Herramientas de Seguridad y Python

ESET Latinoamérica | *info@eset-la.com* | *www.eset-la.com*

Contenido

Introducción

Hasta este momento vimos un montón de componentes del lenguaje y cómo utilizarlos dentro de nuestro código. Python, al igual que muchos otros lenguajes de programación, tiene más de un uso y se adapta a las necesidades de quienes lo usan. A lo largo de los años la comunidad ha desarrollado una gran cantidad de herramientas de seguridad completamente programadas en este lenguaje, o incluso se han agregado interfaces que nos permiten extender o automatizar sus capacidades a través del uso de Python.

En relación a la seguridad y al estudio de códigos maliciosos, Python cuenta con una gran cantidad de librerías que nos ayudan en las más diversas tareas. Desde el cálculo de hashes, identificación de archivos, ***packers*** y desensamblado automático hasta ***debuggers*** completamente armados en Python como ***pydbg***[1], o las extensiones que se pueden hacer a herramientas como el ***Immunity Debugger***[2] o IDA Pro. La sencillez y capacidades de este lenguaje lo han masificado entre los especialistas de seguridad y en parte también, entre usuarios malintencionados.

A lo largo de este último módulo del curso de Introducción a Python, veremos algunas de las herramientas que se han decidido por este lenguaje para brindarle a los especialistas la más amplia gama de recursos.

Herramientas de Análisis de Malware y Python

Python cuenta con un gran conjunto herramientas asociadas a la seguridad y en particular al análisis de malware. Debido a su gran versatilidad y sencillez Python ha sido seleccionado en más de una oportunidad para ser el lenguaje con el cual se desarrollan e implementan estos sistemas.

A continuación veremos algunas de las herramientas que se utilizan para el análisis y estudio de códigos maliciosos entre algunas otras de sus funciones.

Pydbg

Pydbg es un ***debugger*** de Ring 3 escrito completamente en Python, que nos permite ejecutar un programa y estudiar todas las acciones que realiza en el sistema, pudiendo definir los ***breakpoints***, ***hooks*** y muchas cosas más. Esta herramienta es *Open Source* y pueden acceder a su código completo desde el repositorio de **GitHub** de **OpenRCE**[3].

Para poder utilizar ***pydbg*** en un sistema primero tienen que instalar algunas dependencias como PaiMei[4] y la librería de ***ctypes*** de Python.

Además de poder escribir nuestros propios scripts para realizar tareas de debugging este ***framework*** también nos da la posibilidad de desarrollar nuestras propias herramientas de monitorización a través de su API. No existe un camino sencillo para conocer y aprovechar todo el potencial de herramientas como ***pydgb*** salvo que la práctica, dedicación y el tiempo.

Immunity Debugger

Immunity Debugger es una excelente herramienta para el análisis de códigos maliciosos, ***exploits*** y otras cuantas acciones. Además de todas sus funcionalidades por defecto, incluye también una API que nos permite utilizar scripts en Python para la automatización de tareas y hasta desarrollar nuestros propios scripts.

---

[1] Pydbg: https://github.com/OpenRCE/pydbg
[2] Immunity Debugger: http://www.immunityinc.com.ar/products-immdbg.shtml
[3] Repositorio de OpenRCE en GitHub: https://github.com/OpenRCE
[4] PaiMei: https://github.com/OpenRCE/paimei

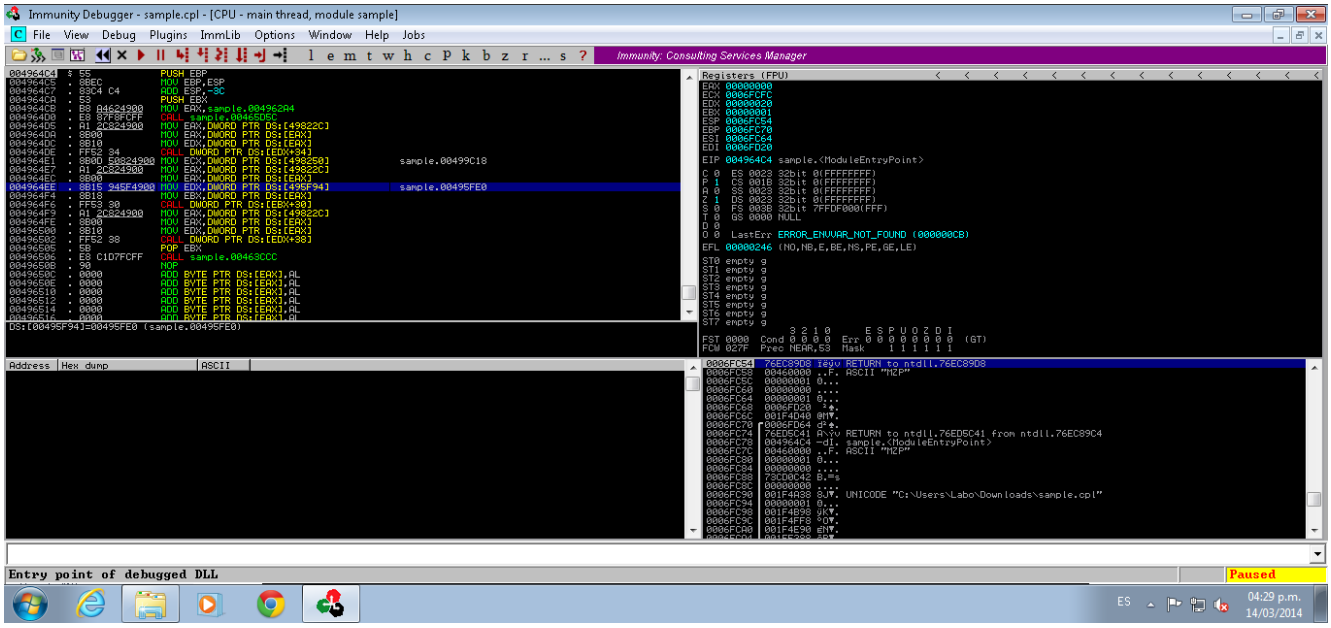

Imagen 1 - Immunity Debugger

Estos scripts se conocen como ***PyCommands***, y además de todos los que incluye Immunity Debugger por defecto, podemos escribir nuestros propios scripts para tareas puntuales, o modificar los ya existentes para que se ajusten a nuestras necesidades. Entre los PyCommands disponibles contamos con scripts para ocultar el debugger, búsqueda de rutinas de encriptación o detección de packers, búsqueda de instrucciones, definir breakpoints y muchas cosas más:

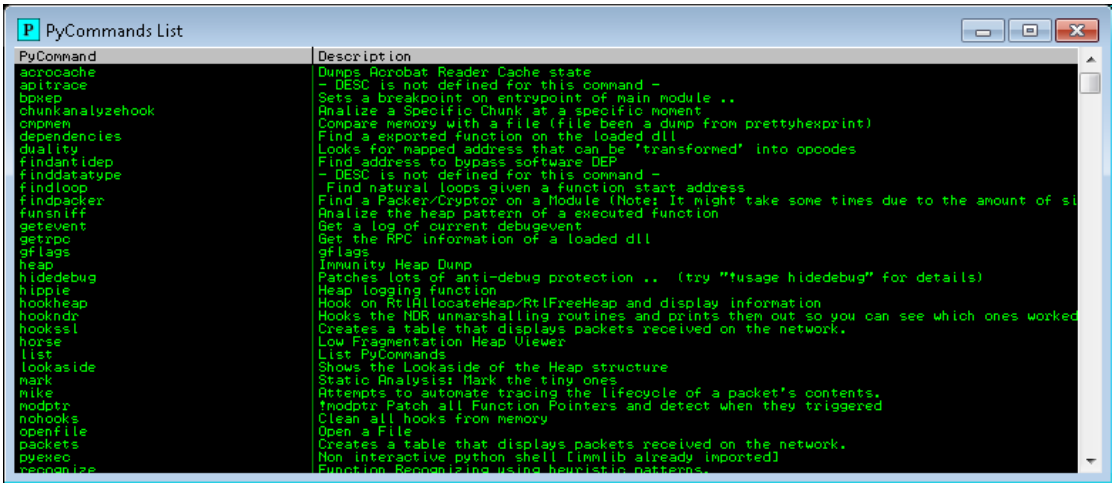

Imagen 2 - Listado de PyCommands

Además de estas funcionalidades, esta herramienta también permite, que a través de comando en Python, podamos definir las direcciones de memoria en las que queremos establecer los *breakpoints* y cosas por el estilo. Existen diferentes tutoriales[5] y manuales en Internet que nos ayudan a entender y escribir nuestros propios ***PyCommands***.

IDA

IDA[6] es uno de los desensambladores más potentes que existen en la actualidad. A partir de la versión 6 de IDA se incluyó directamente en el programa lo que se conoce como IDAPython, una interfaz completa que al igual que en Immunity Debugger, nos permite automatizar tareas o crear nuestros propios scripts para tareas puntuales de análisis.

[5] PythonHosted: http://pythonhosted.org/pycommand/
[6] IDA: https://www.hex-rays.com/products/ida/

Este desensamblador es ampliamente utilizado por profesionales de seguridad a lo largo del mundo para el análisis de códigos maliciosos y muchas otras cosas más. A través de su interfaz es posible realizar las más diversas acciones. A través de la carga de scripts y otras funcionalidades se pueden desencriptar secciones de código, analizar protocolos de comunicación o investigar la estructura de una amenaza:

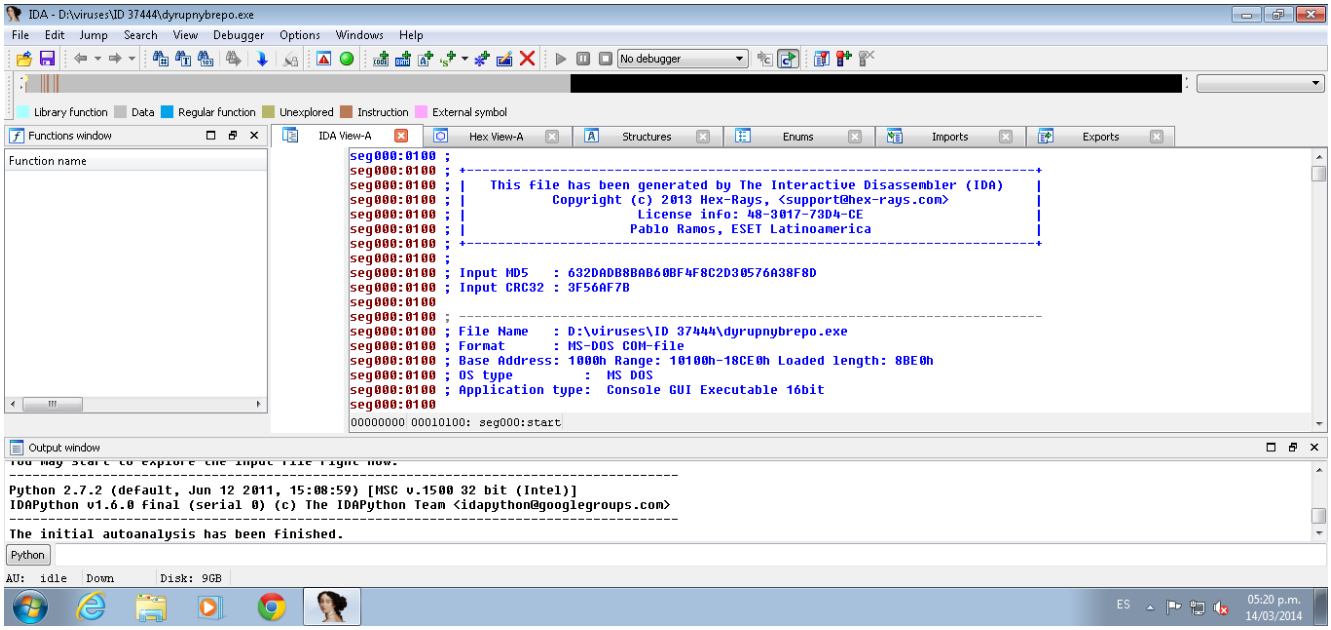

Imagen 3 - IDA

Si bien, el estudio en detalle de estas funcionalidades excede a nuestra introducción de Python, es más que claro que muchas herramientas de seguridad se basan en este lenguaje para su desarrollo o extensión de funcionalidades. La elección de especialistas en seguridad y desarrollo en este lenguaje para tareas de Ingeniería Inversa, Análisis de malware, Debugging y muchas otras más demuestras lo práctico y dinámico que es este lenguaje.

Otras aplicaciones

Además de las herramientas que mencionamos en el apartado anterior Python es utilizado en muchas otras aplicaciones. A continuación les dejamos un pequeño listado de herramientas o aplicaciones que han sido desarrolladas en Python y se utilizan en el mundo de la seguridad:

- **Scapy**: Una librería que para el análisis, inyección y captura de paquetes de red.[7]
- **Yara Python**: Una herramienta de clasificación y detección de capacidades de ejecutables. [8]
- **Androguard**: Un framework completo para análisis e Ingeniería Inversa de aplicaciones o malware para Android.[9]
- **Ssdeep**: Un wrapper en Python para el uso de la herramienta de SSDEEP para el cálculo de hashes sobre ejecutables o archivos[10].

Si quieren un listado más extenso de estas herramientas pueden encontrarlo en el siguiente enlace: http://dirk-loss.de/python-tools.htm

[7] Scapy: http://secdev.org/projects/scapy/
[8] Yara: https://code.google.com/p/yara-project/source/browse/trunk/yara-python/README
[9] Andoguard: https://code.google.com/p/androguard/
[10] SSDEEP: https://pypi.python.org/pypi/ssdeep/2.9-0.3

Conclusión

Llegando hacia el final del curso, fuimos aprendiendo cómo es que se construye este lenguaje de programación, vimos cómo crear nuestros programas en Python para satisfacer nuestras necesidades. Luego de entender cómo funciona el lenguaje y sus estructuras nos dedicamos a ver algunas de sus capacidades y librerías para poder crear nuestro propio código y acceder a archivos, bases de datos y sitios web.

Después de ver el lenguaje en sí hicimos un repaso de algunas herramientas que se utilizan en el mundo de la seguridad y en particular para el análisis de los códigos maliciosos. Ahora, les queda a ustedes la decisión de seguir profundizando sus conocimientos en este lenguaje y ver cuál es la utilidad que le quieren dar.

Pueden utilizar Python para analizar malware, crear sus propias herramientas, automatizar tareas de búsqueda y descarga de malware en Internet, extender la funcionalidad de otros programas o dedicarse a cualquier otra tarea de programación con este lenguaje. Python es una herramienta, que nos ayuda a lograr un objetivo, pero eso no es solo una característica de Python, sino una característica de la programación.

Saber programar es una gran ayuda para los especialistas en seguridad, no solo nos permite conocer cómo funcionan las cosas sino que también nos permite automatizar tareas, extender nuestras capacidades y reducir nuestros tiempos operativos. Ahora les queda a ustedes la responsabilidad de profundizar sus conocimientos, comentarnos acerca de qué herramientas quieren saber e investigar sobre cómo protegerse en Internet y cómo proteger al resto de los usuarios.