Ciberseguridad

17 de julio de 2025

José Luis

Índice general

1. Temario tentativo

Temario en orden cronológico: Wed 16 Jul 25

(1) Redes y protocolos

(2) Sistemas operativos

(3) Programación baja, vulnerabilidades y reversing

(4) Pentesting

(5) Hardening, blue team y forensics

(6) Criptografía

1.1. Algunos objetivos a mediano plazo

- Completar libros.
- Resolver máquinas o labs como `Hack The Box`, `PicoCFT`, `Root-Me`.
- Experimentar con laboratorios pequeños en casa: una red virtual con PfSense, Kali Linux, Window vulnerable, etc.
- Construir un portafolio de aprendizajes.
- Sacar las certificaciones eLearnSecurity Junior Penetration Tester v2 (`eJPTv2`), eLearnSecurity Certified Professional Penetration Tester v2 (`eCPPTv2`), eLearnSecurity Web Application Penetration Testing (`eWPT`), CompTIA Security +, Offensive Security Certified Professional (`OSCP`), `CISM` o `CISSP`.

2. Networking

2.1. Algunas definiciones básicas Thu 17 Jul 25

Cualquier dispositivo que se conecte a Internet se llamará **host** o **end system**.

3. Operating Systems

Existen tres conceptos clave para entender como funciona un sistema operativo, cómo decide qué programa ejecutar después en la CPU, cómo maneja la sobrecarga de memoria en un sistema de memoria virtual, cómo funcionan los monitores de máquinas virtuales, cómo manejar la información en discos o incluso cómo construir un sistema distribuido que funcione cuando algunas partes hayan fallado. Estos conceptos son los de **virtualización**, **concurrencia** y **persistencia**. Thu 17 Jul 25

3.1. Sistemas informáticos

El código máquina `x86-64` es el último en un camino de evolución seguido por Intel y su competencia que empezó con el procesador `8086` en 1978. Esta clase de procesadores se conoce coloquialmente como `x86`. Consideraremos como las máquinas ejecutan código `C` en Linux. Otros sistemas operativos que tienen herencia del sistema operativo Unix son Solaris, FreeBSD y MacOS X. Estos sistemas han mantenido un buen nivel de compatibilidad gracias a Posix y al Single UNIX Specification. Se puede usar Linux en un entorno de máquina virtual como VirtualBox o VMWare, que permiten ejecutar programas escritos para un sistema operativo (el "guest OS") en otro (el "host OS").

4. Vulnerabilities

4.1. Reverse Engineering

Thu 17 Jul 25

La ingeniería inversa es un conjunto de técnicas y herramientas para entender lo que hace un software en realidad. Formalmente, es "el proceso de analizar un sistema sujeto para identificar los componentes del sistema y su interrelación, así como crear una representación del sistema de una forma diferente o a un nivel mayor de abstracción." [3].

Bibliografía

[1] Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. *Operating Systems: Three Easy Pieces*. Arpaci-Dusseau Books, Inc., 2014. `http://www.ostep.org`.

[2] Randal E. Bryant and David R. O'Hallaron. *Computer Systems: A Programmers Perspective*. Pearson Education, 3 edition, 2016.

[3] Eldad Eilam. *Reversing: Secrets of Reverse Engineering*. Wiley Publishing, Inc., 2005.

[4] James F. Kurose and Keith W. Ross. *Computer Networking*. Pearson Education, 6 edition, 2013.