Gestión de vulnerabilidades

Podemos escanear las redes corporativas en busca de vulnerabilidades usando software de gestión de vulnerabilidades. Si encontramos una vulnerabilidad durante el escaneo, las herramientas de gestión de vulnerabilidades sugerirán o activarán un parche. Las tecnologías de gestión de vulnerabilidades escanean redes en busca de brechas de seguridad y las corrigen para evitar más intrusiones. Por lo tanto, el daño que puede causar un ciberataque se reduce mediante el uso de sistemas de gestión de vulnerabilidades. Las correcciones oportunas deben implementarse según las prioridades establecidas por las herramientas de gestión de vulnerabilidades. Mediante un proceso metódico, puedes reducir tu dependencia de sistemas de detección de intrusiones de terceros mientras fortaleces tu red.

Lolbas – project

> Living of the Land binaries (LOLBins) son ejecutables legítimos de Windows que pueden ser utilizados por atacantes para realizar acciones maliciosas sin levantar sospechas. Utilizar LOLBins permite a los atacantes integrarse en la actividad normal del sistema y evitar ser detectados, lo que los convierte en una opción popular para los atacantes. El proyecto LOLBAS es una lista MITER mapeada de LOLBINS con comandos, información de uso e detección para defensores.

Visita https://lolbas-proetc.github.io/ .

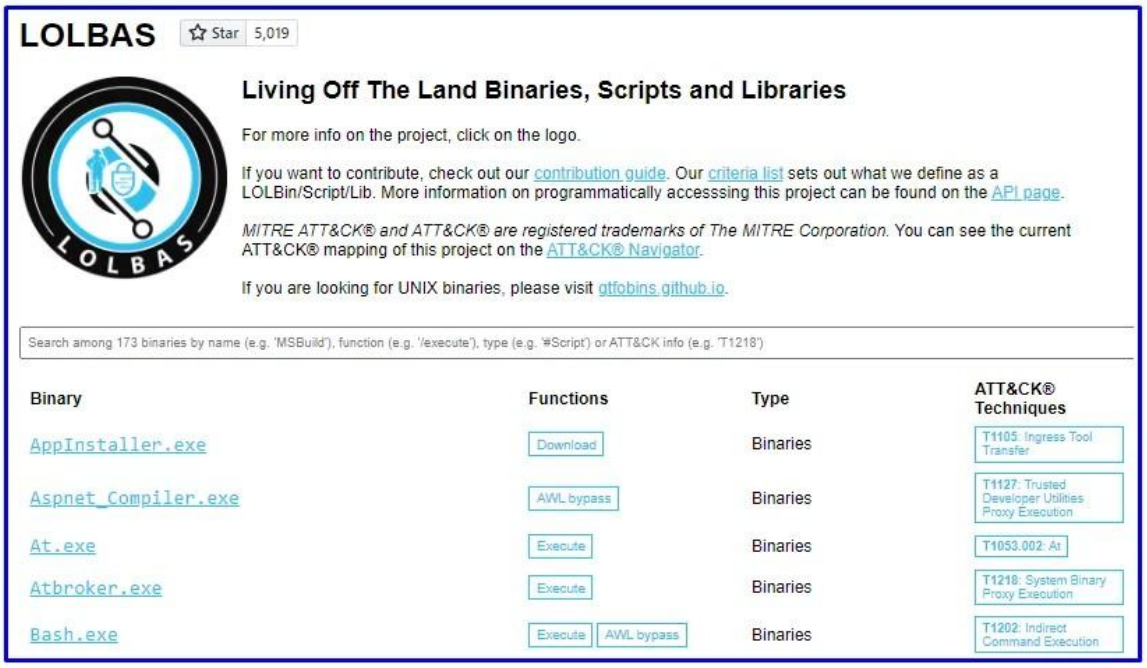
Uso:

Utiliza la información para capacidades de detección y así proteger tu infraestructura del uso de LOLBIN.

Aquí tienes algunos enlaces de proyectos para que empieces:

- Bitsadmin.exe
- Certutil.exe

Cscript.exe



GTFOBins

> GTFOBins (abreviatura de "Get The F* Out Binaries") es una colección de binarios Unix que pueden usarse para elevar privilegios, eludir restricciones o ejecutar comandos arbitrarios en un sistema. Pueden ser utilizados por atacantes para obtener acceso no autorizado a sistemas y realizar acciones maliciosas. El proyecto go awayBins es una lista de binarios Unix con información de comandos y uso para atacantes. Esta información puede utilizarse para implementar la detección Unix.

Visita https:// GTFOBINS.github.io/ .

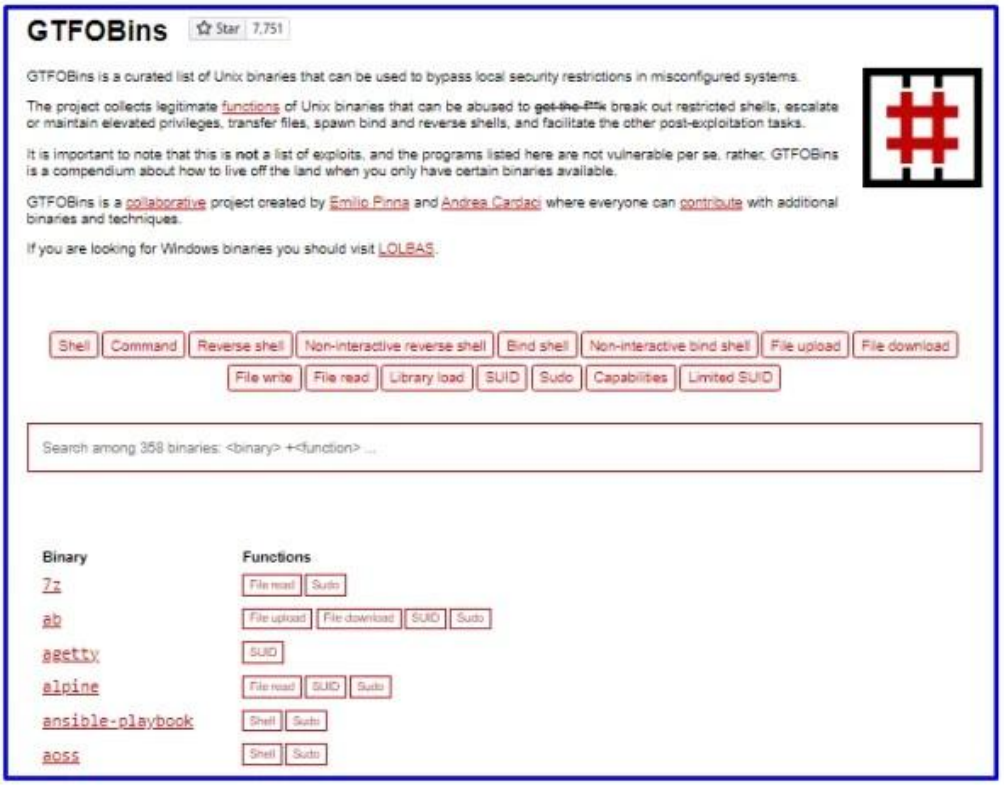
Uso:

Aquí tienes algunos enlaces de proyectos para que empieces:

base64

curl

nano



Filesec

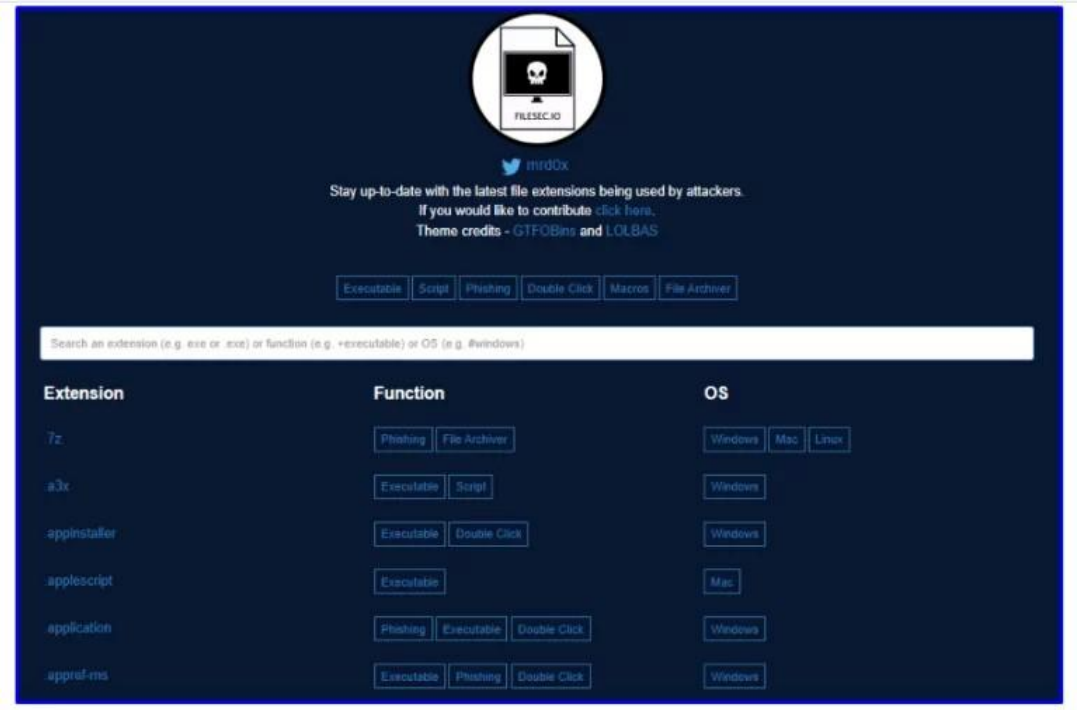
> Filesec es una lista de extensiones de archivo que los atacantes pueden usar para phishing, ejecución, macros y más. Es un recurso excelente para entender los exploits comunes de extensiones de archivo y cómo protegerse contra ellos. Cada página de extensión de archivo contiene una descripción, sistema operativo correspondiente y recomendaciones.

Visita https://filesec.io/ .

Uso:

Aquí tienes algunos enlaces de proyectos para que empieces:

- .Docm
- .Iso
- .Ppam



KQL Search

KQL significa "Kusto Query Language" y es el lenguaje de consulta utilizado para buscar y filtrar datos en los registros de Azure Monitor. Es similar a SQL, pero más optimizado para analizar registros y datos de series temporales. El lenguaje de consulta KQL es especialmente útil para los equipos azules porque permite buscar rápida y fácilmente entre grandes volúmenes de datos de registro para identificar eventos de seguridad y anomalías que puedan indicar una amenaza. KQL Search es una aplicación web creada por @ugurkocde que agrega consultas KQL compartidas en GitHub.

Puedes visitar el sitio en https://www.kqlsearch.com/. Más información sobre el Lenguaje de Consulta Kusto (KQL) se puede encontrar aquí.

Unprotect

Los autores de malware dedican mucho tiempo y esfuerzo a desarrollar código complejo para realizar acciones maliciosas contra el sistema objetivo. Es muy importante que el malware permanezca indetectable y evite el sandboxing, el antivirus o el análisis de malware. Gracias a esta técnica, el malware puede pasar desapercibido y permanecer indetectado en el sistema. El objetivo de esta base de datos gratuita es centralizar la información sobre técnicas de evitación de malware. El proyecto pretende proporcionar a los analistas y defensores de malware información accionable y capacidades de detección para reducir el tiempo de respuesta.

El proyecto se puede consultar en https://unprotect.it/. El proyecto tiene la documentación de la API aquí.

Chainsaw

Chainsaw proporciona una potente capacidad de "primer interviniente" para identificar rápidamente amenazas en artefactos forenses de Windows, como registros de eventos y MFTs. Chainsaw ofrece un método común y rápido para buscar en registros de eventos palabras clave e identificar amenazas, con soporte integrado para las reglas de detección Sigma y las reglas personalizadas de detección de Chainsaw.

Funciones:

- Busca amenazas usando reglas de detección Sigma y reglas especiales de detección de motosierras Busca y extrae artefactos forenses usando coincidencia de cadenas y patrones de expresión regular

Lightning, escrito en Rust, cubre la biblioteca de analizadores EVTX de @OBenamram
Ejecución y formatos de salida limpios y ligeros sin sobrecarga
El etiquetado de documentos (lógica de descubrimiento coincidente) proporcionado por la Biblioteca del Motor TAU.
Los resultados de salida se reciben en varios formatos, como formato de tabla ASCII, formato CSV y formato JSON

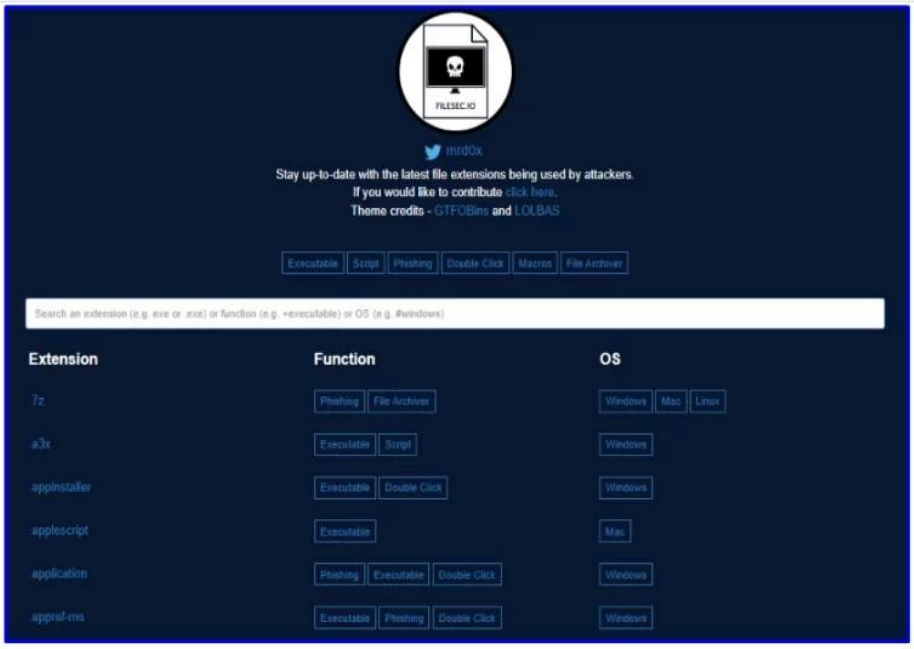
Puede funcionar en MacOS, Linux y

Windows

Instalación:

```
git clone https://github.com/countercept/chainsaw.git
cargo build --release
git clone https://github.com/SigmaHQ/sigma
git clone https://github.com/sbousseaden/EVTX-ATTACK-SAMPLES.git
```

Uso:




```
./chainsaw hunt EVTX-ATTACK-SAMPLES/ -s sigma/ --mapping mappings/sigma-event-logs-all.yml
```

Freq

Los atacantes intentan saltarse métodos basados en firmas/coincidencias de patrones/listas negras introduciendo al azar: nombres de archivos, nombres de servicios, nombres de estaciones de trabajo, dominios, nombres de host, sujetos de certificados SSL y sujetos de publicadores, etc. Freq es una API en Python desarrollada por Mark Baggett para gestionar pruebas de entropía masivas. Está diseñado para usarse con soluciones SIEM, pero puede funcionar con cualquier cosa que envíe una solicitud web. La herramienta utiliza tablas de frecuencia que muestran la probabilidad de que un personaje siga a otro

Instalación:

```
git clone https://github.com/MarkBaggett/freq
cd freq
```

Uso:

```
# Running freq_server.py on port 10004 and using a frequency table of /opt/freq/dns.freq
/usr/bin/python /opt/freq/freq_server.py 10004 /opt/freq/dns.freq
```

yarGen

> yarGen es un generador de reglas YARA. El principio básico es crear reglas yara a partir de las líneas que se encuentran en los archivos de malware, eliminando cualquier línea que también aparezca en los archivos de software buenos.

Así, yarGen incluye grandes cadenas de buen software y una base de datos de códigos de operación en forma de archivos ZIP que deben descomprimirse antes de su primer uso. El proceso de creación de reglas también intenta determinar similitudes entre los archivos analizados y luego combina las cadenas en las

llamadas superreglas. Generar una superregla no elimina una regla simple para archivos combinados en una sola superregla. Esto significa que hay cierta redundancia al crear superreglas. Puedes suprimir una regla simple para un archivo que ya está cubierto por una superregla usando –nosimple.

Instalación:

```
pip install -r requirements.txt
python yarGen.py --update
```

Descarga la última versión.

Uso:

Aquí se pueden encontrar ejemplos de uso.

```
# Create a new strings and opcodes database from an Office 2013 program directory
yarGen.py -c --opcodes -i office -g /opt/packs/office2013

# Update the once created databases with the "-u" parameter
yarGen.py -u --opcodes -i office -g /opt/packs/office365
```

EmailAnalyzer

Con EmailAnalyzer puedes analizar correos electrónicos sospechosos. Puedes extraer cabeceras, enlaces y hashes de un archivo .eml

Instalación:

```
git clone https://github.com/keraattin/EmailAnalyzer
cd EmailAnalyzer
```

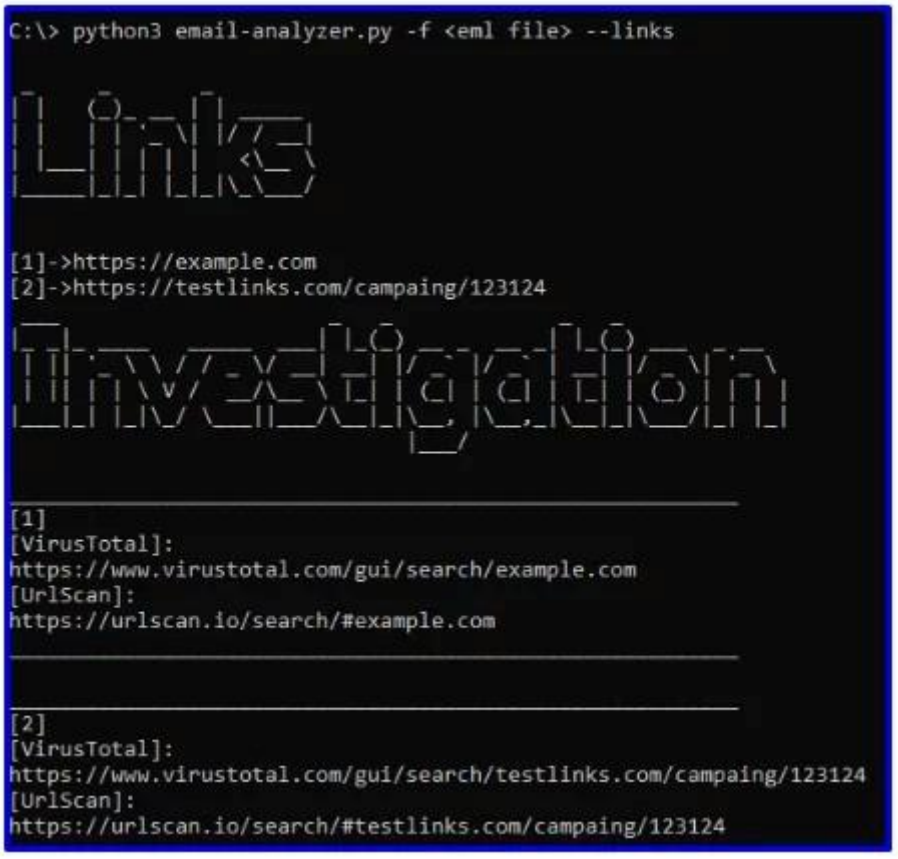
Uso:

```
# View headers in eml file
python3 email-analyzer.py -f <eml file> --headers

# Get hashes
python3 email-analyzer.py -f <eml file> --digests

# Get links
python3 email-analyzer.py -f <eml file> --links

# Get attachments
python3 email-analyzer.py -f <eml file> --attachments
```

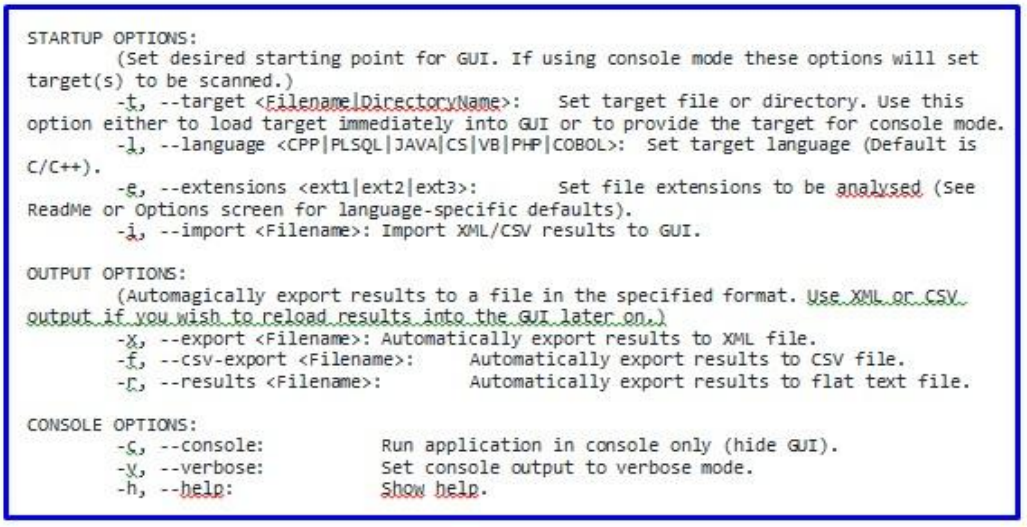


VCG

VCG es una herramienta automatizada de pruebas de seguridad de código que funciona con C/C++, Java, C#, VB y PL/SQL. Cuenta con varias funciones que esperamos sean útiles para cualquiera que realice pruebas de seguridad de código, especialmente cuando el tiempo es limitado:

1. Además de hacer comprobaciones más complejas, también tiene un archivo de configuración para cada lenguaje, que básicamente te permite añadir cualquier característica defectuosa (u otro texto) que quieras buscar
2. Intenta encontrar unas 20 frases en los comentarios que puedan indicar código roto ("ToDo", "FixMe", "Kludge", etc.).
3. Proporciona un buen gráfico circular (para toda la base de código y para archivos individuales) que muestra las proporciones relativas de código, espacios en blanco, comentarios, comentarios estilo "ToDo" y código defectuoso

Uso:



```
STARTUP OPTIONS:
        (Set desired starting point for GUI. If using console mode these options will set
target(s) to be scanned.)
        -t, --target <Filename|DirectoryName>:   Set target file or directory. Use this
option either to load target immediately into GUI or to provide the target for console mode.
        -l, --language <CPP|PLSQL|JAVA|CS|VB|PHP|COBOL>:  Set target language (Default is
C/C++).
        -e, --extensions <ext1|ext2|ext3>:       Set file extensions to be analysed (See
ReadMe or Options screen for language-specific defaults).
        -i, --import <Filename>: Import XML/CSV results to GUI.

OUTPUT OPTIONS:
        (Automagically export results to a file in the specified format. Use XML or CSV
output if you wish to reload results into the GUI later on.)
        -x, --export <Filename>: Automatically export results to XML file.
        -f, --csv-export <Filename>:     Automatically export results to CSV file.
        -r, --results <Filename>:        Automatically export results to flat text file.

CONSOLE OPTIONS:
        -c, --console:          Run application in console only (hide GUI).
        -v, --verbose:          Set console output to verbose mode.
        -h, --help:             Show help.
```