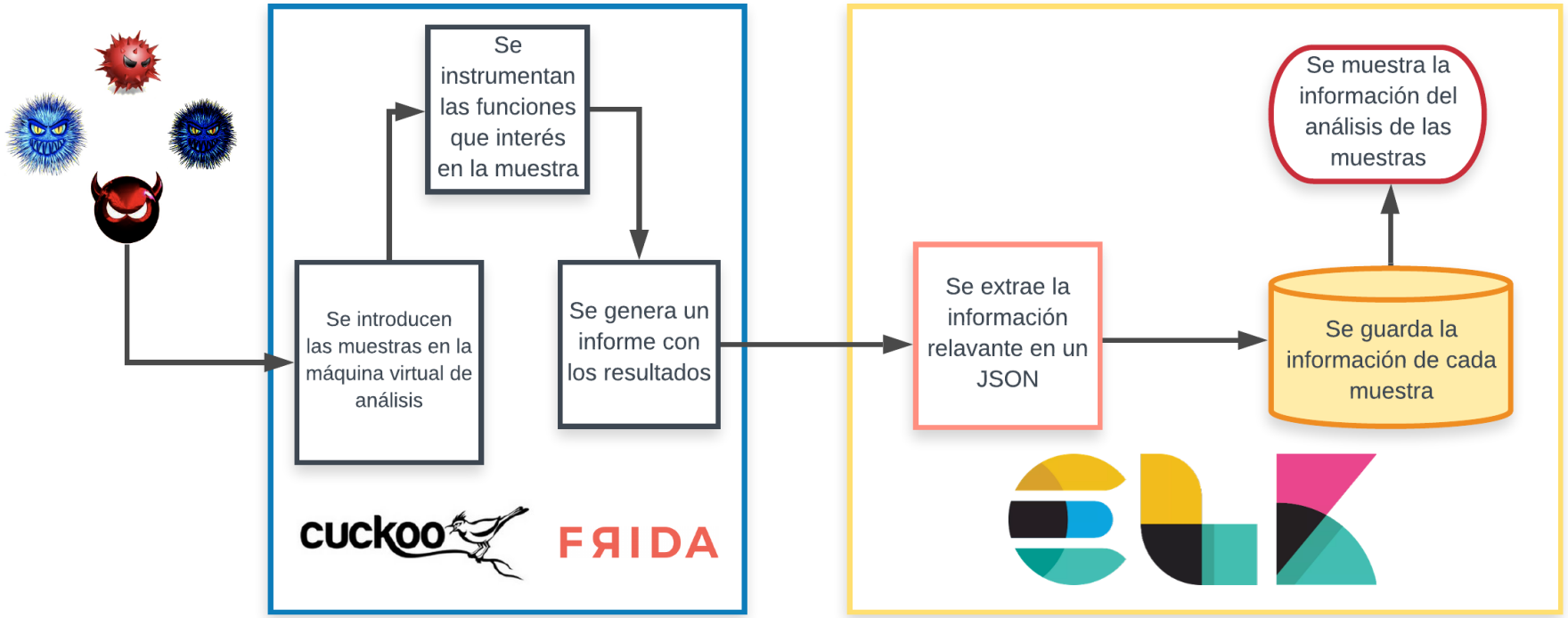
Conceptos previos

- ¿ Qué es la instrumentación dinámica de binarios?

  Técnica que altera el comportamiento de las funciones inyectando código arbitrario en la ejecución normal de una aplicación

Herramientas	Desventajas/Ventajas
DRAKVUF	- Alta dependencia del hardware
Drltrace	- No permite elegir qué funciones se quieren registrar - No se puede lanzar sobre un proceso en ejecución
WinAPIOverride	- No permite elegir qué funciones se han de monitorizar - Aplicación de pago que no permite ser instalada en maquinas virtuales
EasyHook	- Funciones de instrumentación escritas en C++ - Únicamente para aplicaciones creadas para .NET Framework 3.5/4.0+, así como librerías DLL nativas
Frida	- Integración en un sistema Windows es fácil - Los resultados de un análisis pueden ser manejados de varias formas - Muestra las llamadas al sistema o a funciones propias, además de los parámetros y sus valores de retorno

Sistema desarrollado



Sistema desarrollado (sistema de análisis)

Descripción del entorno de análisis

- Cuckoo Sanbox está montado sobre un Ubuntu 18.04.2 LTS y trabaja juntamente con VirtualBox
- Se realizan los análisis sobre una máquina virtual Windows 7 x86 que tiene instalado Frida y Python 2.7

Sistema desarrollado (sistema de análisis)

- Cuckoo Sandbox está compuesto por diferentes módulos:
  - *Auxiliary:* define los procedimientos que se llevan a cabo a la vez que el procedimiento de análisis
  - *Machinery:* define como interactúa Cuckoo con el software de virtualización
  - *Analysis Package:* describe como se lleva a cabo el procedimiento de análisis
  - *Processing:* define como se analizan los resultados del análisis
  - *Reporting:* recoge los resultados procesados y los hace accesibles

Sistema desarrollado (sistema de análisis)

- Módulo Analysis Package
  - Coge una muestra de ransomware y la copia a la máquina virtual de análisis
  - Lanza el análisis con unas opciones determinadas
  - Recoge los resultados y los envía el módulo *Processing*

Sistema desarrollado (sistema de análisis)

- Frida
  - Ejecuta la muestra en la máquina de análisis
  - Instrumenta las funciones de la CryptoAPI

Funciones CryptoAPI
CryptEncrypt()
CryptHashData()
CryptDeriveKey()
CryptDecrypt()
CryptAcquireContextA()
CryptAcquireContextW()
CryptExportKey()
CryptGenKey()
CryptSetKeyParam()
CryptStringToBinaryA()
CryptGetKeyParam()
CryptCreateHash()

Sistema desarrollado (sistema de análisis)

- Frida
  - Ejecuta la muestra en la máquina de análisis
  - Instrumenta las funciones de la CryptoAPI
  - Para cada función se desarrolla un código JavaScript

Sistema desarrollado (sistema de análisis)

```
Interceptor.attach(Module.findExportByName(null, 'CryptAcquireContextA'),
    {
        onEnter: function (args) {
            var file = new File("C:\\Users\\test\\Desktop\\frida.br","a");
            file.write('CryptAcquireContextA() attach\n');
            file.write("[+] HCRYPTPROV *phProv: " + args[0] + "\n");
            file.write("[+] LPCSTR     szContainer: " + args[1] + "\n");
            file.write("[+] LPCSTR     szProvider: " + args[2] + "\n");
            file.write("[+] DWORD      dwProvType: " + args[3] + "\n");
            file.write("[+] DWORD      dwFlags: " + args[4] + "\n\n");
            file.close();
        },

        onLeave: function (retval) {
        }
    });
```

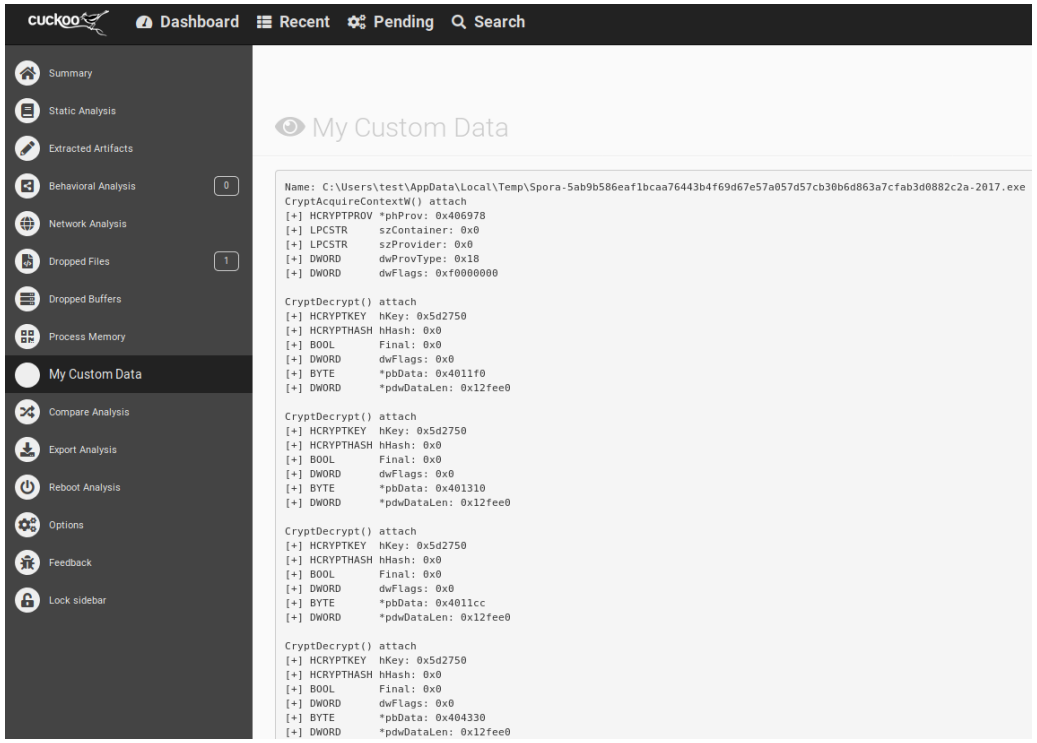
Sistema desarrollado (sistema de análisis)

- Frida
  - Ejecuta la muestra en la máquina de análisis
  - Instrumenta las funciones de la CryptoAPI
  - Para cada función se desarrolla un código JavaScript
  - Genera el resultado del análisis en un fichero “frida.br”

Sistema desarrollado (sistema de análisis)

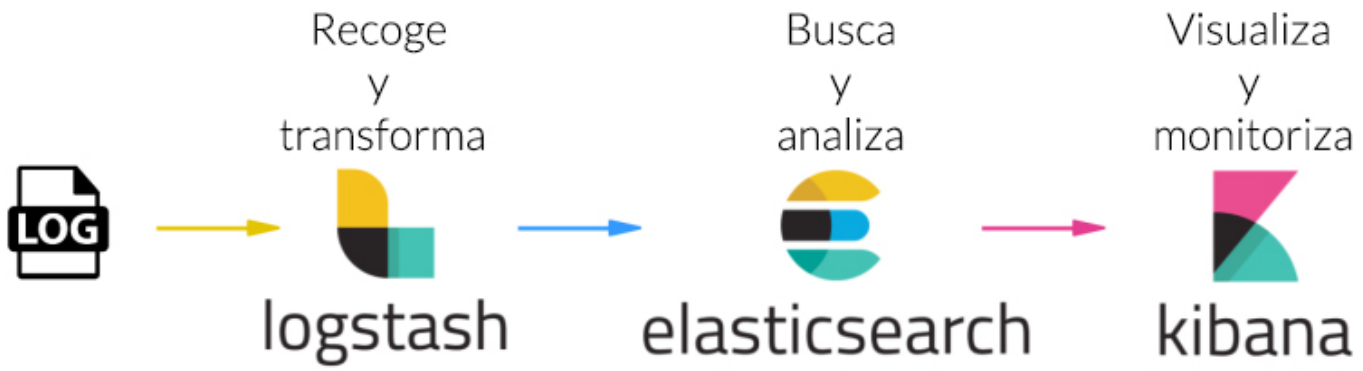
- Módulo *Processing*
  - Construye un JSON que contiene los siguientes campos:
    - *Name, Hash, Date, Algorithm, ProviderType, Functions, Full_Log*
  - Cada JSON es guardado en un fichero llamado “samples.json”

- Módulo *Reporting*
  - Recibe el resultado completo del análisis
  - Lo muestra en la consola web de Cuckoo



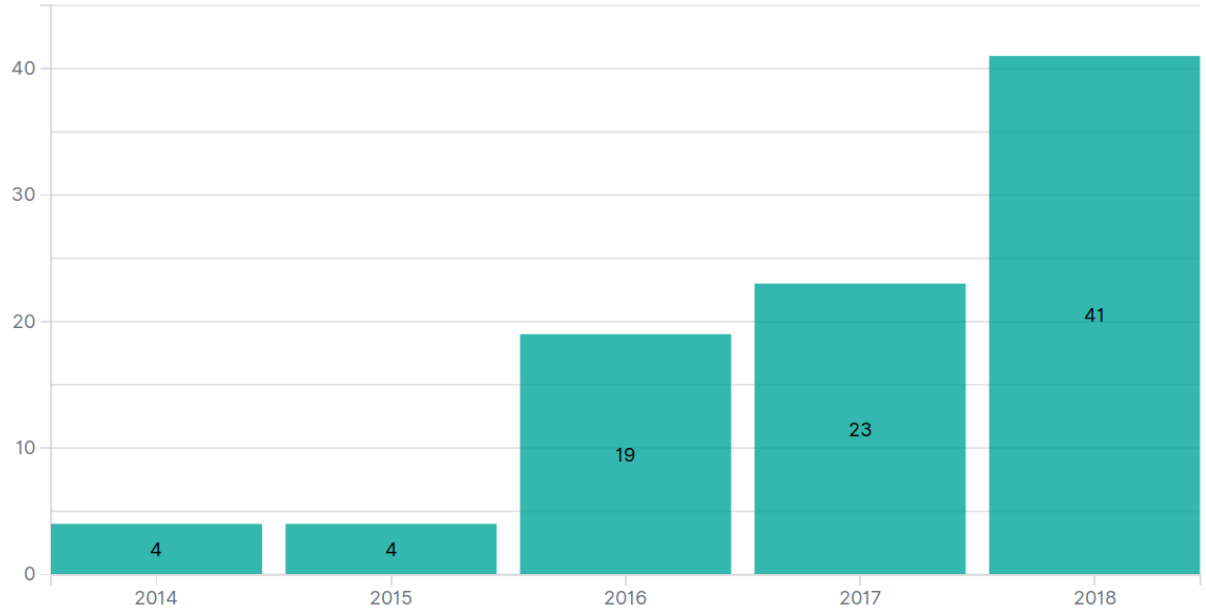
Sistema desarrollado (Sistema de tratamiento de información)

- Compuesto por un ELK



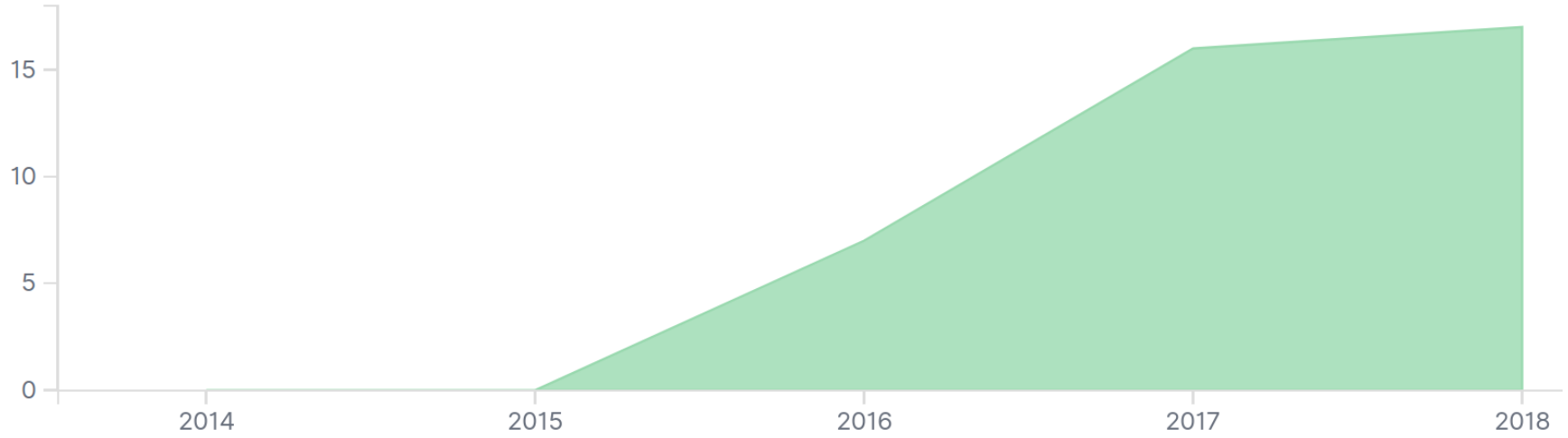
Experimentos y evaluación

- 99 muestras de ransomware analizadas



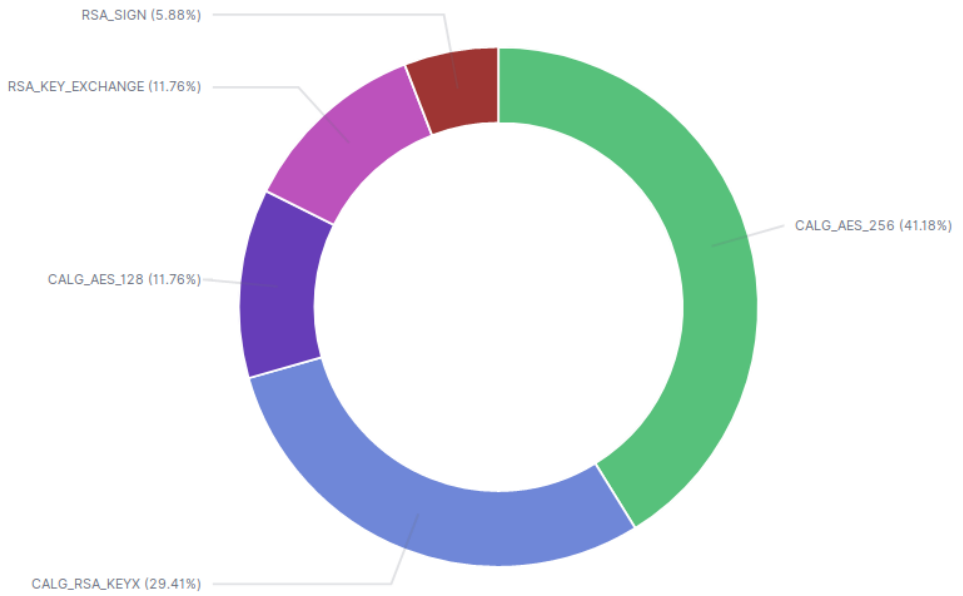
Experimentos y evaluación

- El 40,4% de las muestras analizadas utilizaban la CryptoAPI



Experimentos y evaluación

- El 40 % de las muestras utilizaban PROV_RSA_FULL como CSP y el restante PROV_RSA_AES
- Solo se utilizan algoritmos RSA y AES



Experimentos y evaluación

- El 40 % de las muestras utilizaban PROV_RSA_FULL como CSP y el restante PROV_RSA_AES
- Solo se utilizan algoritmos RSA y AES
- El algoritmo involucrado en el cifrado no siempre está presente en el resultado del análisis

Estado del arte

- Trabajos centrados en soluciones para detectar anomalías
  - Implementando un CSP (A. Palisse et al., 2016)
  - A través de ficheros tipo FIFO (N. Engineering and S. Group, 2017)
  - Detección de anomalías en el sistema de ficheros (A. Continella et al., 2016)
  - Observando las operaciones de entrada/salida (A. Kharraz and E. Kirda, 2017)
- Análisis de los ataques de ransomware
  - La mayoría del ransomware es de tipo locker (A. Kharraz et al., 2015)

Conclusiones y trabajo futuro

- CryptoAPI sigue siendo usada por el ransomware
- Algoritmos utilizados son RSA y AES

Trabajo futuro

- Análisis pero con la evolución de CryptoAPI, la Cryptography API: Next Generation (CNG)
- Creación de una herramienta que monitorice la actividad de las aplicaciones y detenga su ejecución


FIN