Vulnerabilidades de seguridad en sistemas embebidos

Patricia Martín, Sergio Martin, Gabriel Díaz y Manuel Castro-Gil
Universidad Nacional de Educación a Distancia (España)

DOI: http://dx.doi.org/10.6036/8026

1. INTRODUCCIÓN

En la actualidad es difícil imaginar la vida cotidiana sin utilizar dispositivos embebidos o empotrados puesto que están integrados perfectamente en el día a día [1]. La seguridad es un aspecto que cada vez está cobrando más importancia en el diseño de los sistemas embebidos (SSEE) tanto en la industria como en el ámbito académico [2]. Las características de los dispositivos hacen que sean inherentemente vulnerables a muchos problemas operacionales y ataques intencionados debido a su naturaleza "integrada" o "incrustada" [3]. El propósito de este estudio es poder ofrecer una visión global de las posibles vulnerabilidades a las que están expuestos estos dispositivos junto con algunas herramientas útiles que nos pueden ayudar a detectarlas para posteriormente poder actuar y protegernos.

MÉTODOS

Considerando las características de los sistemas operativos para SSEE existentes se procede a realizar el análisis con el sistema operativo Haiku. En la máquina atacante o analizadora, se instala la versión Kali Linux y en la máquina atacada o analizada se instala la versión del SO Haiku. Las fases a seguir para la realización del experimento son:

1. Instalación SO Haiku en máquina virtual objetivo.
2. Instalación SO Kali Linux en máquina analizadora.
3. Instalación y configuración de Metasploit y Nessus.
4. Búsqueda de vulnerabilidades. El orden de uso de las herramientas para la realización del análisis de vulnerabilidades es el siguiente: Metasploit, NMAP y Nessus.
5. Obtención de conclusiones y buenas prácticas de diseño.

RESULTADOS

Una vez instalados los sistemas operativos y las herramientas, la primera actividad realizada es un escaneo de los puertos abiertos en la máquina virtual Haiku a través de la herramienta metasploit. De este análisis se observa que se encuentran abiertos los puertos:

- **Puerto 21**: Puerto del protocolo de transferencia de archivos (FTP).
- **Puerto 22**: Servicio de *shell* seguro (SSH).
- **Puerto 23**: Servicio telnet.
- **Puerto 80**: Protocolo de transferencia de hipertexto (HTTP) para los servicios WWW.
- **Puerto 53**: Servicios de nombre de dominios DNS.
- **Puerto 445**: Bloque de mensajes de servidor (Server Message Block ,SMB) sobre TCP/IP.
- **Puerto 139**: Servicios de sesión NETBIOS utilizados en Red Hat Enterprise Linux por Samba.

Para complementar el análisis de metasploit se realiza un escaneo con la herramienta **nmap** para comprobar si nos devuelve la misma información que el escaneo anterior. Metasploit Framework ofrece una información más completa que nmap. En este caso descubre un puerto abierto más.

Además de los puertos abiertos detectados con Mestasploit Framework, con Nessus se detecta el **puerto TCP 1780** necesario para garantizar la entrega de paquetes de datos en el mismo orden que fueron enviados.

Problema potencial de seguridad detectado / servicios disponibles	Nessus	Metasploit Framework	Nmap
Puerto 21	✓	✓	✓
Puerto 22	✓	✓	✓
Puerto 23	✓	✓	✓
Puerto 53	✓	✓	
Puerto 80	✓	✓	✓
Puerto 139	✓	✓	✓
Puerto 445	✓	✓	✓
Puerto 1780	✓		
Información ampliada sobre servidor SSH	✓		
Información ampliada sobre DNS	✓		
Información ampliada sobre servicio telnet	✓		

Tabla 1: Resumen servicios detectados por herramientas

DISCUSIÓN Y CONCLUSIONES

Como conclusión final del estudio se muestra en la Tabla 1 un resumen de los servicios detectados por cada herramienta utilizada poniendo de manifiesto la importancia de cada una de ellas. Claramente se deduce que la herramienta más completa es Nessus, utilizándola de forma complementaria con Metasploit Framework.

A la hora de abordar la protección de los sistemas de control es fundamental empezar conociendo cuál es el estado o nivel de seguridad del mismo [4]. Se puede partir de de la información que pueda haber sobre las amenazas potenciales, vulnerabilidades típicas y los factores de riesgo. Sin embargo, es necesario ir más allá y conocer la situación de los sistemas en relación al negocio. Esto se consigue realizando auditorías (como por ejemplo, análisis de vulnerabilidades y test de intrusión) que nos permiten conocer de primera mano el estado de seguridad de que se dispone.

REFERENCIAS

[1] Berreteaga O. "Tendencias en productos con sistemas embebidos". ULMA Embedded Solutions. Abril 2011.
[2] Ravi S, Raghunathan A, Kocher P, et al. "Security in embedded systems: design challenges". ACM Transactions on Embedded Computing Systems 3(3):461–491, 2004.
[3] Wolf T. "Embedded Systems Security – an overview", Design Automation for Embedded Systems, Springer, September 2008, Volume 12, Issue 3, pp 173-183.
[4] Martin-Gutierrez, S., Martin, P., Diaz-Orueta, G., Castro-Gil, M.. (2016). VULNERABILITY ANALYSIS IN ELECTRONIC DEVELOPMENTS BASED ON EMBEDDED SYSTEMS . DYNA New Technologies, 3(1). 0. DOI: http://dx.doi.org/10.6036/NT7950.

AGRADECIMIENTOS

Los autores agradecen el soporte prestado por la Escuela de Ingenieros Industriales de la UNED en el proyecto 2016- IEE10 (AVANCES EN EL INTERNET DE LAS COSAS), así como los proyectos eMadrid (S2013/ICE-2715), IN-CLOUD (2015-1-IT01-KA202-00473) y Go-Lab (FP7-ICT-2011-8/317601).