IMPLEMENTACION DE UN LABORATORIO VIRTUAL PARA UNA ASIGNATURA DE SEGURIDAD INFORMATICA

Pamela Riquelme Estrada, Universidad del Bío-Bío, pvriquel@alumnos.ubiobio.cl
Patricio Galdames Sepúlveda, Universidad del Bío-Bío, pgaldames@ubiobio.cl
Claudio Gutiérrez Soto, Universidad del Bío-Bío, cogutier@ubiobio.cl

RESUMEN

Para un profesional del área de la informática, la adecuada aplicación de políticas de seguridad informática y de mecanismos que permitan su cumplimiento se han tornado en pilares fundamentales para el diseño e implementación de software de calidad. La principal dificultad para lograr un alto nivel de aprendizaje en seguridad informática se debe a que esta asignatura requiere integrar conocimientos previos adquiridos de diversas otras, tales como ingeniería de software, bases de datos, sistemas operativos, redes de computadoras y ética profesional. Además, en diversas instituciones educacionales no existe un ambiente propicio en el cual los alumnos puedan experimentar de forma práctica los contenidos impartidos. Este trabajo comprende el estudio del arte de diversas iniciativas desarrolladas para cursos de seguridad informática llevadas a cabo en diversas universidades. A partir de este estudio, se propone el diseño de un ambiente de laboratorio virtual escalable en el tiempo y de costo efectivo. Además, este trabajo presenta una serie de ejercicios prácticos que se han desarrollado a la fecha como también algunos que se incluirían a corto plazo. Por último se presentan diversas ideas para retroalimentar al docente con el estado del aprendizaje de sus alumnos particularmente destacando aquellos aspectos donde estos presenten mayores dificultades de aprendizaje.

PALABRAS CLAVES: Seguridad Informática, Virtualización, Habilidades de aprendizaje, Satisfacción estudiantil.

INTRODUCCIÓN

El mercado tecnológico en el mundo y particularmente en Chile sigue creciendo a un ritmo acelerado, especialmente el segmento de los usuarios de telefonía móvil quienes están interesados en acceder a servicios de voz y datos en la Internet. Lamentablemente en los últimos años, Chile se ha hecho integrante de la lista de los 10 mayores países que han sufrido más infecciones y ataques de seguridad (Neira, 2014). Esto se debe en parte a la falta de conocimientos de los resguardos que se deben considerar al construir y utilizar sistemas informáticos online para prevenir o mitigar la posible pérdida de confidencialidad, integridad y disponibilidad de nuestros datos.

Para los alumnos de carreras informáticas, es de gran importancia contar con las herramientas y los conocimientos necesarios para enfrentar las dificultades que se pueden generar en el desarrollo o administración de un sistema computacional. Ningún sistema por muy bien diseñado que esté, está exento de trasgresiones a su seguridad. Diversos estudios muestran que podemos encontrar en el software comercial un promedio de 0,5 bugs por cada 1000 líneas de código (Viega, 2002), y por ende ningún sistema es 100 % seguro. Este riesgo es cada día mayor, dado que cada día nos apoyamos más para realizar nuestro quehacer en un sistema informático. Por tanto, se requiere de profesionales que sean capaces de identificar estos riesgos informáticos y que luego propongan soluciones que prevengan o hagan muy difícil en complejidad y /o costo la posible explotación de estos riesgos.

En diversas instituciones de educación superior que ofrecen carreras ligadas a la informática, se incorpora en su currículo una asignatura de seguridad informática. En general, esta asignatura es impartida utilizando metodologías de aprendizaje centrados en la labor del docente, que si bien proporcionan al estudiante los conceptos necesarios para comprender un ataque informático, fallan al momento que el mismo estudiante aplique su conocimiento para prevenir o mitigar los efectos de este. Esto se debe en parte al deficiente aprendizaje de conocimiento previo que posee el estudiante sobre otras asignaturas de su carrera, tales como Ingeniería de Software, Sistemas Operativos y Redes de Computadoras.

En este trabajo se presenta el estado de desarrollo de una implementación de un laboratorio de seguridad informática que permita a los alumnos comprender las causas que pueden conformar una vulnerabilidad de software, que puedan explotarla y finalmente sean capaces de proponer mecanismos de seguridad que puedan detectarlas, prevenirlas o bien mitigar su efecto. Se presentaran una serie de experimentos prácticos desarrollados a la fecha para mejorar diversos conceptos presentados en una asignatura típica de seguridad informática. Un aspecto destacable de mencionar es que los experimentos desarrollados no tan solo mejoran el aprendizaje de los alumnos en la asignatura de Seguridad informática sino que estos también permitirán reforzar conceptos adquiridos de otras asignaturas relacionadas a su carrera.

Este articulo resume los principales resultados obtenidos en (Riquelme, 2016) y se divide en las siguientes secciones. Primero daremos una breve descripción de laboratorios de especialidad implementados en distintas instituciones académicas. Segundo describiremos que recursos computacionales de software y hardware hemos escogido para nuestra implementación. Luego, describiremos una serie de experimentos que se han desarrollado para presentar algunos temas de interés de la asignatura. Finalmente, daremos algunos comentarios finales.

REQUERIMIENTOS

En general las asignaturas de introducción a la Seguridad Informática cubren al menos las unidades que se indican en la tabla 1. Basado en este programa se propone diseñar e implementar un laboratorio que provea un ambiente en el cual los alumnos puedan realizar una serie de ejercicios prácticos, que les permita entender una vulnerabilidad de seguridad y luego que los mismos estudiantes propongan mecanismos que resuelvan o mitiguen las consecuencias de un ataque que aproveche tal vulnerabilidad.

Tal laboratorio debería ofrecer una plataforma de experimentación que facilite y refuerce:

- ***Instalación***: Que los ejercicios prácticos a desarrollar sea fácilmente instalables en cualquier ambiente computacional.
- ***Aislación***: El desarrollo de los ejercicios deben circunscribirse a los límites del ambiente del laboratorio, de modo de evitar que el desarrollo de ellos pueda interferir con actividades normales que se realicen en redes externas al laboratorio.
- ***Accesibilidad***: El ambiente del laboratorio no puede ser accesible remotamente, ya sea dentro del laboratorio mismo (a menos que se habilite explícitamente esta posibilidad), pero nunca desde fuentes externas a él.
- ***Monitoreo***: Los ejercicios prácticos deben permitir al docente o encargado del laboratorio el seguimiento de las actividades realizadas por los estudiantes con el fin de detectar dificultades de aprendizaje o eventualidades durante su desarrollo.

Unidad	Contenidos
Conceptos básicos	Definición de Seguridad Informática, Modelos, terminología
Criptografía	Criptografía de llave privada: DES, AES Criptografía de llave publica: RSA Funciones de Hashing, MD5, SHA-3 Certificados digitales y firmas digitales Control de Acceso
Seguridad de Software	Saturación de búfer, inyección de código, XSS, CSRF. Programación segura de software. Software malicioso: Virus, gusanos, troyanos, etc.
Seguridad de Redes	Cortafuegos y Sistemas de detección de intrusos (IDS) en redes. Protocolos de autenticación de usuarios remotos (SSH, SSL), túneles (IPSec, VPN)
Ética y Derecho Informático	Códigos de ética de entidades profesionales, legislación Chilena e Internacional.

Tabla 1. Programa típico de una asignatura de introducción a la seguridad informática

ESTUDIO DEL ARTE

Observando los requerimientos planteados en la sección anterior se realizo una búsqueda bibliográfica sobre proyectos de diseño e implementación de una plataforma de experimentación de seguridad informática para fines educativos (Anderson, 2006; Du, 2009; Mirkovic, 2011; Tripathi, 2010; Wang, 2013). A continuación se describen los proyectos más citados:

El primero proyecto existente corresponde a Xen Worlds (Anderson, 2006). Este software está desarrollado en base a un monitor de máquinas virtuales llamado Xen. Este monitor es un hipervisor de tipo 1 desarrollado originalmente por la Universidad de Cambridge. Es de código libre, razón por la cual ha sido utilizado como base para virtualización por diversas empresas, universidades y particulares desde que Amazon lo implementara como una solución para administrar sus servidores.

El proyecto utiliza como base el hipervisor para crear un banco de pruebas virtuales donde los estudiantes puedan configurar redes de máquinas virtuales (MVs) completamente funcionales a las que denomina un Xen Worlds. En la Universidad Estatal de Iowa (Rursch, 2013) se destinaron un máximo de 30 máquinas virtuales por curso al semestre, ejecutadas en un único computador con un procesador Pentium 4 y 2 GB de RAM. En cada computador pudo correr 30 máquinas simultáneamente sin que el rendimiento decreciera notoriamente

Dado que Xen Worlds está diseñado a partir de Xen, gran parte de sus ventajas vienen dadas por la arquitectura del hipervisor, el que permite crear un ambiente aislado, donde cada alumno al correr su MV, no es consciente del tráfico y de los programas ejecutados por sus compañeros. Esta característica es fundamental ya que se espera poder modificar y “romper

cosas" sin afectar el sistema que las aloja o el resto de los sistemas operativos (SO) virtualizados dentro del anfitrión.

En Xen Worlds el acceso a las máquinas virtuales se realiza por medio del protocolo ssh y los alumnos pueden iniciar, respaldar, eliminar y trasladar sus MVs mediante otras aplicaciones desarrolladas para estos efectos.

Otro proyecto desarrollado es SEED (SEcurity EDucation) Labs (Du, 2009). Este consiste en un repertorio de experimentos diseñados para un curso de seguridad informática que se implementan sobre una máquina virtual con el SO Ubuntu 12.04. Este SO tiene pre-instalado y configurado una serie de herramientas, servidores, vulnerabilidades y servicios para realizar de forma individual y con una configuración mínima los ejercicios planteados por cada modulo del proyecto SEED. Cada módulo consiste de un problema de seguridad que es explicado brevemente en un documento y los estudiantes deben intentar lograr los objetivos planteados por el modulo. Este proyecto utiliza un software de paravirtualización, es decir, donde el hipervisor corre como una aplicación sobre el sistema operativo anfitrión para ejecutar una o más imágenes de la máquina virtual.

Actualmente SEED ha desarrollado más de 30 laboratorios de seguridad que abordan problemáticas de desarrollo de software, de redes de computadores, de sistemas operativos y también de criptografía. Estos laboratorios requieren de una o más máquinas virtuales y en ocasiones deben utilizar también el sistema operativo Minix 4.

La gran ventaja de este proyecto es que permite que los laboratorios puedan ser fácilmente instalados y por tanto permite a los estudiantes enfrentar el problema planteado en pocos minutos ya que la máquina virtual de Ubuntu tiene instalada todas las aplicaciones necesarias. El alumno sólo tiene que descargar la imagen, montarla sobre un software de virtualización y seguir los pasos indicados en un documento que describe los objetivos del laboratorio.

La principal desventaja de SEED es que hace uso de la paravirtualización y aunque tanto VMware como VirtualBox cuentan con motores que optimizan las llamadas a los recursos físicos, sigue siendo una aplicación dentro de un sistema operativo. Es el SO anfitrión el que controla los recursos, por lo que tener más de tres máquinas virtuales corriendo en un computador con un procesador de menos de 2 GHz y 2 GB de RAM empobrece notoriamente el rendimiento del sistema. Se recomienda dedicar sólo 512 MB de RAM, pero en ocasiones el sistema requiere de un mínimo de 1024 MB para correr las aplicaciones con las cuales se trabajará.

Otro proyecto similar es DETERLab (Benzel, 2006), que consiste de una plataforma de cientos de servidores sobre los que se ha virtualizado prácticamente todo tipo de hardware y software. Esta disponible de forma gratuita a cualquier persona que desee hacer uso de ella. DeterLab está basado en Linux y fue creado para enseñar a sus alumnos el diseño de sistemas operativos en la Vrije Universiteit de Ámsterdam. Su principal objetivo es proveer un ambiente seguro sobre el cual los profesionales del área de seguridad cibernética puedan desarrollar, descubrir y probar tecnologías, software, configuraciones de red, sistemas operativos, etc.

Se accede de forma remota y por medio de una cuenta online, la cual se crea completando un formulario donde se especifican los motivos para los cuales se utilizará DETERLab. Se debe adjuntar también una autorización o certificado de la institución o empresa a la que pertenece quien solicita la cuenta. Si bien desde un cierto punto de vista, parece ser una solución factible, la falta de mecanismos de monitoreo de los alumnos nos lleva a descartar esta opción.

Otro proyecto similar a Xen World es Sunyit (Bull, 2012). Emplea un hipervisor de Xen mejorado, llamado Xen Cloud Platform, que incluye un modo de administración grafico local y excluye un par de características mininas que posee el Xen original. Sunyit usó Xen Cloud Platform para virtualizar tres servidores con una capacidad de 16 GB de RAM y un procesador Intel Zeon Quad Core. Como medida de seguridad fraccionó una red en subredes. Los alumnos se conectan de forma remota a la plataforma por medio de una aplicación web creada con java.

Finalmente la última alternativa analizada son los Offensive Security PenetrationTesting Virtual Labs (Offensive Security, 2016). Esta es una plataforma en línea de laboratorios virtuales que cuentan con una amplia variedad de laboratorios diseñados por la empresa Offensive Security. Esta alternativa cuenta con alrededor de 40 máquinas con múltiples sistemas operativos. Estas maquinas son empleadas para simular diversos escenarios reales.

Para acceder a los laboratorios se debe crear una cuenta y pagar el registro. La cuenta permite tomar una serie de cursos disponibles en la página web y conectarse a los servidores por medio de una conexión VPN, lo que implica que los estudiantes deben contar con una conexión rápida. Cabe destacar que Offensive Security es una empresa que ha ganado popularidad por ser quien ha desarrollado el sistema operativo Kali Linux. Este SO tiene instalado por defecto más de 600 aplicaciones para realizar diversos tipos de ataques, pruebas y auditorías de seguridad informática y además está disponible de forma gratuita.

En (Riquelme, 2016) se realiza un análisis mas detallado de la las características y limitaciones de los proyectos presentado y sugiere que si se desea priorizar rendimiento y escalabilidad Xen Worlds es una buena opción. Esto porque es un ambiente basado en un hipervisor tipo uno, que permitiría virtualizar un número mayor de máquinas virtuales en un mismo ambiente sin sacrificar en demasía el rendimiento. Además, permite mantener aislados los experimentos del resto de las instalaciones del laboratorio. En comparación con soluciones comerciales similares como VMware, XenWorlds es una solución de bajo costo y gran rendimiento. Sin embargo si la prioridad es el costo, la mejor opción es Virtual Box. Esto es porque es un software disponible gratuitamente para los principales SOs y cumple con funciones similares al del hipervisor tipo 1.

Si bien estamos trabajando en adaptar Xen Worlds a los propósitos de nuestro laboratorio, se decidió inicialmente preparar experimentos del tipo SEED sobre Kali Linux y VirtualBox.

EXPERIMENTOS

Se realizó un estudio de los contenidos de cada unidad de la tabla 1 en relación a los ataques más comúnmente usados para penetrar sistemas seleccionando los siguientes.

1. *Ataque de fuerza bruta:* Corresponde a la acción de recuperar una contraseña o clave a través de la prueba de cada una de las posibles combinaciones de claves hasta dar con aquella correcta. (Wikipedia, Ataque de Fuerza Bruta, 2016)

2. Ataque de Inyección SQL: Es un tipo de ataque donde se logra ejecutar sentencias SQL maliciosas que permiten obtener acceso no autorizado a datos privados almacenados en una base de datos SQL. (Cid, 2014)

3. *Ataque de Denegación de Servicio (DOS, Denial of Service):* consiste en crear un gran flujo de tráfico de red el cual es encauzado hacia un servidor objetivo con la intención de sobrecargarlo para ralentizar su servicio o forzarlo a desconectarse de la red. La idea es

evitar que usuarios legítimos puedan acceder a través de la red al objetivo. (Jacobson, 2008).

4. *Ataque del Hombre en el Medio (Man-in-the-Middle):* El atacante es capaz de intervenir la comunicación entre dos entidades de modo tal que estas no son conscientes de la existencia del interventor. Este atacante podría ya sea leer, insertar y/o modificar a voluntad, los mensajes transmitidos entre las dos entidades. (Stamp, 2011)

5. *Ataque de Saturación de Búfer (Buffer Overflow):* Este ataque consiste en sobrescribir el puntero de retorno de una función vulnerable e identificada en un programa, con el fin de alterar el curso de ejecución normal del programa. La idea es identificar un arreglo o búfer que emplee la función para el cual no se chequea el tamaño de los datos que pudieran ser cargados en ella (Aleph1, 1996).

Para realizar los laboratorios, cada estudiante debe tener acceso a una red que consista de tres maquinas virtuales cuyos detalles se muestran en la tabla 2. Para los requerimientos se utilizo como base, aquellos recomendados por el desarrollador del SO.

	MV 1	**MV 2**	**MV3**
Sistema Operativo	Ubuntu 14.04TrustyJar	Kali Linux 64 bits	Ubuntu Server 14.04
Hardware Virtualizado	1CPU 3.9 GHz 1 GB RAM Red Interna 12 MB video	1CPU 3.9 GHz 1 GB RAM Red Interna 12 MB video	1CPU 3.9 GHz 1 GB RAM Red Interna 12 MB video
Particiones	300 MB ext2 /boot 15 GB ext3 /root 2 GB swap 65 GB ext4 /home	300 MB ext2 /boot 15 GB ext3 /root 2 GB swap 65 GB ext4 /home	100 MB ext2 /boot 15 GB ext3 /root 2 GB swap 50 GB ext4 /home

Tabla 2: Maquinas Virtuales empleadas en los Laboratorios

En las MV se instalaron las siguientes aplicaciones:

MV1	**MV2**	**MV3**
Webmin	John The Ripper	Apache
Php5	DVWA	Mysql
Mysql	Apache2	Php5
DVWA (modificado)	Mysql	FTP
phpmyadmin	Php5	Webmin
Apache	Ettercap	Phpmyadmin
	Wireshark	SSH
	hydra	DVWA
	Goldeneye	OpenSSH
	Tor	
	Crunch	
	Nmap	

Tabla 3: Software instalado en cada máquina virtual

Para el *ataque de fuerza bruta,* los estudiantes deben descubrir la contraseña de acceso a un servidor FTP. Para ello, los estudiantes deben crear una lista de usuarios y contraseñas con el

software Crunch (Liatsisfotis, 2013) y luego realizar un ataque de fuerza bruta con el programa Hydra (Van Hauser, 2016) al servidor FTP.

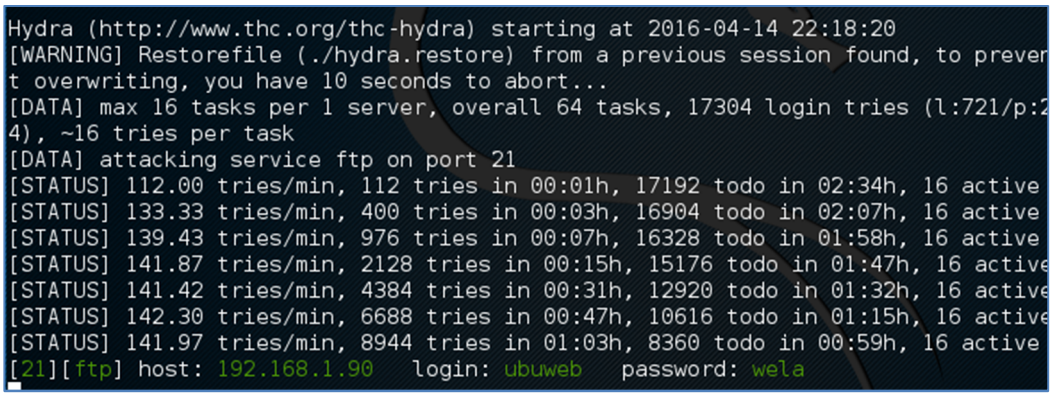

Figura 1: Ataque de fuerza bruta con Hydra.

Para realizar *el ataque de inyección SQL*, los estudiantes deben ejecutar la aplicación DVWA (Fpalenzuela, 2014), la cual permite rápidamente configurar y poner online un servidor web y una base de datos MySQL que presenta la vulnerabilidad que permite la ejecución exitosa de este ataque. Los alumnos deben intentar acceder a través de la interfaz web acceder a la tabla *information_schema* para descubrir detalles de las tablas de la base de datos y luego descubrir los usuarios y contraseñas de acceso almacenadas en la misma base de datos. En MySQL, las contraseñas son protegidas con MD5, por lo que a los estudiantes se les sugiere emplear John The Ripper (Rm-rf.es, 2014) (ataque de fuerza bruta o de diccionario) para descifrar estas contraseñas.

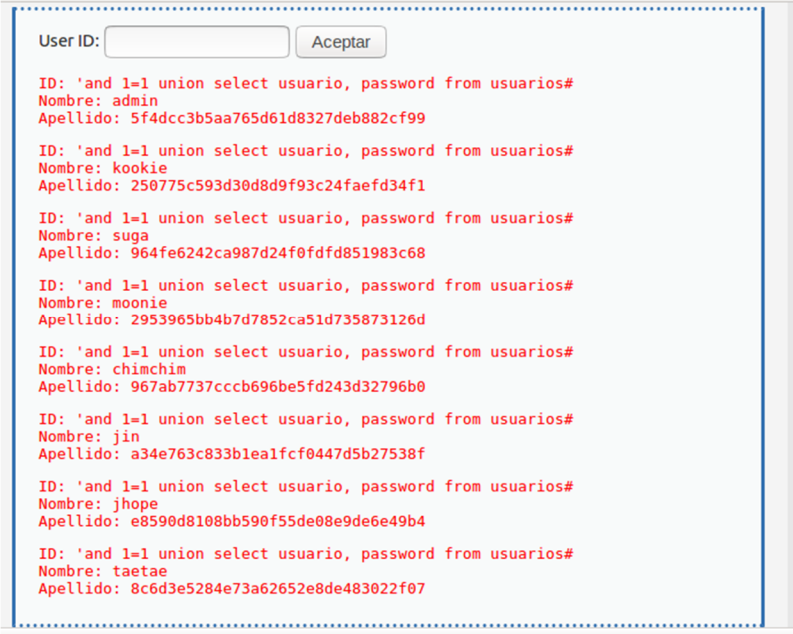

Figura 2: Ataque de inyección SQL

Para probar un ataque de denegación de servicio, los alumnos deben configurar un servidor web y desde otra máquina virtual ejecutar la aplicación Goldeneye (Wroot, 2016), la que permite generar intentos de conexión TCP al servidor web objetivo. Para observar al aumento de tráfico

en la red, los alumnos emplean la aplicación EtherApe (Toledo, 2016), la cual permite monitorear de forma gráfica la efectividad del ataque.

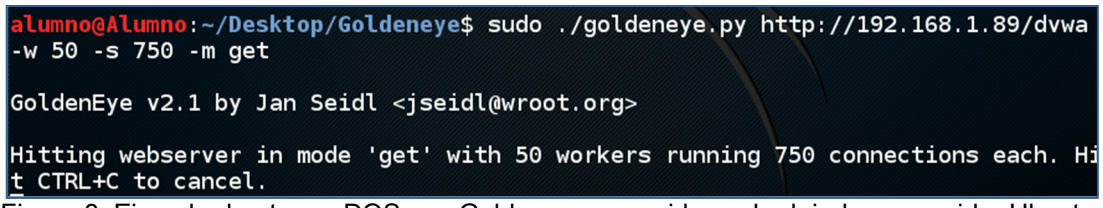

Figura 3: Ejemplo de ataque DOS con Goldeneye a servidor web alojada en servidor Ubuntu.

Como un ejemplo de un ataque de hombre en el medio, los estudiantes deben realizar un ataque de falsificación ARP (ARP Spoofing y Sniffing) en una red virtual. Esta red consiste de tres MVs, de las cuales dos simulan a entidades legitimas que intercambian datos y la tercera corresponde a la maquina del atacante, quien intenta “envenenar” o modificar las tablas ARP de las dos primeras utilizando el programa Ettercap (Ettercap Project, 2016). Posteriormente el spoofing en la red se realiza con Wireshark (Wireshark Foundation, 2016).

Para ejecutar un ataque de saturación de búfer. Los alumnos deben intentar utilizar exitosamente un exploit previamente creado para generar un buffer overflow haciendo uso de otro código que contiene una consola o shell. Este shell es almacenado dentro de un arreglo que al “saturarse” permite sobrescribir la dirección de retorno de la función ejecutada y obtener una terminal con acceso root.

Figura 4: Ataque de saturación de Búfer

CONCLUSIONES

A corto plazo estamos preparando una serie de rubricas para evaluar la efectividad del aprendizaje de los alumnos al realizar los laboratorios, para identificar problemas en la ejecución de estos y para revisar posibles eventualidades que puedan surgir. Diversos trabajos (Mirkovic, 2011; Wang, 2013) sugieren que el “aprendizaje haciendo (learning by doing)” es una metodología adecuada para que los alumnos asimilen conceptos complejos que requieren una diversidad de conocimiento previo como los problemas que surgen en la seguridad informática. Sin duda este esfuerzo no sera eficaz sino se le facilita al docente de un mecanismo que le apoye en el rápida detección de problemas en la realización de los laboratorios. Es por ello que estamos trabajando en un sistema de monitoreo que permita realizar un seguimiento a las actividades que realiza un estudiante. En estos momentos estamos capturando el log de las actividades que realiza un alumno al realizar un ataque de saturación de búfer desde un

terminal. Estos logs son almacenados en una base de datos externa de modo de permitirle al profesor un análisis posterior. Esperamos en el futuro incluir el monitoreo de los restantes y nuevos ataques.

AGRADECIMIENTOS

Este trabajo fue parcialmente financiado por el proyecto de investigación Nº DIUBB 135515 3/RS de la Universidad del Bío-Bío y por el Departamento de Sistemas de Información de la misma Universidad.

REFERENCIAS

- Aleph1 (1996). Smashing The Stack For Fun And Profit. Phrack Magazine Nº 49, Vol 7, Aug 8th.
- Anderson B. y Daniels T. (2006). Xen Worlds: Xen and the Art of Computer Engineering Education. Proceedings of the 2006 ASEE Annual Conference and Exposition.
- Benzel, T., Braden, R., Kim, D., Neuman, C., Joseph, A., Sklower, K. y Schwab, S. (2006). Experience with deter: a testbed for security research. In 2nd International IEEE Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006.TRIDENTCOM 2006.pp. 10-pp.
- Bull, R. L. (2012). Design and Implementation of Computer Science Virtualized Lab Environment at SUNYIT. MSc Thesis dissertation, State University of New York Institute of Technology.
- Cid, D. (2014). Sucuri Blog. Retrieved from Ataques web – Inyección SQL Y La Amenaza Que Representan: https://blog.sucuri.net/espanol/2014/10/08/ataques-web-inyeccion-sql-y-la-amenaza-que-rrepresentan.html
- Du W. (2011) The SEED Project: Providing Hands-on Lab Exercises for Computer Security Education. In the IEEE Security and Privacy Magazine, September/October. Invited paper.
- Ettercap-Project (2016). Retrieved from https://ettercap.github.io/ettercap/
- Fpalenzuela. (2014). Como instalar DVWA (Damn Vulnerable Web App) en Kali linux. Retrieved from Aprendiz de sysadmin: http://aprendizdesysadmin.com/como-instalar-dvwa-damn-vulnerable-web-app-en-kali-linux/
- Jacobson D. (2008). Introduction to Network Security, Chapman & Hall/CRC Computer and Information Science Series, 1st Edition.
- Liatsisfotis. (2013). Create Wordlists using Crunch. Retrieved from liatsisfotis.com: http://www.liatsisfotis.com/2013/02/create-wordlists-using-crunch.html
- Mirkovic J., Ryan M., Hickey J., Sklower K.,Reiher P., Peterson P., B. Hoon Kang, ChooChuah M., Massey D. y Ragusa G. (2011) Teaching Security With Network Testbeds, Proceedings of the ACM SIGCOMM Workshop on Education.
- Neira, J. (2014). EMOL. Retrieved from
- http://www.emol.com/noticias/tecnologia/2014/04/10/654699/experto-en-seguridad-informatica-indica-que-chile-esta-en-el-top-10-de-paises-infectados.html
- Offensive Security (2016), Retrieved from
- https://www.offensive-security.com/offensive-security-solutions/virtual-penetration-testing-labs/
- Riquelme P. (2016). Implementación de un Laboratorio de Seguridad Informática para Uso Educacional. Proyecto de Titulación para la Obtención del Titulo Profesional en Ingeniería Civil Informática, Departamento de Sistemas de Información, Universidad del Bío-Bío sede Concepción.

- Rursch, J. y Jacobson D. (2013). When a testbed does more than testing.The Internet-Scale Event Attack and Generation Environment (ISEAGE) – providing learning and synthesizing experiences for cyber security students, Proceeding of the 2013 IEEE Conference on Frontiers in Education (FIE), pp. 1267-1272.
- Rm-rf.es,(2014, Octubre). John the Ripper: password cracking en Linux. Retrieved from http://rm-rf.es/john-the-ripper-password-cracking-linux/
- Stamp M. (2011). Information Security: Principles and Practice. 2nd Ed. Wiley.
- Toledo J., y Ghetta R. (2016). EtherApe. Retrieved from http://etherape.sourceforge.net/
- Tripathi A., Mishra A. y Raghunathan G. V. (2010). A Test Bed for Information Security Skill Development with Virtual Training Environment. Proceedings of the 2010 Information Security for South Africa (ISSA)
- Van Hauser (2016), THC Hydra. Retrieved from https://www.thc.org/thc-hydra/
- Viega J. y McGraw G. (2002) Building Secure Software, Addison Wesley.
- Vigna G. (2003) Teaching Hands-On Network Security: Testbeds and Live Exercises. Journal of Information Warfare Vol 3, No. 2, 2003, pp. 8-25.
- Wang, X., Hong, X., Li, T., Gao, B., Wang, Z., y Luo, Y. (2013). pVEE: A Personalized Virtualized Experimentation Environment for Education Based on Virtual Machines. In Joint International Conference on Pervasive Computing and the Networked World, pp. 620-631. Springer International Publishing.
- Wroot. (2016). GoldenEye. Retrieved from http://wroot.org/projects/goldeneye
- Wireshark Foundation (2016). Wireshark. Go Deep. Retrieved from https://www.wireshark.org/