jamk.fi

Virtual learning environment for penetration testing

Jarmo Puttonen

Thesis
March 2015

Degree programme in Software Engineering
The School of Technology

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

DESCRIPTION

Author(s)	Type of publication	Date
Puttonen, Jarmo	Bachelor´s Thesis	7.3.2015
	Pages 69	Language English
		Permission for web publication (X)

Title
VIRTUAL LEARNING ENVIRONMENT FOR PENETRATION TESTING

Degree Programme
Degree Programme in Software Engineering

Tutor(s)
Salmikangas, Esa

Assigned by
Jyväskylä Security Technology
Niemelä, Antti

Abstract
The thesis was assigned by the cyber security related JYVSECTEC project of JAMK University of Applied Sciences. JYVSECTEC wanted to get a laboratory environment suitable for the training and evaluation of penetration testing skills. This laboratory was to be created inside their cyber security research, education and development environment. During the thesis writing process, the assignment was expanded to include a tracking software and a Moodle platform with a tracking software-integration. The tracking software was required to be usable for tracking users' progress inside multiple different penetration testing laboratories and also in large scale cyber security exercises.

The theory section of the thesis focuses on explaining Penetration Testing Standard describing the structure of penetration testing, OWASP-project's Top10 vulnerability categories of the year 2013, Kali Linux operating system designed for penetration testing, and some of the existing environments similar to the created laboratory environment.

The result of the thesis was a virtual learning environment that can be used in the way specified by JYVSECTEC. When a student advances inside the created laboratory environment, the tracking software tracks this progress. The created Moodle-plugin downloads this tracking data automatically and based on this data Moodle instructs and grades the progress of the user automatically. The tracking software and Moodle-plugin are capable of tracking multiple different users inside multiple different laboratory and exercise environments simultaneously. The laboratory environment was created to model a computer network of a fictional middle sized company. The laboratory environment contains multiple different vulnerabilities on various different operating systems, and it can be used efficiently as a scenario-based exam or as a target for training and testing usage of penetration testing tools.
The created laboratory environment can be used in further development as a template for creating new environments. The tracking software can be easily integrated for different purposes.

Keywords
JYVSECTEC, Cyber security, Penetration testing, Vulnerabilities, Exploits, Virtual learning environment, Capture-the-flag, Realistic Global Cyber Environment, Virtualization.

Miscellaneous
Due to a request from JYVSECTEC, all documentation related to the results of the thesis was attached as appendices to the thesis.

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

KUVAILULEHTI

Tekijä(t) Puttonen, Jarmo	Julkaisun laji Opinnäytetyö	Päivämäärä 7.3.2015
	Sivumäärä 69	Julkaisun kieli Englanti
		Verkkojulkaisulupa myönnetty (X)

Työn nimi
VIRTUAALINEN OPPIMISYMPÄRISTÖ PENETRAATIOTESTAUKSEEN

Koulutusohjelma
Ohjelmistotekniikan koulutusohjelma

Työn ohjaaja(t)
Salmikangas, Esa

Toimeksiantaja(t)
Jyväskylä Security Technology
Niemelä, Antti

Tiivistelmä
Työn tavoitteena oli luoda Jyväskylän ammattikorkeakoulun IT-instituutin kyberturvallisuusteknologia-aiheisen JYVSECTEC-projektin kehittämään kyberturvallisuuden tutkimus-, koulutus- ja kehitysympäristöön laboratorio, jota voidaan käyttää penetraatiotestaustaitojen kouluttamisessa sekä näiden taitojen osaamisen arvioinnissa. Työn tavoitteet laajenivat toteutuksen aikana sisältämään seurantaohjelmiston sekä tälle integraation Moodle-oppimisalustaan. Seurantaohjelmistolla tuli olla mahdollista seurata käyttäjien etenemistä erilaisissa penetraatiotestaukseen liittyvissä laboratorioympäristöissä sekä esimerkiksi kyberturvallisuusharjoituksissa.

Työn teoriaosuus keskittyy kuvaamaan penetraatiotestauksen rakenteen määrittelevän Penetration Testing Standardin, OWASP-projektin vuoden 2013 Top10-haavoittuvuuskategoriat, penetraatiotestaukseen suunnitellun Kali Linux käyttöjärjestelmän ja joitain kehitettyä laboratorioympäristöä vastaavia olemassaolevia ympäristöjä.

Työn tuloksena toteutettu virtuaalinen oppimisympäristö toimii toimeksiantajan haluamalla tavalla kokonaisuutena niin, että opiskelijoiden edetessä laboratorioympäristössä seurantaohjelmisto välittää opiskelijoiden etenemistiedot Moodlelle, joka automatisoidusti ohjaa opiskelijoita sekä pisteyttää opiskelijoiden osaamista seurantatietojen perusteella. Moodle-ympäristöllä ja tämän seurantaohjelmistointegraatiolla pystytään seuraamaan samanaikaisesti useita erilaisia laboratorio- ja harjoitusympäristöjä. Luotu laboratorioympäristö toteutettiin mallintamaan fiktiivisen keskisuuren yrityksen tietoverkkoa. Laboratorioympäristö sisältää lukuisia haavoittuvuuksia useissa eri käyttöjärjestelmissä ja sitä pystytään käyttämään sellaisenaan skenaariotyyppisenä kokeena tai penetraatiotestaustyökalujen koulutuksessa ja testaamisessa.

Jatkokehityksessä luotua laboratorioympäristöä pystytään käyttämään pohjana uusien ympäristöjen luomisessa ja seurantaympäristö on helposti integroitavissa erilaisiin käyttökohteisiin.

Avainsanat (asiasanat)
JYVSECTEC, Kyberturvallisuus, Haavoittuvuudet, Haittakoodi, Penetraatiotestaus, Oppimisalusta, Lipunryöstö, Realistic Global Cyber Environment, Virtualisointi.

Muut tiedot
Toimeksiantajan pyynnöstä kaikki työn tuloksiin liittyvä dokumentaatio lisättiin opinnäytetyön liitteiksi.

CONTENTS

FIGURES

GLOSSARY

Active directory	”Microsoft's directory service database for Windows networks. Stores information about resources on the network and provides a means of centrally organizing, managing, and controlling access to the resources. – –” (Mueller, R. 2013)
AES-encryption	“– – A U.S. government-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. – –” (Kissel, R. 2013)
Asset	“A major application, general support system, high impact program, physical plant, mission critical system, personnel, equipment, or a logically related group of systems.” (Mueller, R. 2013)
Application Programming Interface (API)	“An application programming interface (API) is a software program that facilitates interaction with other software programs.” “An API allows a programmer to interact with an application using a collection of callable functions. The goal of an API is to allow programmers to write programs that will not cease to function if the underlying system is upgraded.” (What is Application Programming Interface (API) - Definition from Techopedia)
Base64 encoding	“Base64 is an encoding and decoding technique used to convert binary data to an American Standard for Information Interchange (ASCII) text format, and vice versa. It is used to transfer data over a medium that only supports ASCII formats, such as email messages on Multipurpose Internet Mail Extension (MIME) and Extensible Markup Language (XML) data.” (What is Base64? - Definition from Techopedia)
Bourne Again Shell (Bash)	“Bourne again shell (Bash) is a free Unix shell that can be used in place of the Bourne shell. It is a complete implementation of the IEEE Portable Operating System Interface for Unix (POSIX) and Open Group shell specification.” (What is Bourne Again Shell (Bash) - Definition from Technopedia)
Batch file	“A specific script file format (.bat) containing one MS-DOS command on each line of the file. When run, each line executes in sequential order. Batch files run on Microsoft-compatible operating systems, such as DOS, OS/2, and Windows.” (McAfee Threat Glossary)
Blacklist	“A list of discrete entities, such as hosts or applications, that have been previously determined to be associated with malicious activity.” (Mueller, R. 2013)
Block cipher	“A symmetric key cryptographic algorithm that transforms a block of information at a time using a cryptographic key. For a block cipher algorithm, the length of the input block is the same as the length of the output block.” (Mueller, R. 2013)

Botnet	"Definition: A collection of computers compromised by malicious code and controlled across a network." (Explore Terms: A Glossary of Common Cybersecurity Terminology)
Brute-force attack	"A method of accessing an obstructed device through attempting multiple combinations of numeric and/or alphanumeric passwords." (Mueller, R. 2013)
Buffer	"An area of memory reserved for temporarily holding data before that data is used by a receiving device or application. Buffering protects against the interruption of data flow." (Microsoft Terminology Collection)
Buffer overflow	"A condition at an interface under which more input can be placed into a buffer or data holding area than the capacity allocated, overwriting other information. Attackers exploit such a condition to crash a system or to insert specially crafted code that allows them to gain control of the system." (Mueller, R. 2013)
Buffer overflow attack	"A method of overloading a predefined amount of space in a buffer, which can potentially overwrite and corrupt data in memory." (Mueller, R. 2013)
CamelCase	"Camelcase is a naming convention for writing file or object names using compounded or joined words with at least of those words beginning in a capital letter. Camelcase is used in programming language to name different files and functions without violating the naming laws of the underlying language." (What is Camelcase - Definition from Techopedia)
Capture-the-flag	"CTFs (short for capture the flag) are a type of computer security competition. Contestants are presented with a set of challenges which test their creativity, technical (and googling) skills, and problem-solving ability. Challenges usually cover a number of categories, and when solved, each yields a string (called a flag) which is submitted to an online scoring service. CTFs are a great way to learn a wide array of computer security skills in a safe, legal environment, and are hosted and played by many security groups around the world for fun and practice." (picoCTF 2014 - Frequently Asked Questions)
Clear text	"Information that is not encrypted." (Mueller, R. 2013)
Comma-Separated Values File (CSV)	"A comma separated values (CSV) file contains different values separated by a delimiter, which acts as a database table or an intermediate form of a database table. In other words, a CSV file file is a set of database rows and columns stored in a text file such that the rows are separated by a new line while the columns are separated by a semicolon or a comma. A CSV file is primarily used to transport data between two databases of different formats through a computer program." (What is Comma-Separated Values File (CSV) - Definition from Techopedia)
Common Gateway	"Common Gateway Interface (CGI), in the context of web development, is an interface for running executables via a web-server. In most intances, this means taking an HTTP request and passing it to an application in order to deliver a

Interface (CGI)	dynamically-generated HTML page back to a browser. While pretty much any program that can run on a web server is usable as a CGI script, Perl is the most popular language.“ (What is Common Gateway Interface (CGI) - Definition from Techopedia)
Common Vulnerabilities and Exposures (CVE)	“A dictionary of common names for publicly known information system vulnerabilities.” (Mueller, R. 2013)
Cookie	”A piece of state information supplied by a Web server to a browser, in a response for a requested resource, for the browser to store temporarily and return to the server on any subsequent visits or requests.” (Mueller, R. 2013)
CSS	”CSS defines the concept of style where style defines appearance. Each and every tag of HTML can be modified by associating a particular style with it. The concept of style optimizes the performance of HTML content rendering, and simplifies developers' work by uniformly rendering a style specification across different browser implementations. CSS consists of selectors and declarations. The declaration is a combination of property and value.” (What is Cascading Style Sheets Level 1 (CSS1) - Definition from Techopedia)
CVE Identifier	An identifier that has been designated to a known vulnerability. See Common Vulnerabilities and Exposures (CVE).
Cyber domain	See cyberspace.
Cyber Exercise	“Definition: A planned event during which an organization simulates a cyber disruption to develop or test capabilities such as preventing, detecting, mitigating, responding to or recovering from the disruption.” (Explore Terms: A Glossary of Common Cybersecurity Terminology)
Cyber security	”The ability to protect or defend the use of cyberspace from cyber attacks.” (Mueller, R. 2013)
Cyber attack	”An attack, via cyberspace, targeting an enterprise’s use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information.” (Mueller, R. 2013)
Cyberspace	”A global domain within the information environment consisting of the interdependent network of information systems infrastructures including the Internet, telecommunications networks, computer systems, and embedded processors and controllers.” (Mueller, R. 2013)
Data breach	”Definition: The unauthorized movement or disclosure of sensitive information to a party, usually outside the organization, that is not authorized

	to have or see the information." (Explore Terms: A Glossary of Common Cybersecurity Terminology)
Demilitarized Zone (DMZ)	"Perimeter network segment that is logically between internal and external networks. Its purpose is to enforce the internal network's Information Assurance policy for external information exchange and to provide external, untrusted sources with restricted access to releasable information while shielding the internal networks from outside attacks." (Mueller, R. 2013)
DHCP	"Acronym for Dynamic Host Configuration Protocol. Service that provides centralized control of Internet Protocol (IP) addresses. DHCP servers assign dynamic IP addresses and TCP/IP settings to other computers." (Mueller, R. 2013)
Document Object Model (DOM)	"Document Object Model (DOM) is a language and platform-independent convention that represents the interaction of objects written in markup languages, i.e., Hypertext Markup Language (HTML), Extensible Hypertext Markup Language (XHTML) and Extensible Markup Language (XML)." (What is Document Object Model (DOM) - Definition from Techopedia)
Domain controller	"A server with Active Directory installed. A domain controller (DC) is authoritative for the domain to which the server is joined. It contains the Active Directory database for the domain namespace, plus the Configuration and Schema namespaces for the forest." (Mueller, R. 2013)
Environment variable	"Environment variables are values that impact the processes and behavior of running computer systems and OS environments. Running programs may access environment variable values for configuration purposes." (What is Environment Variable - Definition from Techopedia)
Exploit code	"A program that allows attackers to automatically break into a system." (Mueller, R. 2013)
False Positive	"An alert that incorrectly indicates that malicious activity is occurring." (Mueller, R. 2013)
Garbage collection	"Garbage collection (GC) is a dynamic approach to automatic memory management and heap allocation that processes and identifies dead memory blocks and reallocates storage for reuse. The primary purpose of garbage collection is to reduce memory leaks." (What is Garbage Collection (GC) - Definition from Techopedia)
Hardening	"Configuring a host's operating systems and applications to reduce the host's security weaknesses." (Mueller, R. 2013)
Intranet	"A private network that is employed within the confines of a given enterprise (e.g., internal to a business or agency)." (Mueller, R. 2013)
Intrusion Detection	"Hardware or software product that gathers and analyzes information from various areas within a computer or a network to identify possible security

Systems (IDS)	breaches, which include both intrusions (attacks from outside the organizations) and misuse (attacks from within the organizations.)" (Mueller, R. 2013)
IP Security (IPsec)	"Suite of protocols for securing Internet Protocol (IP) communications at the network layer, layer 3 of the OSI model by authenticating and/or encrypting each IP packet in a data stream. IPsec also includes protocols for cryptographic key establishment." (Mueller, R. 2013)
JSON	"JavaScript Object Notation (JSON) is an open standard data exchange format based on a JavaScript syntax subset. JSON is text-based, lightweight, and generally considered easily readable/writeable." (What is JavaScript Object Notation (JSON)
LDAP	"Acronym for Lightweight Directory Access Protocol. A language based on the X.500 directory standard that allows clients and servers to communicate. The LDAP provider allows access to the hierarchical structure of Active Directory, or any LDAP compliant database. The LDAP syntax is a filter syntax used to query LDAP compliant databases." (Mueller, R. 2013)
LMHash	"The LAN manager hash (LANMAN hash) is an encryption mechanism implemented by Microsoft prior to its release of NTLM. The LANMAN hash was advertised as a one-way hash that would allow end users to enter their credentials at a workstation, which would, in turn, encrypt said credentials via the LANMAN hash." (What is LAN Manager Hash (LANMAN Hash) - Definition from Techopedia)
Localization	"creation of a national or specific regional version of a product." (ISO/IEC/IEEE 24765 Systems and software engineering — Vocabulary. 2010)
Lun-number	"A logical unit number (LUN) is a number used for identifying a logical unit relating to computer storage. A logical unit is a device addressed by protocols and related to fiber channel, small computer system interface (SCSI), Internet SCSI (iSCSI) and other comparable interfaces." (What is Logical Unit Number (LUN) - Definition from Techopedia)
Malicious Code	"Software or firmware intended to perform an unauthorized process that will have adverse impact on the confidentiality, integrity, or availability of an information system. A virus, worm, Trojan horse, or other code-based entity that infects a host. Spyware and some forms of adware are also examples of malicious code." (Mueller, R. 2013)
Man-in-the-middle Attack (MitM)	"An attack on the authentication protocol run in which the Attacker positions himself in between the Claimant and Verifier so that he can intercept and alter data traveling between them." (Mueller, R. 2013)

MD5	”Short for Message Digest 5. A computer algorithm that calculates a unique number, called a hash value, when it receives a string of data, as in a text or EXE file. When compared to the original file, a hash value shows if the file has been changed.” (McAfee Threat Glossary)
Dump	“A duplicate of a program, a disk, or data, made either for archiving purposes or for safeguarding files.” (Microsoft Terminology Collection)
Metadata	”Information about the properties or structure of data that is not part of the values the data contains” (Microsoft Terminology Collection)
Metasploit	”Consider the MSF to be one of the single most useful auditing tools freely available to security professionals today. From a wide array of commercial grade exploits and an extensive exploit development environment, all the way to network information gathering tools and web vulnerability plugins. The Metasploit Framework provides a truly impressive work environment. The MSF is far more than just a collection of exploits, it's an infrastructure that you can build upon and utilize for your custom needs. This allows you to concentrate on your unique environment, and not have to reinvent the wheel.” (Introduction - Metasploit Unleashed)
Meterpreter	”Meterpreter is an advanced, dynamically extensible payload that uses in-memory DLL injection stagers and is extended over the network at runtime. It communicates over the stager socket and provides a comprehensive client-side Ruby API. It features command history, tab completion, channels, and more. Metepreter was originally written by skape for Metasploit 2.x, common extensions were merged for 3.x and is currently undergoing an overhaul for Metasploit 3.3.” (About the Metasploit Meterpreter)
Microsoft NTLM	”Windows NT LAN Manager (NTLM) is a security protocol suite for Microsoft Windows NT 4.0. NTLM replaced Windows LAN Manager (LANMAN). NTLM is used for down-level client and server compatibility up to Windows 2000.” (What is Windows NT LAN Manager (NTLM))
Mode of operation	“An algorithm for the cryptographic transformation of data that features a symmetric key block cipher algorithm.” (Mueller, R. 2013)
Network Sniffing	”A passive technique that monitors network communication, decodes protocols, and examines headers and payloads for information of interest. It is both a review technique and a target identification and analysis technique.” (Mueller, R. 2013)
Overhead time	”the amount of time a computer system spends performing tasks that do not contribute directly to the progress of any user task” (ISO/IEC/IEEE 24765 Systems and software engineering — Vocabulary. 2010.)
Payload	”The “cargo” code in a virus rather than the portions used to avoid detection or replicate. The payload code can display text or graphics on the screen, or it may corrupt or erase data. Not all viruses contain a deliberate payload. However, these codes affect CPU usage, hard disk space, and the time it takes to clean

	viruses. Payload can also refer to the data or packets sent during an attack." (McAfee Threat Glossary)
Penetration Testing	"Security testing in which evaluators mimic real-world attacks in an attempt to identify ways to circumvent the security features of an application, system, or network. Penetration testing often involves issuing real attacks on real systems and data, using the same tools and techniques used by actual attackers. Most penetration tests involve looking for combinations of vulnerabilities on a single system or multiple systems that can be used to gain more access than could be achieved through a single vulnerability." (Mueller, R. 2013)
Private Key	"Definition: A cryptographic key that must be kept confidential and is used to enable the operation of an asymmetric (public key) cryptographic algorithm." "Extended Definition: The secret part of an asymmetric key pair that is uniquely associated with an entity." (Explore Terms: A Glossary of Common Cybersecurity Terminology)
Privilege escalation	"A privilege escalation attack is a type of network intrusion that takes advantage of programming errors or design flaws to grant the attacker elevated access to the network and its associated data and applications." (Rouse, M. 2010.)
Proxy	"A proxy is an application that "breaks" the connection between client and server. The proxy accepts certain types of traffic entering or leaving a network and processes it and forwards it. This effectively closes the straight path between the internal and external networks making it more difficult for an attacker to obtain internal addresses and other details of the organization's internal network. Proxy servers are available for common Internet services; for example, a Hyper Text Transfer Protocol (HTTP) proxy used for Web access, and a Simple Mail Transfer Protocol (SMTP) proxy used for email." (Mueller, R. 2013)
Public key	"Definition: A cryptographic key that may be widely published and is used to enable the operation of an asymmetric (public key) cryptographic algorithm." "Extended Definition: The public part of an asymmetric key pair that is uniquely associated with an entity and that may be made public." (Explore Terms: A Glossary of Common Cybersecurity Terminology)
Regular expression (regex)	"A regular expression (sometimes abbreviated to "regex") is a way for a computer user or programmer to express how a computer program should look for a specified pattern in text and then what the program is to do when each pattern match is found." (Rouse, M. 2006.)
Root-user	The superuser account found in many Unix-based operating systems.
rot13	A letter substitution cipher in which the clear text letter is replaced by the letter that is 13 letters after the original in the alphabet.
Security through obscurity	"Security through obscurity (STO) is a process of implementing security within a system by enforcing secrecy and confidentiality of the system's internal design architecture. Security through obscurity aims to secure a system by deliberately

	hiding or concealing its security flaws." (What is Security Through Obscurity (STO) - Definition from Techopedia)
Shell	"The command interpreter that is used to pass commands to the operating system." (Microsoft Terminology Collection)
SMB-protocol	"Server Message Block (SMB) is an application-layer network protocol that facilitates network communication while providing shared access to client files, printers and serial ports." (What is Server Message Block (SMB) - Definition from Techopedia)
Social engineering	"A general term for attackers trying to trick people into revealing sensitive information or performing certain actions, such as downloading and executing files that appear to be benign but are actually malicious." (Mueller, R. 2013)
Spoofing	"1. Faking the sending address of a transmission to gain illegal entry into a secure system. Impersonating, masquerading, piggybacking, and mimicking are forms of spoofing. 2. The deliberate inducement of a user or resource to take incorrect action." (Mueller, R. 2013)
Standard output	"Standard output, sometimes abbreviated stdout, refers to the standardized streams of data that are produced by command line programs (i.e., all-text mode programs) in Linux and other Unix-like operating systems." (Standard Output Definition.)
Strcpy	C Standard library function for copying given string to a destination in the random access memory.
Stress testing	"Stress testing refers to the testing of software or hardware to determine whether its performance is satisfactory under any extreme and unfavorable conditions, which may occur as a result of heavy network traffic, process loading, underclocking, overclocking and maximum requests for resource utilization." (What is Stress Testing - Definition from Techopedia)
Structured Query Language (SQL)	"Structured Query Language (SQL) is a standard computer language for relational database management and data manipulation. SQL is used to query, insert, update and modify data. Most relational databases support SQL, which is an added benefit for database administrators (DBAs), as they are often required to support databases across several different platforms." (What is Structured Query Language (SQL) - Definition from Techopedia)
Sudo	"Sudo (su "do") allows a system administrator to give certain users (or groups of users) the ability to run some (or all) commands as root while logging all commands and arguments. Sudo operates on a per-command basis, it is not a replacement for the shell." (Sudo in a Nutshell)
Threat agent	"Definition: An individual, group, organization, or government that conducts or has the intent to conduct detrimental activities." (Explore Terms: A Glossary of Common Cybersecurity Terminology)

Threat Source	"The intent and method targeted at the intentional exploitation of a vulnerability or a situation and method that may accidentally trigger a vulnerability. Synonymous with Threat Agent." (Mueller, R. 2013)
Transport Layer Security (TLS)	"An authentication and security protocol widely implemented in browsers and Web servers." (Mueller, R. 2013)
Truecrypt	"TrueCrypt is software for establishing and maintaining an on-the-fly-encrypted volume (data storage device). On-the-fly encryption means that data is automatically encrypted right before it is saved and decrypted right after it is loaded, without any user intervention. No data stored on an encrypted volume can be read (decrypted) without using the correct password/keyfile(s) or correct encryption keys. Entire file system is encrypted (e.g., file names, folder names, contents of every file, free space, meta data, etc)." (TrueCrypt - Introduction)
Whitelist	"A list of discrete entities, such as hosts or applications that are known to be benign and are approved for use within an organization and/or information system." (Mueller, R. 2013)
Virtual machine	"A virtual machine (VM) is a software program or operating system that not only exhibits the behavior of a separate computer, but is also capable of performing tasks such as running applications and programs like a separate computer. A virtual machine, usually known as a guest is created within another computing environment referred as a "host." Multiple virtual machines can exist within a single host at one time." (What is Virtual Machine (VM)
Virtual Private Network (VPN)	"A virtual network, built on top of existing physical networks, that provides a secure communications tunnel for data and other information transmitted between networks." (Mueller, R. 2013)
Vulnerability	"Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source." (Mueller, R. 2013)

1 INTRODUCTION

1.1 JYVÄSKYLÄ SECURITY TECHNOLOGY

Jyväskylä Security Technology (from now on referred to as “JYVSECTEC”) is a facility that specializes in research, training and development of cyber security. JYVSECTEC project was founded by JAMK University of Applied Sciences, and it is partially funded by the Regional Council of Central Finland and the European Regional Development Fund. Project was started in September 2011, and it is financed until January 2015. (JYVSECTEC – Jyväskylä Security Technology)

The objective of the project is to promote JYVSECTEC to one of Finland’s leading cyber security development and training facilities and to create a co-operation network between central Finnish companies and actors in the field of cyber security. JYVSECTEC fulfills its objective in the field of cyber security by developing and maintaining Realistic Global Cyber Environment and Cyber Security Situation Center, providing an environment, services for organizing cyber exercises and operational business models for local and global co-operation networks, acting as an independent community for exchanging and sharing information, taking part in preparation of new international research projects and by mapping opportunities related to cyber security in Central Finland. (JYVSECTEC – Jyväskylä Security Technology)

1.2 REALISTIC GLOBAL CYBER ENVIRONMENT

Realistic Global Cyber Environment (from now on referred to as "RGCE") is an environment produced by JYVSECTEC. It was developed to be an isolated representation of the Internet as closely as possible. RGCE contains multiple different internet service providers that have their own services including domain name services, IP-subnets based on real Internet ISPs with realistic geolocation information and network time protocol servers. The network traffic of the RGCE is routed realistically through multiple points between clients connected to these ISPs. RGCE also contains some of the public services available on the Internet including news sites, discussion forums, TOR onion routing network, video streaming services, instant messaging services, photo galleries and repositories for serving updates and installation media to operating systems and software (e.g. CentOS software repository). Figure 1 visualizes some of the services provided by the RGCE. (RGCE – Realistic Global Cyber Environment)

Figure 1 Services of Realistic Global Cyber Environment. (RGCE – Realistic Global Cyber Environment)

Realistic network traffic on the RGCE is generated by botnet-like software developed and maintained by JYVSECTEC. This software generates automatically non-malicious network traffic to RGCE simulating real end-user network traffic. Non-malicious traffic can be e.g. browsing websites and communicating in instant messaging networks. The software can also be used to produce realistic cyber-attacks on predetermined targets. The structure of RGCE also allows capturing, analyzing, visualizing and retransmitting network traffic in any part of the environment. (RGCE – Realistic Global Cyber Environment)

RGCE is based on a virtual cloud environment provided by VMware's vCloud-environment which means that most of its infrastructure is virtualized and users can create their own networks (containing e.g. switches, routers, servers, and workstations) inside virtual datacenters efficiently without introducing new physical hardware to environment. This also allows handling of malicious software without risk of infecting users own physical devices. (RGCE – Realistic Global Cyber Environment)

RGCE is connected to Cyber Security Situation Room in JAMK Dynamo equipped with computers, displays, projectors, touch screens and it is being actively developed further. Cyber Security Situation Room allows effective usage of RGCE for training, exercise, research, development and testing purposes. The situation room can also be used for experimenting and

developing different kind of visualizations of cyber situation pictures. (RGCE – Realistic Global Cyber Environment)

1.3 Assignment and objectives

The objective of the original assignment (see Appendix 1) was to produce an environment where JYVSECTEC can teach and evaluate the skills of using penetration testing tools and techniques. The requirements specified that the environment should include multiple different vulnerabilities that are located in multiple different network services and that it should be possible to exploit these vulnerabilities in many different ways. The environment was also required to be as license free as possible (e.g. software and operating systems), so that it can be used as a commercial product or service if needed. More detailed specifications and objectives were shaped during the production of the environment in multiple conversations with JYVSECTEC. In these conversations, JYVSECTEC gave suggestions and wishes about the possible functionalities that could be implemented in the environment. The decision to include or not to include these functionalities was left to the author of the thesis who based the decision on an estimation of the feature's relevance and implementation effort. Also during the production of the environment, some of these decisions were changed due to new possibilities and limitations encountered in the process. Consequently, the process of producing the environment was very agile and adaptive. Below are listed the final requirements and objectives of the environment(s) that were specified as mandatory by JYVSECTEC.

The assignment specified three different entities that were to be produced:

- A vulnerable laboratory environment that models a realistic information network of a small to medium-size enterprise.
- A "capture-the-flag"-styled tracking software that can be used to track users' progress inside different laboratory environments.
- A virtual learning platform that can be used to guide and grade user's progress in different laboratory environments.

The objectives for the laboratory environment were:

- Create an environment inside the RGCE using only virtual machines.
- Create a realistic network structure for the environment.
- Use multiple different Windows- and Unix-based operating systems.

- Add realistic devices and services for the environment.
- Configure and install realistic vulnerabilities to the services and operating systems of the environment.
- Design the environment so that it can be easily used in training.
- Design the environment to be scenario-like.
- Document the environment in very close detail.

The objectives for the tracking software were:

- Create a software that can be used to follow users' progress inside laboratory-environments.
- Design the tracking software so that it can be used simultaneously on multiple different laboratory environments by multiple different users.
- Design the tracking software so that it can be used as a flag in capture-the-flag-styled cyber security exercises.
- Design the tracking software so that it provides basic functionality against possible cheating attempts.

The objectives for the learning platform were:

- Create a learning platform that can be used to distribute penetration testing studying materials to students and create theoretical exams that can be graded for each student.
- Create an integration for the tracking software so that the learning platform can automatically follow the users' progress inside different laboratory environments.
- Create a functionality that guides users' automatically by using the tracking software integration.
- Create a functionality that automatically grades users' by using the tracking software-integration.

The resulting source code was specified to be stored on JYVSECTEC's Gitlab source code hosting service, the resulting virtual machines and virtual machine networks were specified to be saved as snapshots inside the RGCE's vCloud environment. Information about installation, configuration, and usage of these entities was to be documented in very close detail. The documentation was specified to be provided by this thesis and its appendices and the comments inside the produced

source codes. Due to the nature of these entities, some of the detailed information in the thesis cannot be published in the public version.

2 THEORETICAL BACKGROUND

2.1 PENETRATION TESTING

May 5, 2013 cyber-attack resulted in loss of more than one million US dollars for Washington state hospital, **October 10, 2013** Adobe confirmed a security breach which resulted in leaking identities of 150 million accounts, **November 11, 2013** Target Corporation confirmed a data breach that resulted in leaking credit card information of 110 million accounts, **December 12, 2013** 105 million South Korean credit card details were stolen in a security breach. These examples represent only a small fraction of the large cyber-attacks in 2013. (Internet Security Threat Report 2014.)

Every day thousands and thousands of computer systems are being attacked in the cyber domain. Targets of these attacks range from small insignificant websites to huge government and organization computer networks. Not all of these attacks are malicious, threat agents can be divided into friendly and hostile cyber agents. Friendly cyber agents do not try to cause harm to the target; they usually conduct these attacks for research purposes or to increase security of the target systems. Hostile cyber agents can aim to cause direct/indirect harm to targets or to achieve personal profit. Figure 2 shows a multitude of the most significant threat agents in 2013. (European Union Agency for Network and Information Security.)

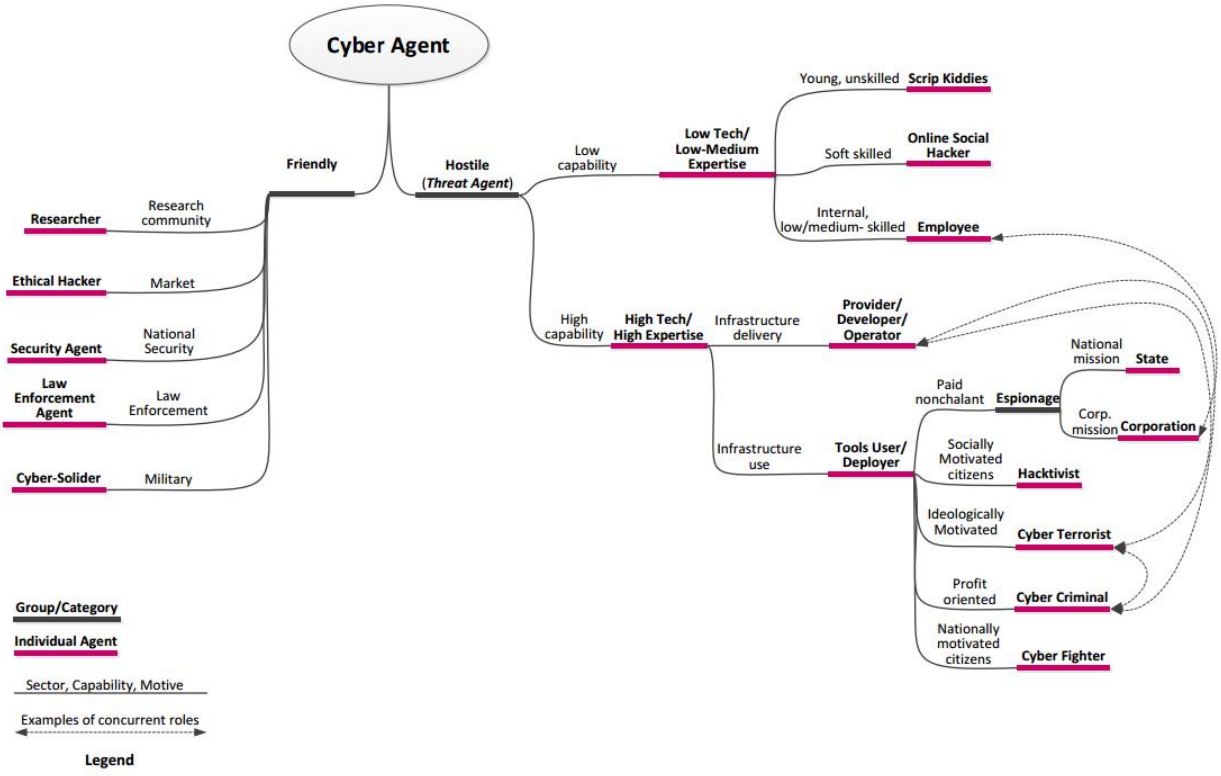


Figure 2 Overview of Agents in Cyber Space. (ENISA Threat Landscape 2013)

To counter hostile attacks, organizations are investing in cyber security more than ever before. This same trend is also visible on the opposing side as the hostile agents are getting more and more skilled and capable of attacking wider range of targets. The number of cyber-attacks that resulted in the disclosure or potential exposure of data is at an all-time high. A rough estimate is that the number of data breaches in year 2013 is ten times the amount of breaches in 2008. (M-Trends 2014 Threat Report; 2014 Data Breach Investigations Report (DBIR).)

Penetration testing can be carried out as a part of improving the cyber security situation. Penetration testing is a security testing type in which a tester tries to attack and exploit the target computer system in a way that uses the same techniques that the hostile cyber agents use in attacks. These attacks are allowed and authorized by the owner of the target computer system for the purpose of finding vulnerabilities in the environment and making it more secure. (Engebretson, 2014, 1-3.)

A vulnerability can be defined as a weakness inside a software or a device that can abused to affect the reliability, integrity and/or availability of the software or device. (Definition of a Security Vulnerability.). A vulnerability can be categorized as a known vulnerability or as a ”zero-day” vulnerability. The known vulnerabilities concentrated on certain devices or software versions and are either fixed in the newest version of the device/software or there is a fix available that removes

the vulnerability. These vulnerabilities are also often widely analyzed and have a proof-of-concept exploit that prove the exploitability of the vulnerability. The “zero-day” vulnerability is a vulnerability that has not been fixed and has a proof-of-concept exploit and/or is being exploited. “Zero-day” vulnerabilities are usually revealed after they have been used in an attack or when the finder of the vulnerability publishes it before notifying the manufacturer. (Internet Security Center – Definitions.)

Penetration testing can be executed as a ‘white box’ or as a ‘black box’-testing. The white box-penetration testing aims to assess the level of security in the targeted system in very close detail. To produce this close detail security assessment, the friendly agent tries to find every possible vulnerability in every part of the target system using all methods possible. The full details of the target system can be provided to a friendly agent before starting testing and target network security persons can be notified of the testing. The black box-penetration testing simulates the real hostile agents’ actions more closely. In the black box penetration testing, friendly agent tries to stay undetected and exploit the target through the easiest way possible. This type of testing does not produce wide image of the level of security in the target system, but it provides more accurate model of the actions of the possible hostile agents. (Engebretson, 2014, 4.)

The penetration testing can be carried out with a structure created from multiple different sections. The laboratory environment created in this thesis was specified to support the structure of the Penetration Testing Execution Standard (from now on referred to as “the PTES”). The laboratory environment was designed and produced so that every section of the PTES can be carried out in the environment one way or another. The PTES is a standard created by multiple different professional companies and individuals in the field of cyber security. It was designed to produce a common starting point of carrying out a penetration test for companies and cyber security service providers.

The PTES divides the penetration testing into a seven different sections that are:

1. Pre-engagement interactions.
2. Intelligence gathering.
3. Threat modeling.
4. Vulnerability analysis.
5. Exploitation.
6. Post-exploitation.
7. Reporting.

These sections will be explained in the following theory sections. The word "customer" will be used to refer to the client who requested a penetration test. (The Penetration Testing Execution Standard.)

Pre-engagement interactions

The pre-engagement interactions section acts as a definition phase for the penetration test. In this section, the objectives of the penetration test should be defined by finding out why the customer wants the environment to be tested and what the customer wants as a result of the penetration test. The length of the penetration test should be defined by for example with a starting and ending date. The scope of the test should be defined as well, which means that the tester should find out which devices and software of the target will or will not be tested and these devices and software should be defined using for example domains or IP addresses. Also, the existence of the possible indirect targets should be investigated, these can be for example a firewall or a network device between the tester and the target to be tested. A time estimate should be defined for each of the phases of the penetration test. A padding can be added to these time estimations. This padding can forgive the mistakes made in the time estimation or unknown roadblocks that might appear during the test. Possible involvement of third-parties in the test should be investigated. Third-parties could appear for example in a situation where the targets of the test are located in a third-party cloud environment. In this situation, the authorization for executing the test should also be asked from the third-party. Limits of the possible social engineering should be defined. For example, the tester can bribe the employees of the customer in order to acquire sensitive information and that type of social engineering could not be wanted nor accepted by the customer. The limits of possible stressing of the target environment should be defined so that there are no unwanted slowdowns. Both parties should agree on the payment terms of the penetration test before starting it. Communication channels used for communicating with the customer should be specified, and also the amplitude and frequency of communication should be defined. All possible rules related to the penetration test should be specified, these rules can include, for example, how to handle the penetration test results and in which physical location the test will be carried out. This section can also define how the targets response to different sections of the penetration test will be tested and documented. The PTES offers a comprehensive question list, which can be used to execute a satisfactory pre-engagement interaction. (Pre-engagement - The Penetration Testing Execution Standard.)

Intelligence gathering

Intelligence gathering is a phase, where all possible intelligence related to the penetration test and its targets is gathered. This intelligence is used for planning vulnerability assessment and exploitation phases. Intelligence can be gathered in physical and electrical locations of the target and also, for example, from employees of the targeted customer. How widely and thoroughly this phase is executed relates straightly to the ease of planning the strategies and entry vectors of the next phases. Before starting the intelligence gathering, the targets of the intelligence gathering should be determined by compliance to the rules and scope determined in the pre-intelligence-gather phase. Also, the objective of the intelligence gathering should be determined in close detail so that the time available is not wasted on unimportant targets. (Intelligence Gathering - The Penetration Testing Execution Standard.)

The PTES divides intelligence gathering into three different levels. These levels can be used when the objectives and resources of the intelligence gathering are being defined. Level 1 intelligence gathering is executed by using automated tools. The objective of the level is to gather automatically intelligence specified by the used compliance rating. In Level 2 intelligence gathering, manual analysis is carried out in addition to the gathering specified on Level 1. The tester who executes Level 2 information gathering should have basic knowledge of the customer and its business domain so that the tester can find and analyze detailed information effectively. On Level 3, there is much precise manual analysis of the information in addition to Level 1 and Level 2 information gathering. The tester must have a wide and profound understanding of the customer and their business domain. In Level 3 information gathering, the tester can, for example, even get new human relations only for the purpose of gathering information. Information gathering can be carried out in multiple different ways, the PTES explains three different ways that are: open source intelligence gathering or OSINT, covert gathering and foot printing. (Intelligence Gathering - The Penetration Testing Execution Standard.)

In the OSINT, information is gathered from the publically available sources and the gathered information will be refined to intelligence usable in later phases. The PTES divides OSINT to three different types: passive information gathering, semi-passive information gathering, and active information gathering. In the passive information gathering, only sources for the gathering are the ones that can be used without causing any visible direct or indirect traffic to the target of the penetration test. This type of gathering is useful for example in a situation where the customer has required that the target should not be able to detect any information gathering or when the penetration test is being done as black boxed. In this situation, the targets of the information gathering should be selected very carefully, so that for example the subdomains of the target or

other domains redirecting to the target will not be used. In the semi-passive information gathering there can be visible traffic to the target but it should be similar to the normal traffic of the target. For example, when gathering information from the target's website semi-passively, only the publically available content can be gathered and any content not directly available to the normal user should not be used or even tried to be located. Active information gathering is a strong and thorough investigation of the target. Traffic generated by this type of gathering can be detected at least as suspicious and abnormal. For example, a tester can try to survey the target network and its services by using multiple different vulnerability and network service scanners. (Intelligence Gathering - The Penetration Testing Execution Standard.)

The types of information gathered in OSINT can be divided into following categories: physical information (e.g. office locations of the customer), logical information (e.g. competitors and clients of the customer), organization chart, electrical information (e.g. meta-data of the publically files in company web services), financial information, assets related to the infrastructure of the customer and human-based information (e.g. criminal history of the customers employees). The PTES details closely many subsections of these types. Each of these subsections is assigned to an information gathering level in which it should be used. (Intelligence Gathering - The Penetration Testing Execution Standard.)

In the covert gathering, information is gathered by physically watching or affecting the target without getting detected. For example, covert gathering can be done by visiting the physical location of the customer with a faux reason. During this visit all information related to e.g. security checks, security devices and behavior of the employees is gathered imperceptibly. Covert gathering can also include e.g. searching the trashcans near the customer's office for sensitive information or sniffing the wireless traffic of the customer's office from outside the perimeter. (Intelligence Gathering - The Penetration Testing Execution Standard.)

The foot printing can be used to expand or detail the scope of the penetration test. In foot printing, information can be gathered by e.g. connecting to the devices and services of the target. The foot printing can be either external or internal. External foot printing can be divided into active and passive foot printing executed from outside the target network. In the passive external foot printing, for example, domains belonging to target can be foot printed from public domain name records. In the active external foot printing for example open network ports and public services can be scanned, non-public websites can be tried to found by conducting a brute-force search and banner-messages sent by public services can be collected. When the passive and active

external foot printing has been executed, the gathered information can be detailed by e.g. surveying and analyzing software versions of the services. Internal foot printing is possible if the tester has access inside the target network. In this situation the tester can do foot printing similar to the external foot printing and in addition can e.g. listen to network traffic created by users or devices related to network infrastructure. Listening to the network traffic is useful because it can reveal e.g. structure of the network, services of the network and information about its users. Result of the foot printing can be used to select the devices and services of the scope. (Intelligence Gathering - The Penetration Testing Execution Standard.)

The PTES defines that in the intelligence gathering phase, all protection systems of the following categories should be recognized and surveyed in as close detail as possible:

1. Network-based protection systems.
2. Host-based protection systems.
3. Application-level protection systems.
4. Data storage protection systems.

Network-based protection systems can include e.g. filtering of network packets, devices that modify network traffic and different kinds of encryption and tunneling methods. Host-based protection systems act by e.g. protecting the heap- and stack-memory of the device from unauthorized usage, allowing usage of only the whitelisted applications or by analyzing the behavior of the device. Application-level protection systems can be e.g. encoding used in the application or whitelist-filtering of a web application's user inputs. Data storage protection systems can be e.g. masqueraded a LUN-number of an iSCSI-device or CHAP-authentication used in an iSCSI-device. A user level protection system can be e.g. anti-virus which protects user from storing and running malicious executables. (Intelligence Gathering - The Penetration Testing Execution Standard.)

Threat modeling

In the threat modeling phase, all possible threats towards the target environment should be surveyed. The PTES defines that this surveying should be done as high-level, where first all the relevant information is gathered and after that primary and secondary assets are to be identified. Finally, the possible threat communities should be identified, and their relation to the primary or secondary assets should be determined. (Threat Modeling - The Penetration Testing Execution Standard.)

First step of the thread modeling is an analysis of the business assets. In this step all the information gathered in the previous phases is used to determine which the assets of the customer are, which of these are the most likely targeted by threat agents and what the consequences of losing these assets are. Business assets can include for example information about the internal procedures of the customer, confidential research and business information, marketing plans, financial information, network infrastructure, sensitive information about employees or clients and ”human assets” which are the persons critical to the wellbeing of the customer. (Threat Modeling - The Penetration Testing Execution Standard.)

The second step is the analysis of the business process. This contains the processes that produce money to the customer. The analysis contains also the processes that support the infrastructure crtitical to the business process. These can be, for example, the technical infrastructure, the information assets, the persons critical to the business process and the possible functionalities provided by a third-person. (Threat Modeling - The Penetration Testing Execution Standard.)

The third step of the threat modeling is an analysis of the possible threat agents and communities. This step uses the information gathered in previous phases to determine which agents have the capabilities and the motivations to attack the target. These threat agents can be divided into internal and external threats. Internal threats are for example the employees of the customer. External threats are for example competing companies, hacktivists and individuals who cause destruction for self-amusement. When all the possible threat agents and threat communities have been surveyed, a possibility to carry out an attack should be analyzed for each one of them. The analysis should determine the availability of tools, skills to use these tools, possibilities to get malicious software and code related to the target and possible communication methods which can be used to move information to and from the target. Also, the possible motivations for an attack should be analyzed. These motivations can vary greatly between threat agents from a financial benefit of a competitive company to an attempt to increase reputation by a single individual. (Threat Modeling - The Penetration Testing Execution Standard.)

The last step of the threat modeling is to find examples relevant to the threats. For example, a tester can gather news and analysis of the actual attacks that have occurred to a target which fits the created threat model. This can be used to justify the reality of the threats and to compare the results of the threat modeling to the actual occurred threats. (Threat Modeling - The Penetration Testing Execution Standard.)

Vulnerability Analysis

Vulnerability analysis phase aims to survey all the vulnerabilities of the target. The structure of the process depends on the target to be tested, however, it can roughly be divided into the following steps: finding vulnerabilities by testing the target devices and applications, validating the test and found vulnerabilities and finally researching the found vulnerabilities. (Vulnerability Analysis - The Penetration Testing Execution Standard.)

The testing for vulnerabilities can either be active or passive. Active testing is carried out directly by interacting with the target of the testing. For example, it can be automatic when using different tools e.g. vulnerability scanners or it can also be manual when connecting to the target and finding possible known vulnerabilities. Passive testing is carried out by getting information about the target without directly interacting with it. This can be, for example, sniffing unprotected network traffic which can result in detection of vulnerabilities in the source or the destination of the traffic. (Vulnerability Analysis - The Penetration Testing Execution Standard.)

After the testing, the testing process and its results should be validated. By validating the process, it can be made sure that all targets of the scope were tested and that the test was executed in the specified level of detail. For example, in the validation the found vulnerabilities can be filtered from false positives and similar vulnerabilities can be combined by e.g. CVE identifier. The validation can also include a procedure, in which all vulnerabilities found by the automated scanners are verified by manual testing. Finally, the found vulnerabilities should be researched in close detail so that for example possible causes of the vulnerabilities are identified and different ways of exploiting the vulnerabilities in attacks are surveyed. (Vulnerability Analysis - The Penetration Testing Execution Standard.)

Exploitation

The aim of the exploitation phase is to capture the device or the software by using an attack which has been crafted by using the information gathered in the previous phases. If the information gathering phase and the vulnerability assessment phase were carried out comprehensively, it should be possible to execute the attack as a precise strike. This attack is usually crafted by finding an attack path which contains the least resistance and where the chance of an exposure is the smallest. (Exploitation - The Penetration Testing Execution Standard.)

The objective of the attack is to gain access to information which has been classified as an asset in the previous phases. The attack is carried out by using the found vulnerabilities and the attack can also use a human-element, for example an employee who opens email containing malicious code

sent by the tester. During the planning of the attack, all information related to the possible protection systems and methods should be used because these can hinder or even prevent the attack and cause its exposure to the target. All techniques that can for example help to evade the protection systems, disable those or hide from those by masquerading the attack as something not detected by the protection systems should be used. The PTES explains in close detail some of the possible protection systems which can be encountered during penetration testing and some techniques which can be used to counter these systems. When planning the attack, all vulnerabilities should be researched for possible readily available exploits and attacking methods as these can often be used in the attack directly or after doing small modifications. The tester can also try to find new "zero-day" vulnerabilities on the target, though this is usually the last resort as finding these is a difficult and time consuming process. (Exploitation - The Penetration Testing Execution Standard.)

Post exploitation

In the post exploitation phase the value of the captured device or software is determined by evaluating the possible sensitive information and assets that it might contain, and also by the possibility to use it in a further attack to the network. By using the information gained in post exploitation phase, existing high value targets can be detailed and possibly new high value targets can be specified. Information gathered in this phase can also be used in the reporting phase. (Post Exploitation - The Penetration Testing Execution Standard.)

The PTES offers a comprehensive list of example rules which can be used as a guideline for this phase. By following these rules a tester can make sure that there is no unnecessary harm caused to the customer or the tester. These rules specify for example how the acquired sensitive data should be stored during and after the test and what legal precautions should be taken into account when handling different kinds of sensitive data. It is very important that the rules specified in the pre-engagement phase are also taken into account. When the rules of the post exploitation phase have been determined, the post exploitation phase can be executed. The process can be divided into multiple steps: analyzing the infrastructure, looting, investigating high value targets, exfiltration of acquired data, creating persistence, further attacks towards the network and cleaning up the evidence of the attack. (Post Exploitation - The Penetration Testing Execution Standard.)

An analysis of the infrastructure can provide helpful information in developing further attacks to the network. For example, by analysing the network configurations of a device, tester finds out e.g. settings of different interfaces, routing configurations, used DNS-servers, cached DNS-

entries, used proxy-servers and ARP table values. From the network services of the device, at least the listening services, current VPN-connections and Directory services should be analyzed. (Post Exploitation - The Penetration Testing Execution Standard.)

In the looting part of the process, the objective is to search and acquire all information that was determined as sensitive or high-value in the pre-engagement phase or in the threat modeling phase from the captured device. Also, any information that can help in further attacks should be searched and looted. The PTES mentions that tester should investigate comprehensibly all services that start on system boot, security services, file and printer sharing services, database services, directory services, domain name services, deployment services, services related to handling of certificates, source code control services, dynamic host configuration services, software and services related to virtualization, messaging software and services, back-up systems, network systems and monitoring systems. In addition to software and services, tester should also investigate all possible targets of the device that could have user-related information. These can be, for example, files saved by user, log files, history files, configuration files and configuration settings related to system security. Sensitive information can also be acquired by monitoring the users' actions on the captured device. This can be done for example with using a software that logs all key presses, by saving the network traffic of the device or by taking screenshots of the screens of logged users. (Post Exploitation - The Penetration Testing Execution Standard.)

After all the sensitive information has been sought and acquired, it needs to be exfiltrated outside from the target network. The target location of the exfiltration should be a place which has been determined in the pre-engagement phase and it can be e.g. a server provided by the customer or the device that the tester uses for the testing. Exfiltration of the information should be done so that it does not expose the tester, because of this all possible ways to exfiltrate the data from the captured device should be analyzed. These ways can include for example compressing the information and transferring it to an external FTP-server by using direct connection or encrypting the information and sending it to the exfiltration location by hiding it inside multiple DNS-requests which are routed through multiple devices. After the exfiltration ways have been analyzed, the data should be exfiltrated. Purpose of the exfiltration is to examine the response of the target's security systems and security personnel on detecting and preventing the transfer of the sensitive data. (Post Exploitation - The Penetration Testing Execution Standard.)

After all the information has been exfiltrated, the tester should ensure that the captured device can be accessed later if needed. This can be done for example with installing a backdoor which can

handle restarting of the device or by reconfiguring the network service settings so the services allow tester to access the device. (Post Exploitation - The Penetration Testing Execution Standard.)

After the persistence is ensured, the device should be used to execute an attack further to the target network. This can be done by using the captured device or by routing an attack through the captured device. The captured device can be used in the attack for example by scanning the network, by doing brute force attacks towards the other devices of the network and by abusing the services and sensitive data located on the captured device. The attack can also be done by pivoting, which means routing the tester's network connection through the captured device. This is useful when the tester does not want or cannot install penetration testing software to the captured device. Pivoting can be done with multiple different methods including e.g. creating a VPN tunnel, creating port forwards and tunneling the connection through SSH. (Post Exploitation - The Penetration Testing Execution Standard.)

When the captured device is not needed anymore, it should be cleaned from all the modifications made in the penetration test. For example, all the configuration values should be returned to the original state, all the malicious software should be uninstalled and all created user accounts should be destroyed. (Post Exploitation - The Penetration Testing Execution Standard.)

Reporting

The last phase is reporting of the process and results. The PTES offers an example of the reporting structure, in which two different summaries of the penetration test are specified. These summaries should contain information about the objectives of the test, the methods of testing and the results of the test.

The first of the summaries reports is called an executive summary. It is targeted for the persons who are responsible of the security strategy and the persons who are affected directly or indirectly by the threats uncovered in the penetration test. An executive summary is recommended to contain all following sections: background information, overview, risk ranking, general findings, recommendation summary and strategic roadmap. (Reporting - The Penetration Testing Execution Standard)

The background information section should inform the readers about the reasons why the penetration test was ordered, what the scope of the penetration test is and why the targets inside

the scope were picked, a short summary of the penetration test process and other possible information explaining the necessity of the penetration test in the situation. (Reporting - The Penetration Testing Execution Standard)

The overview-section should explain how effective and successful the penetration test was and how it achieved its objectives. The section can also be used to inform about vulnerabilities and other problems recurring during the entire test. (Reporting - The Penetration Testing Execution Standard)

The risk ranking-section defines the risk level of the customer by analyzing the found vulnerabilities and problems. This definition should give the reader a picture of how likely the threat is, what the cause of the threat is and what the consequences are in case the threat occurs. The section can also be used to explain the most critical vulnerabilities found and what the consequences of the risks are that can occur because of these vulnerabilities. Figure 3 shows an example of a risk rating scale that can be used for defining the risk level. (Reporting - The Penetration Testing Execution Standard)

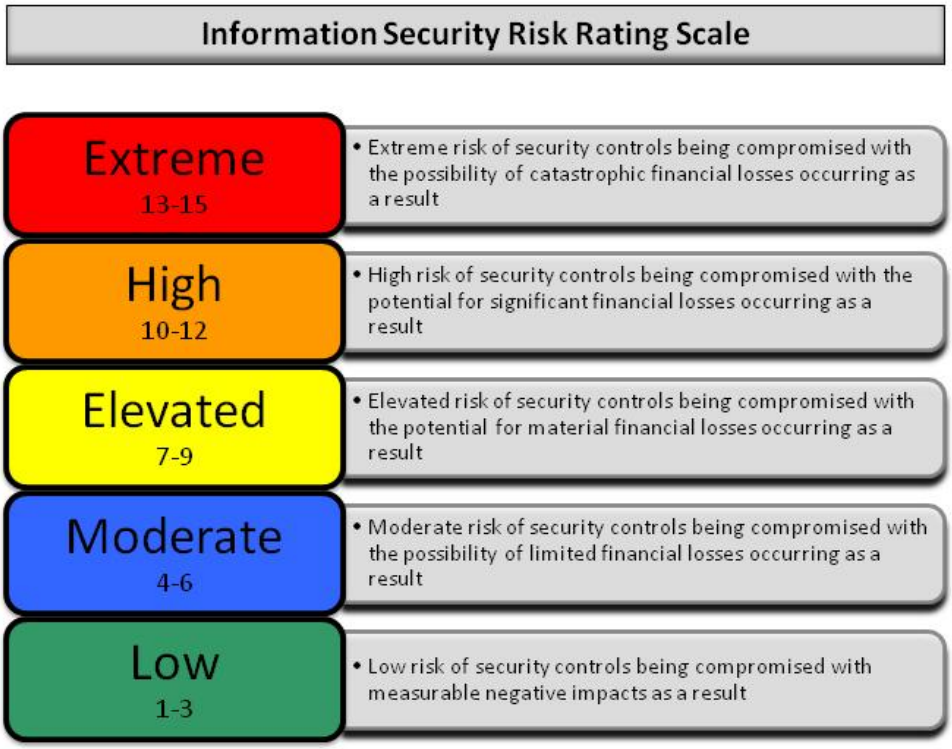


Figure 3 Information Security Risk Rating Scale by Penetration Testing Standard. (Reporting - The Penetration Testing Execution Standard)

The general findings section offers and summarizes the found problems and vulnerabilities. This summary should be offered in an easily readable simple and statistical format. The summary should also include all visualizations defined in the pre-engagement phase. For example, the summary can include graphs visualizing the targets of the test, the testing process, the categories of the found vulnerabilities, the categories of the risks and categories of the threat agents. This section can also be used to represent summaries and visualizations of the performance of the protection systems and other possible information specified in the pre-engagement phase. Figure 4 shows an example of the PTES on how to visualize the results of penetration tests. (Reporting - The Penetration Testing Execution Standard)

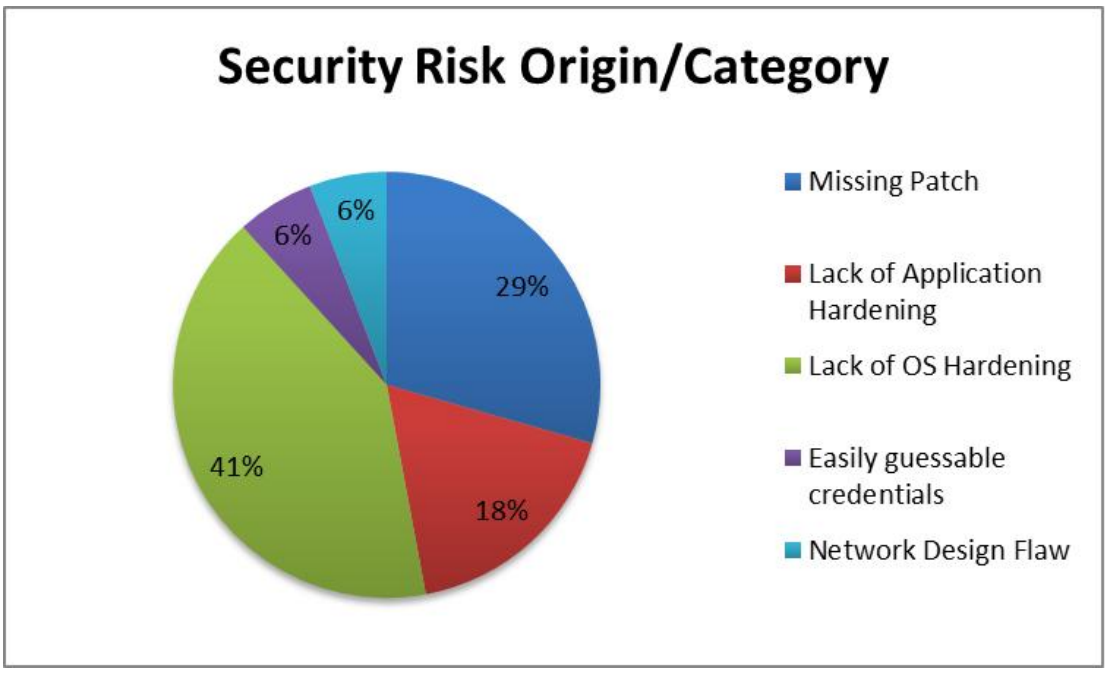


Figure 4 PTES example of a graphical representation of the security risk origins. (Reporting - The Penetration Testing Execution Standard)

The recommendation summary section should inform the reader about the tasks needed to resolve these threats and how much effort these tasks need. The section can also be used to explain how the roadmap in the last section has been created and on what basis the prioritization of the tasks was done. The roadmap of the last section can define timespans with different tasks that should be done to increase the security level. Figure 5 shows an example of a roadmap by PTES. (Reporting - The Penetration Testing Execution Standard)

Completed at the time of this assessment

Tasks

Identify internal security point of contact

- Identify current resources to dedicate the task of resolving security concerns within the environment. The remediation process should be owned and supported by senior staff in order to effectively manage its completion.
- Secure appropriate funding for initial program review and 3rd party assessment

Identify Current Security State of security

- This task will be performed at an executive level. CLIENT will identify the proper ownership and executive support channel to champion this effort. In addition, CLIENT will need to take inventory of the "Security Management Chain of Command", Policy, Procedure, and Compliance tracking sophistication.

One (1) to Three (3) Months

Tasks

Create Remediation Strategy

- Leverage results found within the Penetration Test to create a full remediation strategy
- This assessment report will provide the basis for this action. It must now be formalized and approved by the CLIENT Security Team.

Create Information Security Council/Task Force

- To gain better traction in the remediation and security onboarding process, CLIENT should create a specific ISEC council to aid in remediation and adequately involve each individual team.
- The council should consist of Management of each individual business unit
- ....

Begin Security Project planning

- Assign Executive owners of security for CLIENT
- ...

Prioritize Remediation Events

- Leverage results found within Penetration Test to gain understanding of the tasks needed to be performed in order to resolve the risks identified.
- Assign priority listing to remediation tasks that will provide the highest level of impact and largest reduction of identified risk.
- Start process with server patching to gain quick increases in environment security.

Patch Services

- **Specific things to be fixed/how…**
- **...**

Harden Servers

- ...
- ...

Three (3) to Twelve (12) Months

Tasks

Security Self Assessment

Adequate security of information and the systems that process it is a fundamental management responsibility. CLIENT officials must understand the current status of their information security program and controls in order to make informed judgments and investments that appropriately mitigate risks to an acceptable level. Self-assessments provide a method for CLIENT officials to determine the current status of their information security programs and, where necessary, establish a target for improvement. A good guide for this is **NIST SP 800-53a** :found at http://csrc.nist.gov/publications/PubsDrafts.html. Another approach would be to run the **Microsoft Security Assessment Tool** : found at http://www.microsoft.com/technet/security/tools/msat/default.mspx

Twelve (12) Months+

Tasks

Perform 3rd Party Assessment of Information Security and Compliance with 27001/2 (or any other compliance control set chosen).

- Perform a Corporate wide assessment of CLIENT's ability to defend against targeted & generic attacks
- Identify the root cause of compliance gaps
- Identify strategy for using the output of the assessment to facilitate a security baseline

Begin remediation planning/budgeting

Figure 5 Example of strategic roadmap by PTES. (Reporting - The Penetration Testing Execution Standard)

The second summary report of the reporting phase is called a technical report. This report should contain detailed technical information about the penetration test process and its results. The report should also contain detailed technical information on the consequences occuring if the uncovered risks come true, and on technical tasks how to prevent these risks. The technical report should contain the following parts: introduction, information gathering, vulnerability assessment, exploitation or vulnerability confirmation, post exploitation, risk/exposure and conclusion. (Reporting - The Penetration Testing Execution Standard)

The introduction-section should act as a summary of the resources related to the penetration testing and about what the penetration testing includes. At the least it should include contact information of the persons who took part in the planning and executing the penetration test, information about the scope and assets of the penetration test, objectives of the penetration test, starting point of the test, summary of the strengths of the test and explanation of the threat and grade structure. (Reporting - The Penetration Testing Execution Standard)

The information gathering section should present the results of the intelligence gathering phase. These results can be divided into four different categories: passive intelligence, active intelligence, corporate intelligence and personnel intelligence. Passive and active intelligence categories present the result of active and passive information gathering that was done. Corporate intelligence presents all gathered information related to the customer's business process and business assets. Personnel intelligence presents all gathered information relating to the customers' employees. In each of these categories, the report should also describe the methods and techniques used to acquire the results. (Reporting - The Penetration Testing Execution Standard)

The vulnerability assessment-section should present exact descriptions of the possible vulnerabilities that were found during the testing. The section should present the threat levels of these vulnerabilities, categories of the vulnerabilities, information on how the vulnerabilities were found and how visible the vulnerabilities are. In case of the network vulnerabilities, the OSI layer of the vulnerability should be also presented. The section should also provide a summary of the results of the vulnerability assessment phase. (Reporting - The Penetration Testing Execution Standard)

The exploitation-section, also known as vulnerability confirmation-section should describe in very close detail, how the vulnerabilities of the previous section were validated and which preconditions the vulnerabilities required. Each executed attack should be described in step-by-step detail and the descriptions should also include location of the attack in the timeline of the

testing, targets of the attack, actions taken during the attack, results of these actions and how successful these actions were. (Reporting - The Penetration Testing Execution Standard)

The post exploitation-section should describe the results of the post exploitation phase in a very close detail. The section must be written so that it combines the uncovered risks to the consequences that will occur to the customer if the risks come true. The section should represent the results of the post exploitation phase which can include e.g. privilege escalations, acquiring critical assets, acquiring sensitive information, acquiring any other valuable information, access to restricted data and devices, possibility to create persistence to the captured device, exfiltration methods and performance analysis of the possible protection systems. The evidence can be for example screenshots and captured network traffic. All the evidence should be sanitized before attaching to the report, so that the readers cannot see the actual restricted information. (Reporting - The Penetration Testing Execution Standard)

The risk/exposure-section should give the customer information which the customer can use to evaluate the results of previous sections in financial and business related views. This section should include for example an estimation of which skills and access rights are required for the threats to come true, how often the threats can occur and what the direct and indirect consequences of a threat coming true to the customer are. (Reporting - The Penetration Testing Execution Standard)

The conclusion section is the last part of the report. It should contain discussion about the test and results of the test as a big picture. The section can also be used to guide the customer on how to start improving the security situation and to define which cyber security related actions and testing the customer should carry out in the near future. (Reporting - The Penetration Testing Execution Standard)

2.2 OWASP Top Ten project

The Open Web Application Security Project (from now on referred to as ”the OWASP”) is an open community, whose objective is to provide reliable information related to the security of web applications. The OWASP provides for example tools related to the security of the software, books, software components, software libraries and top-level research. (OWASP Top 10 – 2013.)

OWASP Top 10-project is one of the main contents provided by the OWASP, its purpose is to improve the software security related knowledge of the persons working in the software industry.

The OWASP Top 10-project releases publications roughly every third year and these publications contain descriptions of the 10 most critical web application vulnerability categories for the year of the publication. During the production of the thesis, newest publication was the OWASP Top 10 2013. This publication is based on 8 different data sets which include information of over 500000 vulnerabilities in thousands of different applications and in hundreds of different organizations. The created laboratory environment was designed to include all of the OWASP Top 10 2013 OWASP vulnerability categories in one way or another. Following sections provide short descriptions of the each vulnerability category of the OWASP Top 10 2013 publication. (OWASP Top 10 – 2013.)

A1 – Injection

By exploiting the injection vulnerabilities, malicious commands and queries can be executed in the web application. These commands and queries are not designed to be executed in the web application by the developers of the application and can be used to change the operation of the application. The most common causes of the injection vulnerabilities are unvalidated user inputs and outputs. These vulnerabilities can reside, for example, in database queries and directory queries created by the web application or in the fields of the HTTP request header which the web application uses. The injection vulnerabilities can be used for example to bypass authentication of the web application and to retrieve database contents. In the year 2013, roughly 80 % of the attacks toward web applications were carried out by using SQL injections. (OWASP Top 10 – 2013.; 2014 Data Breach Investigations Report (DBIR).)

For example, in the case of an injection vulnerability, this section illustrates an SQL injection attack towards a database by using a vulnerability of a web application. A “login.php” page of the web application retrieves all the rows from the SQL-database that match the credentials sent in a POST request of a login form. If there is at least one matching row, the user that sent the POST request will be logged in to the web application. Figure 6 shows an example of a vulnerable source code that creates the SQL-query used in the credential check. Figure 7 shows the created SQL-query when a user called “jack” tries to log in using his password “hunter2”.

```
$sql = "SELECT * FROM users WHERE username='" . $_POST['username'] . "' AND password='" . $_POST['password'] . "'";
$user_rows = mysql_num_rows(mysql_query($sql));
```

Figure 6 PHP-script that creates a SQL-query vulnerable to an injection attack.

```
SELECT * FROM users WHERE username='jack' AND password='hunter2';
```

Figure 7 SQL-query which is generated by the PHP-script in figure 6.

In the SQL injection attack, this query can be modified by injecting parts of an SQL query to the unvalidated login form fields. Figure 8 shows an example of the injected parts that result in a SQL-query that bypasses the password check by inserting an "OR"-clause that results always as true instead of the password. The web application sends this query to the database and gets the admin-user row which results in the attacker being logged in as the admin.

```
Username: admin
Password: ' OR ''='

SELECT * FROM users WHERE username='admin' AND password='' OR ''='';
```

Figure 8 SQL-injection by using unvalidated fields.

A2 – Broken authentication and session management

The A2 category vulnerabilities can appear in web applications that have misconfigured or misimplemented authentication and session management related functionalities. These functionalities can be for example login and logout methods, password change and restore functionalities, session timeouts and functionalities related to the session token. (OWASP Top 10 – 2013.)

An example of an A2 category vulnerability could be an attack which uses misimplemented user account handling of a web application. The registration page of the web application can be used to create an account to the web application. The page adds the clear text username and password sent by the new user to a "users" table of the database that the application uses. Because the implementation of the registration-page does not use any kind of hashing or protecting of the credentials, the attacker that gets access to the database can retrieve credentials of all users of the web application in clear text and use these for example in an attack to another web application. (OWASP Top 10 – 2013.)

A3 – Cross-site scripting

Cross-site scripting (from now on referred to as "XSS") can be executed when the web application shows unvalidated user inputs in a web-page. The XSS can be categorized in three different categories: stored XSS, reflected XSS and DOM based XSS. In the stored XSS, malicious code

sent by the attacker is saved on the vulnerable page of the web site. When the targeted user(s) open the vulnerable page, this malicious code is executed in the browser of the user(s). Stored XSS can be found in e.g. messages of Internet forums, comments of blog pages or even in a product page that was created by the attacker to an online auction website. In the reflected XSS, no malicious code is saved to the vulnerable web page, instead the malicious code can be included in e.g. query parameters of a HTTP request. For example, when the user opens a hyperlink that contains malicious XSS payload, the target web page with the XSS vulnerability executes the payload in the user's browser. In the DOM based XSS, the malicious code is added to the DOM-structure of the web page in a way that the HTTP response of the web page does not contain any part of this malicious code. This can appear for example in a situation where the XSS payload is being supplied in a query parameter of the HTTP request and a JavaScript code inside the page retrieved in the HTTP response adds this XSS payload to the DOM-structure of the page. (OWASP Top 10 – 2013.)

As an example of a XSS vulnerability, a DOM based XSS vulnerability is presented. The admin of the web application has created an error page which reads the error message from a query parameter of the HTTP request (see figure 9). In normal situation this page shows an ”Error!”-heading and a message that describes the error to the user.

```
<HTML>
    <HEAD>
        <TITLE>ERROR!</TITLE>
    </HEAD>
    <BODY>
        <H1>Error!</H1><BR />
        <SCRIPT>
            var position=document.URL.indexOf("errortext=")+10;
            document.write(decodeURI(document.URL.substring(position,document.URL.length)));
        </SCRIPT>
    </BODY>
</HTML>
```

Figure 9 Error page that contains Dom based XSS vulnerability.

The attacker sends a hyperlink that contains a XSS payload to the targeted user. When the user opens this hyperlink, the browser of the user retrieves a page from the web site that does not contain any XSS payload or malicious code. After the page is loaded, the vulnerability of the page results in adding the payload to the DOM-structure and executing an XSS-attack. The figure 10 shows a result of an attack executed to the web page created by the source code in the figure 9. Figure 11 shows the DOM-structure of the page after the attack has been executed. The

malicious “script”-element was added locally to the page, hence the HTTP-response body did not contain any malicious code.

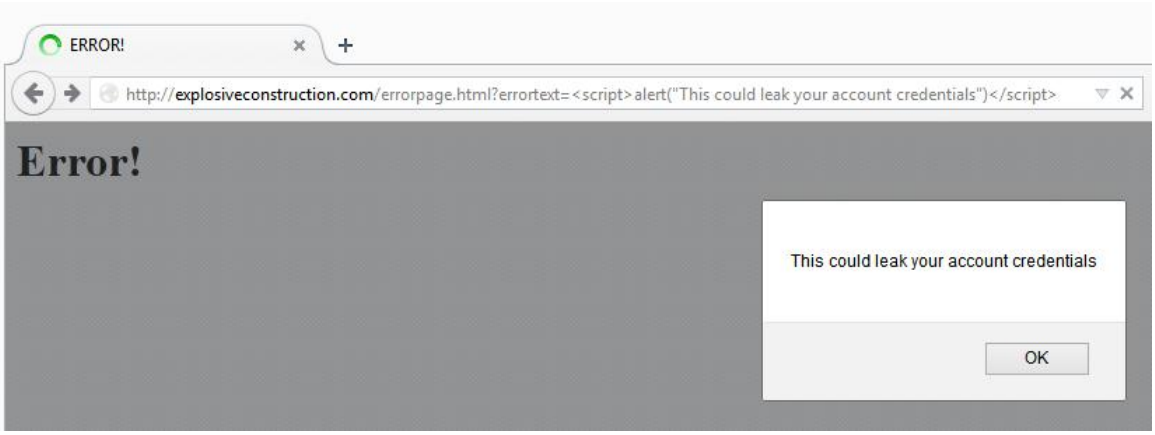


Figure 10 DOM based XSS vulnerability exploited.

```
<html>
  <head>
    <title>ERROR!</title>
  </head>
  <body>
    <h1>Error!</h1>
    <br>
    <script>
      1
      2 var position=document.URL.indexOf("errortext=")+10;
      3 document.write(decodeURI(document.URL.substring(position,document.URL.length)));
      4
    </script>
    <script>
      1 alert("This could leak your account credentials")
    </script>
  </body>
</html>
```

Figure 11 Resulting DOM-structure with script-element created by DOM based XSS exploit.

A4 – Insecure direct object references

By using A4 category vulnerabilities, restricted objects can be accessed and used. These objects can be e.g. information about users of a web application or any other sensitive information. These vulnerabilities can be found in misimplemented functions that provide information by using direct references to objects.

For an example, a web page in which the administrator has implemented retrieving of customer data with an REST-call to the address “http://vulnerable.com/customer/[customerid]”. The call results in a response containing the JSON-formatted data of the user. Because the admin hasn’t created any kind of access restriction to this functionality, the attacker can simply iterate the

numeric ”[customerid]”-field and retrieve data of all other users of the service. (OWASP Top 10 – 2013.)

A5 – Security misconfiguration

The A5 category vulnerability allows the attacker to access sensitive data or restricted system due to misconfiguration of the security settings in the target system. A vulnerability of this category can appear in every level of the software used in the server, for example it can reside in frameworks and libraries used in web application, server software used to serve the web application, other services of the server and in the operating system of the server. These vulnerabilities can include e.g. old software versions, user credentials left to the default value, misconfigured security related configuration values, unneeded services, error messages that show sensitive information and security-related variables of the frameworks used. (OWASP Top 10 – 2013.)

A6 – Sensitive data exposure

The A6 category vulnerability stands for exposure of data, which results in harm to the service provider or to the users of the service. Common causes of this vulnerability are missing or improper implementation of data encryption. (OWASP Top 10 – 2013.)

For an example, the administrator has configured a backup-system after creating an online store. Later the services of the online store were modified so that all existing user credentials inside the database were encrypted and new user credentials are automatically encrypted. The administrator forgot to encrypt the already created backup-files. If the attacker can access the backup-server, all the credentials created before adding the encryption can be retrieved in clear text from the old backup-files. (OWASP Top 10 – 2013.)

A7 – Missing function level access control

Missing function level access control can result in using restricted functionalities without needed access rights. For example vulnerability can result in anonymous user gaining access to restricted functionality or normal user gaining access to functionality intended only to administrators. These vulnerabilities can appear in different function calls which can reside for example in URL-addresses or in the API-calls of a web application.

For an example, a web application has a page “http://vulnerable.com/admin/reset_settings”. When the page is opened, the functionality resets the settings of the web application to the factory defaults. The developer of the web application has added a hyperlink pointing to this page only to the admin pages of the applications and assumes that it will not be used from elsewhere thus any kind of access restriction is not implemented. If the attacker knows the URL of this functionality, the attacker can connect to it directly and reset the settings of the web application. (OWASP Top 10 – 2013.)

A8 – Cross-site request forgery

The A8 category vulnerability enables sending HTTP requests to the vulnerable web page from malicious websites. By using this vulnerability, attacker can for example create a malicious web page, which when opened sends a GET- or a POST-request to the vulnerable page from the targeted user’s browser. After the attacker has created the page, the attacker can send hyperlink to this page to the targeted user. When the user opens the page, the browser of the targeted user sends the request determined by attacker to the vulnerable web page and the web page executes the actions requested in the request by using the account of the targeted user. The attacker does not have to know anything specific about the user, only the knowledge about the vulnerability and how to use it is needed and the same attack can be applied to multiple different users. (OWASP Top 10 – 2013.)

As an example, the web application of an online bank uses a POST-request to send money between accounts. Because the developer has not added a token that prevents the vulnerability to the POST-form, the attacker can execute a CSRF-attack. In this attack, the attacker creates a malicious web page on a third-party server. When opened, the malicious page sends a crafted POST-request to the vulnerable web page. The figure 12 shows an example of the malicious page.

```
<!DOCTYPE html>
<html>
<head>
<script>
    window.onload = function(){
        document.forms[0].submit()
    }
</script>
</head>
<body>
    <form action="http://vulnerable.com/sendmoney" method="POST">
        <input type="hidden" name="acct" value="ATTACKER"/>
        <input type="hidden" name="amount" value="10000"/>
        <input type="submit" value=""/>
    </form>
</body>
</html>
```

Figure 12 Automatic POST-request page used in CSRF-attack.

After the page is created, the attacker can execute a spoofing type attack in which the attacker sends the hyperlink of the malicious page to thousands of users. When any of these users open the page, the page tries to send money to the attacker by using the CSRF vulnerability. If the user was logged in to the online bank at the moment when the page was opened, money is transferred to the attacker. Figure 13 shows an example of the requests created by the malicious web page of Figure 12. The first request of the figure is the result of the user opening the malicious page and the second and third requests are launched automatically by the malicious page towards the vulnerable online bank.

Name Path	Method	Status Text	Type	Initiator
pagecreatedbyattacker.html	GET	200 OK	text/html	Other
sendmoney vulnerable.com	POST	302 Found	text/html	pagecreatedbyattacker.html:16 Script
ww12.vulnerable.com	GET	200 OK	text/html	http://vulnerable.com/sendmoney Redirect

Figure 13 HTTP requests created in CSRF-attack.

A9 – Using components with known vulnerabilities

The A9 category vulnerability is a result of using a component which has known vulnerabilities. These components can be e.g. operating system of the server, server software, web application frameworks and plugins of the web applications. These vulnerabilities can also appear as client-side in a web browser and its plugins. The A9 category vulnerability can be fixed by using the

latest version of the software or by installing a fix released by the manufacturer. Known vulnerabilities of the A9 category vulnerability can belong to any of the other OWASP Top 10 categories. (OWASP Top 10 – 2013.)

A10 – Unvalidated redirects and forwards

By using an A10 category vulnerability, the attacker can force the targeted user to be redirected or forwarded to a malicious web page. This vulnerability can be caused by redirect- and forward-functionalities of a web site which accept unvalidated user inputs. (OWASP Top 10 – 2013.)

As an example, a vulnerable online bank that has been configured to redirect mobile users from the desktop site to the mobile site. This redirecting functionality has been created by using a ”redirect.php” page. When the user opens the desktop site with a mobile device, the user is forwarded to the “redirect.php” page which then redirects the user to the URL supplied as a GET-parameter of the HTTP request. The attacker can abuse this functionality by crafting a URL that points to the “redirect.php”page and add a malicious URL-address to the GET-parameter that defines the target URL of the redirecting. After crafting the URL, attacker can send it as a spoofing-type email to thousands of users. If the targeted users don’t read the full URL but only the beginning of it, the URL appears as a non-malicious URL pointing to the website of the online bank. After connecting to the URL, the targeted users are redirected to the malicious web page created by the attacker. (OWASP Top 10 – 2013.)

2.3 KALI LINUX – OPERATING SYSTEM FOR PENETRATION TESTING.

Kali Linux is an operating system based on the Debian Linux-distribution. It was designed for to be used in penetration testing. Kali Linux has hundreds of different tools, which can be used for different purposes of penetration testing. These tools provide functionality for e.g. intelligence gathering from networks and files, analyzing vulnerabilities, executing wireless attacks, analyzing and attacking web applications, exploitation, forensic purposes, stress testing, sniffing, spoofing, password attacks, reverse engineering software, documenting and reporting penetration test, hacking embedded devices etc. The laboratory environment created in this thesis and its vulnerabilities were designed and produced so that testing and exploiting those requires only an access to a Kali Virtual machine and its tools. The Kali Linux was chosen because JYVSECTEC used it already for educationing and researching purposes. (What is Kali Linux?.; Kali Linux Tools Listing.)

2.4 EXISTING ENVIRONMENTS

Before starting the development of the laboratory environment, the existing publicly available laboratory environments designed for the testing and teaching of penetration testing techniques and tools usage were studied. JYVSECTEC wanted these environments to be studied briefly and if possible, some of these environments would be used when creating the laboratory for the assignment. In order to use an existing environment directly, it should have comprehensive documentation and a license that allows commercial usage and it should also be easily modifiable to be used as an exam. After the brief studying it was decided by the author of the thesis that none of the existing environments would not be directly used in the laboratory of the assignment, but instead some of the vulnerabilities and services in these environments would be used as a source of ideas on the development phase of the laboratory environment. The readily available environments were not used directly, because most of the ones that were studied contained unclear licensing. It was also estimated that creating the environment from scratch would result in achieving the objectives most effectively with the least amount of overhead time. Following sections provide short descriptions of three different publicly available environments which were used as sources for ideas in the development of the laboratory environment.

PentesterLab

PentesterLab is a web page, which offers multiple different virtual-machine based courses for learning different skills related to penetration testing. These courses consist a virtual machine-image that contains exercises for the course and a theory material that can be used to learn the techniques related to the course topic. Topics of these courses are related to different vulnerabilities and vulnerability categories.

As an example, the “Web for Pentester” course covers the basics of penetration testing web applications for vulnerabilities and the most common web application vulnerability categories. The virtual machine of the course contains a Linux-based operating system, which has a website that includes sample vulnerabilities for the each vulnerability category of the course (see Figure 14).

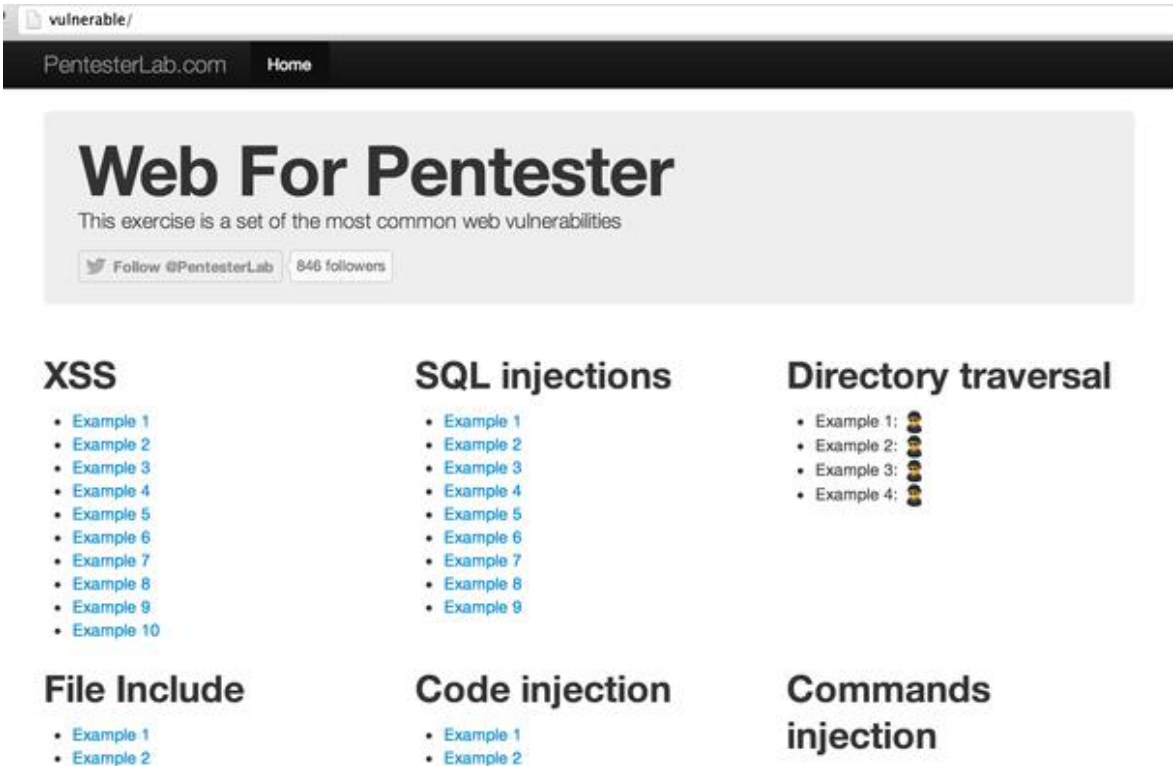


Figure 14 Website included in the virtual machine of the "Web for Pentester"-course. (Web for Pentester)

The theory material of the course is available on PentesterLab.com-website. This material includes the required theory information needed to exploit the vulnerabilities of the course. Figure 15 shows a short part of the theory material provided by the course.

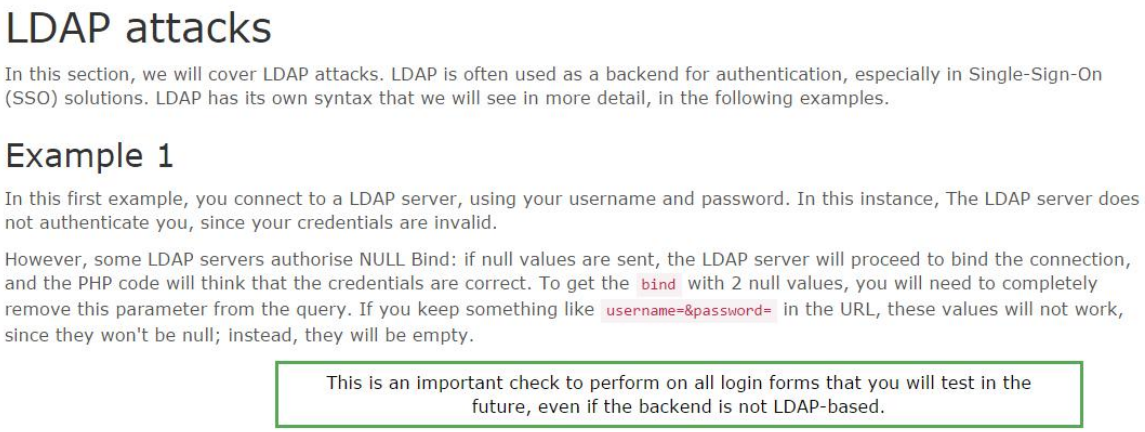


Figure 15 Example of a course theory for "Web for Pentester"-course. (Web for Pentester)

Metasploitable 2

Metasploitable 2 is a virtual machine created by the Metasploit-team. This virtual machine contains multiple different server software and web applications which contain multiple different

vulnerabilities. These can be used for studying and improving skills of penetration testing. Metasploit's weakness is its old age, even the newest vulnerabilities of the environment published in the year 2012 are starting to get obsolete and are not appearing as often in the Internet anymore. (Metasploitable 2 Exploitability Guide.)

Damn Vulnerable Web Application

Damn Vulnerable Web Application or DVWA is a PHP- and MySQL-based web application. Its objective is to help information security professionals, software developers, teachers and students in learning and using techniques related to the security of web applications. Also, the DVWA is published in the year 2012, but this does not matter much because the environment is concentrated on the different techniques and categories of the vulnerabilities and not on the specific versions of software that have known vulnerabilities. The figure 16 illustrates an XSS-vulnerability section of the DVWA. (DVWA – Damn Vulnerable Web Application.)

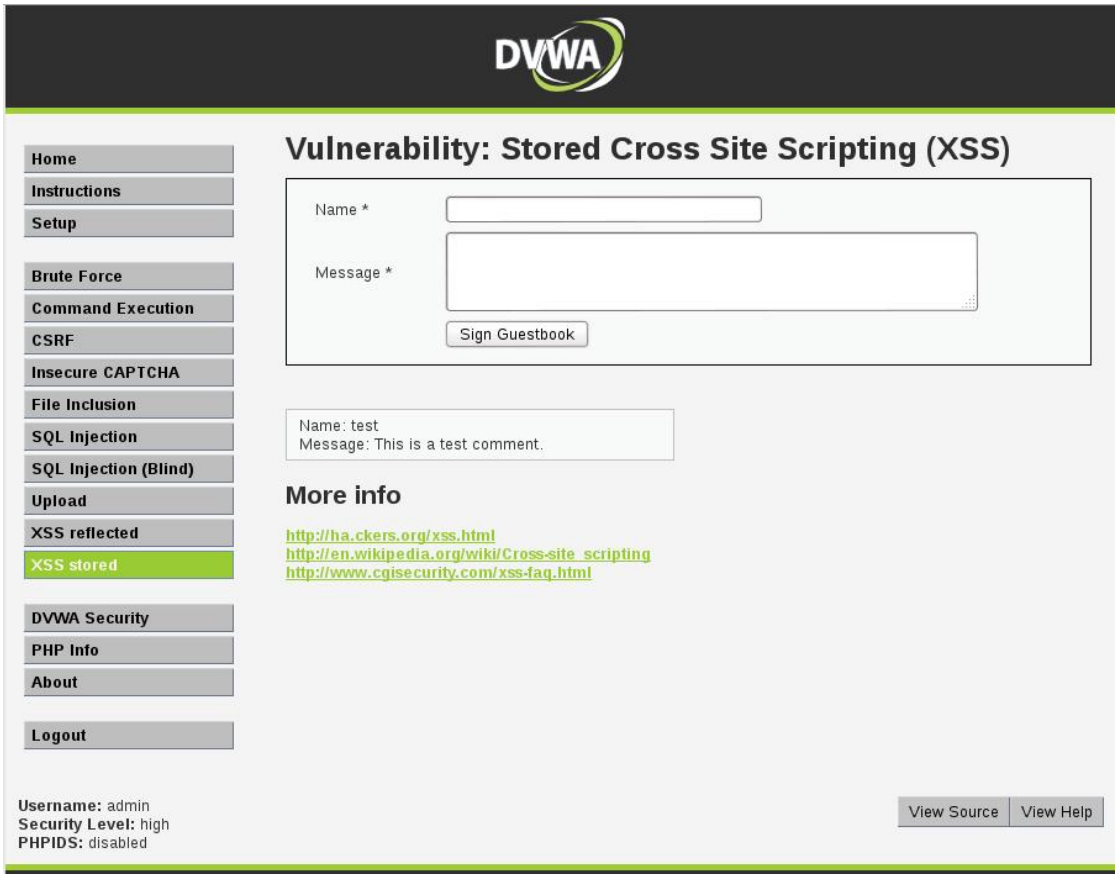


Figure 16 XSS-page of DVWA.

3 TRACKING SOFTWARE

3.1 INTRODUCTION

One of the entities defined in the assignment was to produce a software which can be used to track students' progress inside different laboratory environments. The software was required to be designed so that it can be easily integrated to different learning platforms. It was also required to be designed so that it can be used either in a single-user mode or in a multi-user mode. The single-user mode can be used for example when the software is used to track a single student's progress inside a laboratory environment that is used as an exam. The multi-user mode can be used for example in capture-the-flag type cyber security exercises where the progress of multiple different teams and users must be tracked. The tracking software will be from now on referred to as "the flag-software" or as "the tracking software".

The flag-software was decided to be produced by using the Go programming language. The Go programming language was selected because it supports cross-platform compiling and provides a standard library which can be used for implementing different techniques to the flag-software. One of the reasons for the selection was also that the RGCE-environment has existing software that has been created with the Go-language, thus JYVSECTEC can combine the functionalities of the flag-software to the existing software. For example JYVSECTEC can integrate the flag-software to the botnet of the RGCE and use this to create non-malicious traffic and automated attacks when the student has progressed to a certain point in the laboratory-instance. The figure 17 illustrates the different parts of the flag-software.

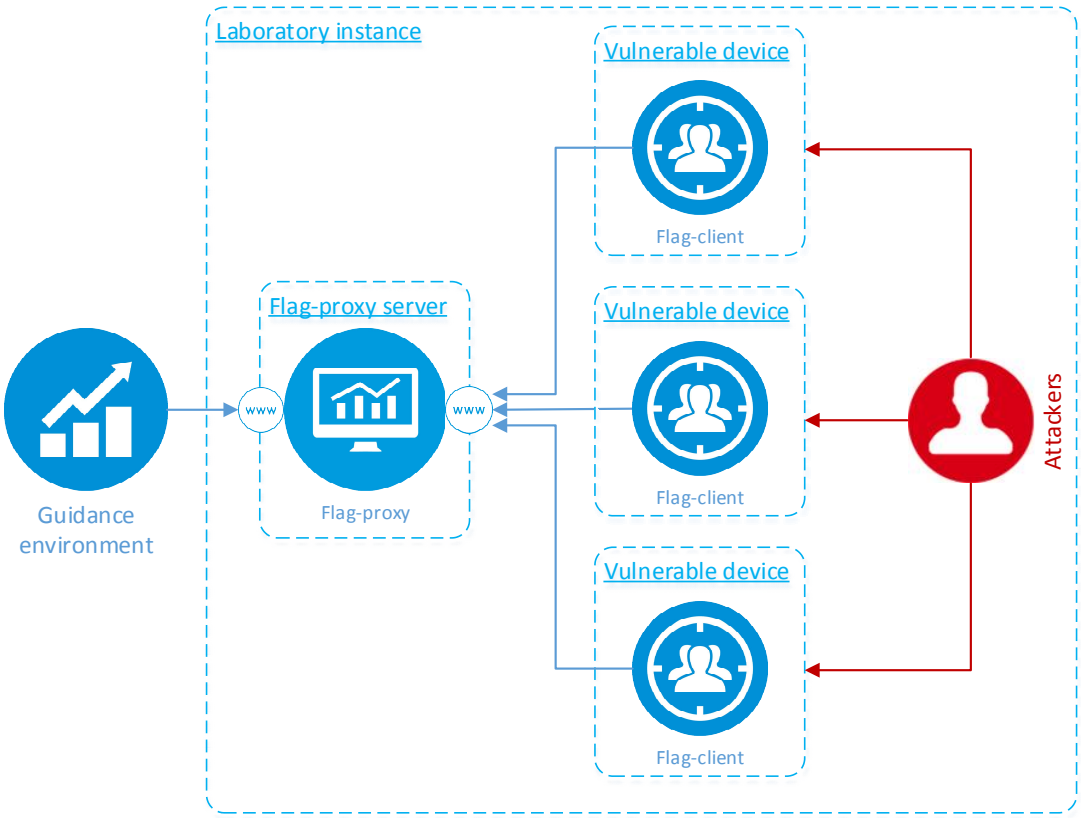


Figure 17 Example of positioning the flag-software to a laboratory.

All created source code was licensed with the MIT-license (http://opensource.org/licenses/MIT), which is a free software license. The license was chosen because it allows closed-source commercial use of the licensed source code.

The more detailed descriptions about the architecture, security measures and usage can be found in the appendix 8.

4 Laboratory environment

4.1 Introduction

The biggest part of the assignment was creation of the laboratory environment suitable for teaching, testing and grading of penetration testing related skills. This is the only one of the three different entities that was included in the original assignment (see appendix 1) and therefore was given the most time and attention. JYVSECTEC wanted an environment which is comprehensively documented, realistic, versatile and easily usable in further development. For the full duration of creation process, JYVSECTEC expressed wishes of different features that could be added to the environment. Some of these features were implemented and some were not.

Selection of the features was left to the creator of the environment (excluding some specific features that JYVSECTEC required to be included).

4.2 Designing the architecture of the laboratory environment

Designing the initial architecture of the environment required that four different points had to be taken into account: usability, the realistic feel, required resources and the extent of the penetration test possibilities. An attempt was made to find the golden mean, where the created environment would represent a realistic network that can be met in the Internet and which would contain realistic vulnerabilities but would not need too much resources and be too difficult to use. The environment was specified to be created inside a vApp-container provided by the vCloud-environment. The vApp can include one or more virtual machines and it can be used for example to start and stop multiple virtual machines with just one click and it can also be to create a virtualized networks that reside only inside the vApp.

The deployment and usage of the environment was required to be as simple as possible and the environment was also required to enable further development. The simplicity was decided to be implemented by using a snapshot-technique provided by the vCloud-environment. A snapshot taken from a virtual machine or vApp can be used to easily deploy new virtual machine or vApp based on the original one. The snapshot contains the settings, power state and the contents of the random access memory and the hard-disks of the original virtual machine(s) from the moment when the snapshot was taken. The snapshot was planned to be taken from the complete vApp where all of the virtual machines are powered off and where these machines are ready to use after powering them on. This snapshot can be used to create multiple different instances of the original vApp containing the laboratory environment. For example the snapshot can be used to create an individual laboratory instance for every student. Therefore the student can use the laboratory instance freely without affecting the other students. The further development was taken into account by designing these virtual machines of the laboratory to be as independent as possible so that they can be used individually outside the laboratory environment directly or after only slight modifications. It was also decided that the services and the operating systems of the laboratory should be as diverse as possible. These design conventions enabled the possibility to create multiple different versions of the laboratory by changing the virtual machines or the vulnerabilities of the laboratory environment.

The realisticity of the environment was designed to be provided by modelling the environment to match a middle-sized company's computer network. A fictional construction company "Explosive Construction Ltd" was invented and designed to be used in branding the services and contents of the created network. All services and devices of the laboratory environment were required to have a purpose that matches a need of the created company. Objective of adding realisticity was to make the laboratory environment exam feel like a real enjoyable penetration test experience instead of just multiple generic exercises that test a specific penetration testing related skill.

The penetration testing possibilities were designed to be implemented so that the student starts penetrating the laboratory environment from services visible to the public side of the RGCE. The machines that provide these public services would reside in the demilitarized zone (from now on referred also as "the DMZ-network") of the network. After the student has captured the virtual machines of the DMZ-network, the student can proceed towards the vulnerable machines of the intranet (from now on referred also as "the LAN-network"). Each of the machines that could be captured was specified to contain at least two different vulnerabilities, one external that allows the access to the device and one internal that allows a privilege escalation to the administrator user or an access to sensitive information.

Tracking the progress of the student was designed to be implemented by using the created tracking software. Each of the capturable machines would have two different flag-clients, one which can be started after using the external vulnerability and one that can be accessed after using the internal vulnerability. The flag-client that is accessible after the external vulnerability would provide half of the points for the machine and would allow student to proceed to the next capturable machine. The flag-client that can be accessed after the internal vulnerability would provide rest of the machine's points to the user. Also a virtual machine for the flag-proxy was decided to be implemented. This virtual machine would be connected to the DMZ-network and to the flag-network and it would allow sharing the tracking data outside the laboratory environment.

The figure 18 illustrates the base of the architectural design of the computer network. The different subnets, the types of the capturable virtual machines and the path of the attacker are specified on the figure and all detailed developing of the architecture is based on this base architecture.

The more detailed descriptions about the architecture, security measures, vulnerabilities and usage can be found in the appendix 9.

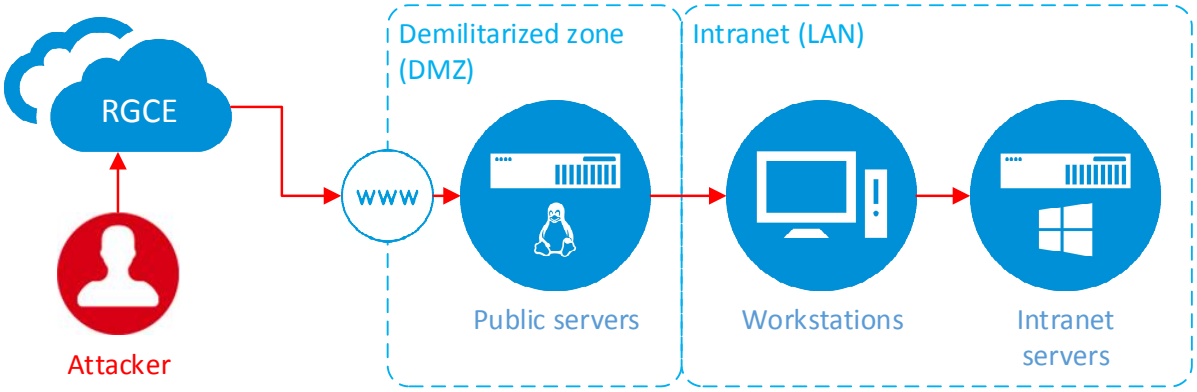


Figure 18 Network subnets, initial attack vector and flow of the scenario.

5 VIRTUAL LEARNING PLATFORM

5.1 INTRODUCTION

During the production, JYVSECTEC added a requirement that a virtual learning platform should be created inside the RGCE. This learning platform was required to provide functionality to be used to guide the students and to grade their skills. The learning platform was specified to be implemented in a way that different courses can be created inside the platform and different students can be added to these courses. Also, a possibility to add theory material, exercises and to run theory exams was required from the learning platform. The learning platform was also required to support the tracking software so that the user's progress inside the created laboratory environment can be automatically followed on the learning platform. The platform should be able to automatically guide and grade the users by using the tracking information.

The learning platform was decided not to be built from scratch due to the large-scale assignment, therefore different open-source licensed learning platforms were studied. From these studied learning platforms, Moodle platform was chosen to be used in the work. Moodle learning platform is a virtual learning platform created with PHP language and licensed with the GPL3 open source license. Moodle can be used in education for e.g. sharing course resources and exercises, as a communication environment and for holding exams and exercises. (About Moodle.). The major reasons for choosing the Moodle were the support for creating plugins and the build-in support of different external authentication services. The plugin support could be used to create a tracking software integration that can be easily deployed. The support for the external authentication services could be used in authenticating the students by using their

existing credentials inside the LDAP service of JAMK. Also, one of the reasons was that Moodle platform was already used in JAMK, therefore the end-users could use the existing knowledge and skills when using the platform. Figure 19 illustrates Moodle's wide support of different external authentication services.

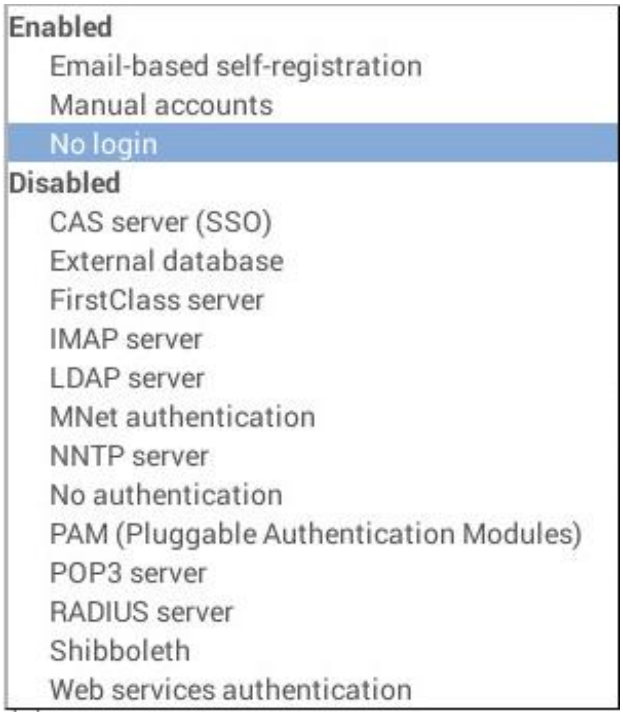


Figure 19 Different external authentication services supported by the Moodle-platform.

During the production of the learning platform, JYVSECTEC detailed the specifications so that the Moodle-platform was required to support tracking, guiding and grading of multiple different laboratory environments simultaneously. The more detailed descriptions about the architecture, security measures and usage can be found in the appendix 10.

6 CONCLUSIONS

The original assignment was to produce a very comprehensively documented laboratory environment which can be used in the cyber security related master's degree programme of JAMK University of Applied Sciences for teaching and grading penetration testing skills. During the production of the environment, the assignment was expanded to include producing of a tracking software and a learning platform. The tracking software was required to provide functionality that can be used to track users' progress inside different laboratory environments and in different cyber security exercises. The learning platform was required to provide functionality for serving theory

material to the students and for carrying out theory exams. The tracking software was required to be integrated into this learning platform so that the students can be automatically guided and graded based on their progress inside the laboratory environment.

The results of the thesis work are a laboratory environment, a tracking software and a learning platform all meeting the assignment's requirements and objectives by providing very versatile possibilities to track, guide and grade users inside different laboratory environments and cyber security exercises. The produced laboratory environment can be used as an exam and it can also be used as e.g. a starting point for creating new laboratory environments and as a target for testing different vulnerability scanners. The tracking software can be used along with the created learning platform and the created laboratory environment and it can also be used independently in different kinds of laboratories and cyber exercises. The created learning platform provides the required functionalities and it can easily be expanded to contain all the theory material, exercises and theory exams that are needed in the degree programme. These results would probably have not been achieved without the summer 2013 internship at JYVSECTEC, where I was familiarized with the RGCE environment and where I learned the basics of the penetration testing.

Due to the large-scale assignment, it could have been possible that the process had taken time excessively or that the results had been incomplete. However, these risks did not come true because the time estimations for each phase of the work and choosing the right amount of features to be implemented on each phase by evaluating these time estimations were performed successfully. Also, documentation of the work and creation of a user friendly environment were carried out successfully because the documentation is clear and comprehensive, and the environment is easy to deploy by using these documentations. For example, deploying the full environment containing the laboratory environment, the tracking software and the learning platform takes only few mouse clicks inside the vCloud environment and customizing it is easy with the created documentation.

The results of the work contain two known problems. The Windows Server of the laboratory environment contains a problem, which results in disconnection from the network after the device has been online for several days. This does not cause great harm as the device can be restarted and within few minutes it is again ready to use. The second problem is the user credentials and other sensitive information of the created laboratory environment. These remain the same in every instance of the laboratory environment. The limitations of the work prevented

implementation of functionality which would change these credentials automatically when the laboratory environment is started.

The biggest slow-downs of the process were the vCloud environment and that I did not have any previous experience for many of the technologies and techniques used in the work. The vCloud environment was very clumsy to use and, for example, changing a network connection of a virtual machine took several minutes and in most of the Unix-based operating systems the clipboard could not be used for transferring text and files from the physical machine to the virtual machine. Therefore, if there was a need to move a small file from the physical computer to the virtual machine, this operation took more than ten minutes at the worst. During the beginning of process, I was completing the final courses of my degree and later I was hired as full time software developer. Therefore, the production of the environment and the thesis was mostly carried out in evenings and weekends.

The environment was required to be easy to use and reliable. These requirements presented the most significant limitations encountered during the thesis. Due to these requirements, the amount of different scripts and configuration possibilities had to be as minimal as possible. For example, a simulated presence of the fictional company's employees could not be implemented to the created laboratory environment because automating it would have required more time than was available. Also, the automatic modifying of credentials and other sensitive data could not be implemented, because automating it reliably would again have required more time than was available.

The results of the work can be used in a very versatile way for different purposes. The resulting environment can be extended with different laboratory environments and also the single entities of the results can be used independently. For example, the tracking software can be used alone to track users' progress in a cyber-security exercise. Appendix 7 illustrates one example of how to use and extend the results, a situation where two different instances of the created laboratory are being guided and graded in the Moodle platform and a cyber-security exercise is also tracked by the tracking software and the learning platform. In the appendix, the term ”Moodle” represents the created learning platform, the term ”putsilab” represents the created laboratory environment and the ”flagproxy” and ”flagclient” terms represent different executables of the created tracking software.

In the further development of the laboratory environment, automated modification of the sensitive information and user credentials of the laboratory environment should be implemented

so that the students cannot share this information. Also, an simulation automation that simulates users of a fictional company should be created by using e.g. JYVSECTEC botnet. Additionally, different versions of the devices of the laboratory environment should be created. In these versions the vulnerabilities and services should be modified. Therefore, by changing between these different versions of the virtual machines, new different laboratory environments could be easily created. In the further development of the tracking software it should be integrated into JYVSECTEC botnet. By using this combination, malicious and non-malicious traffic could be automatically created to the laboratory environment when the user reaches a certain point in the laboratory scenario. The Moodle-platform's further development should mostly be about adding different courses, exams, assignments and theory material. Also, hints specific to certain virtual machines should be implemented and developed so that when the user uses a hint, Moodle automatically reduces points from the user's exam.

REFERENCES

2014 Data Breach Investigations Report (DBIR). Report. Referenced 10.1.2015. http://www.verizonenterprise.com/DBIR/2014/reports/rp_Verizon-DBIR-2014_en_xg.pdf

About IPFire. Referenced 1.2.2015. http://www.ipfire.org/features.

About Moodle. Wiki-page. Referenced 25.2.2015. https://docs.moodle.org/28/en/About_Moodle.

About the Metasploit Meterpreter. Website by Offensive Security. Referenced 6.3.2015. http://www.offensive-security.com/metasploit-unleashed/About_Meterpreter.

Access Tokens (Windows). Referenced 1.2.2015. https://msdn.microsoft.com/en-us/library/windows/desktop/aa374909%28v=vs.85%29.aspx.

Application Programming Interface (API). Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/24407/application-programming-interface-api.

Avoid security by obscurity. Wiki-page of the OWASP. Referenced 6.3.2015. https://www.owasp.org/index.php/Avoid_security_by_obscurity.

Blind SQL Injection. Wiki-page. Referenced 1.2.2015. https://www.owasp.org/index.php/Blind_SQL_Injection.

CSV – Comma Separated Values. Referenced 6.3.2015. http://data.okfn.org/doc/csv.

Definition of a Security Vulnerability. Referenced 20.2.2015. https://msdn.microsoft.com/en-us/library/cc751383.aspx.

DVWA – Damn Vulnerable Web Application. Website of DVWA. Referenced 19.2.2015. http://www.dvwa.co.uk/.

Engebretson Patrick. 2014. The Basics of Hacking and Penetration Testing. Second edition. Waltham, USA: Syngress.

Exploitation - The Penetration Testing Execution Standard. Referenced 21.2.2015. http://www.pentest-standard.org/index.php/Exploitation.

Explore Terms: A Glossary of Common Cybersecurity Terminology. Referenced 6.3.2015. http://niccs.us-cert.gov/glossary.

Intelligence Gathering - The Penetration Testing Execution Standard. Referenced 20.2.2015. http://www.pentest-standard.org/index.php/Intelligence_Gathering.

Internet Security Center – Definitions. Referenced 20.2.2015. http://usa.kaspersky.com/internet-security-center/definitions.

Internet Security Threat Report 2014. 2014. Security response publication. Referenced 10.1.2015. http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_v19_21291018.en-us.pdf

Introduction - Metasploit Unleashed. Website by Offensive Security. Referenced 6.3.2015. http://www.offensive-security.com/metasploit-unleashed/Introduction.

ISO/IEC/IEEE 24765 Systems and software engineering — Vocabulary. 2010. ISO-Standard published on 15.12.2010. Referenced 6.3.2015. http://www.iso.org/iso/catalogue_detail.htm?csnumber=50518.

JYVSECTEC – Jyväskylä Security Technology. Website of JYVSECTEC-project. Referenced 6.1.2015. HTTP://www.jyvsectec.fi/jyvsectec.

Kali Linux Tools Listing. Website of the Kali Linux. Referenced 21.1.2015. http://tools.kali.org/tools-listing.

Kissel, R. 2013. Glossary of Key Information Security Terms. Referenced 6.3.2015. http://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf.

Marinos L. 2013. ENISA Threat Landscape 2013. Report published at 11.12.2013. Referenced 10.1.2015. https://www.enisa.europa.eu/activities/risk-management/evolving-threat-environment/enisa-threat-landscape/enisa-threat-landscape-2013-overview-of-current-and-emerging-cyber-threats/at_download/fullReport.

McAfee Threat Glossary. Referenced 7.3.2015. http://www.mcafee.com/us/threat-center/resources/threat-glossary.aspx.

MELANI - Glossary. Online glossary page by Reportin and Analysis Centre for Information Assurance MELANI. Referenced 6.3.2015. http://www.melani.admin.ch/glossar/index.html?lang=en&action=id&id=260

Microsoft Terminology Collection. Referenced 7.3.2015. http://www.microsoft.com/Language/en-US/Terminology.aspx.

Moore, H.D. 2012. Metasploitable 2 Exploitability Guide. Website documentation created on 31.5.2012. Referenced 19.2.2015. https://community.rapid7.com/docs/DOC-1875.

M-Trends 2014 Threat Report. 2014. Report. Referenced 10.1.2015. https://dl.mandiant.com/EE/library/WP_M-Trends2014_140409.pdf

Mueller, R. 2013. Active Directory: Glossary. Referenced 6.3.2015. http://social.technet.microsoft.com/wiki/contents/articles/16757.active-directory-glossary.aspx

OWASP Top 10 – 2013. 2013. Online publication. Referenced 14.2.2015. http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013.pdf

picoCTF 2014 - Frequently Asked Questions. Website of picoCTF. Referenced 6.3.2015. https://picoctf.com/faq.

Post Exploitation - The Penetration Testing Execution Standard. Referenced 21.2.2015. http://www.pentest-standard.org/index.php/Post_Exploitation.

Pre-engagement - The Penetration Testing Execution Standard. Referenced 20.2.2015. http://www.pentest-standard.org/index.php/Pre-engagement.

Reporting - The Penetration Testing Execution Standard. Referenced 21.2.2015. http://www.pentest-standard.org/index.php/Reporting.

RGCE – Realistic Global Cyber Environment. Website of JYVSECTEC-project. Referenced 6.1.2015. HTTP://www.jyvsectec.fi/rgce.

Rouse, M. 2006. regular expression (regex). Referenced 6.3.2015. http://searchsoftwarequality.techtarget.com/definition/regular-expression.

Rouse, M. 2010. privilege escalation attack. Referenced 6.3.2015. http://searchsecurity.techtarget.com/definition/privilege-escalation-attack.

Sidhpurwala Huzalfa. 2014. Bash specially-crafted environment variables code injection attack. Blog-post on security blog of Redhat published on 24.9.2014. Referenced 31.1.2015. https://securityblog.redhat.com/2014/09/24/bash-specially-crafted-environment-variables-code-injection-attack/.

Standard Output Definition. Web page of The Linux Information Project created on 10.2.2005. http://www.linfo.org/standard_output.html.

Sudo in a Nutshell. Website of the sudo. Referenced 6.3.2015. http://www.sudo.ws/sudo/intro.html.

The Penetration Testing Execution Standard. Referenced 20.2.2015. http://www.pentest-standard.org/index.php/Main_Page.

Threat Modeling - The Penetration Testing Execution Standard. Referenced 20.2.2015. http://www.pentest-standard.org/index.php/Threat_Modeling.

TrueCrypt - Introduction. Un-official mirror of the Truecrypt documentation. Referenced 6.3.2015. http://andryou.com/truecrypt_orig/docs/.

Vulnerability Analysis - The Penetration Testing Execution Standard. Referenced 20.2.2015. http://www.pentest-standard.org/index.php/Vulnerability_Analysis.

Vulnerability Details: CVE-2013-2094. CVE vulnerability listing published at 14.5.2013. Referenced 31.1.2015. www.cvedetails.com/cve/CVE-2013-2094/.

Web for Pentester. Referenced 1.3.2015. https://www.pentesterlab.com/exercises/web_for_pentester/course

What is Bourne Again Shell (Bash) - Definition from Technopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/3520/bourne-again-shell-bash.

What is Camelcase - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/16891/camelcase.

What is Cascading Style Sheets Level 1 (CSS1) - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/24394/cascading-style-sheets-level-1-css1.

What is Comma-Separated Values File (CSV) - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/24364/comma-separated-values-file-csv.

What is Common Gateway Interface (CGI) - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/4865/common-gateway-interface-cgi-web-development.

What is Document Object Model (DOM) - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/945/document-object-model-dom.

What is Environment Variable - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/15664/environment-variable.

What is Garbage Collection (GC) - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/1083/garbage-collection-gc-general-programming.

What is JavaScript Object Notation (JSON) - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/3930/javascript-object-notation-json.

What is Kali Linux?. Documentation. Referenced 21.1.2015. http://docs.kali.org/introduction/what-is-kali-linux.

What is LAN Manager Hash (LANMAN Hash) - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/28604/lan-manager-hash-lanman.

What is Logical Unit Number (LUN) - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/321/logical-unit-number-lun

What is Security Through Obscurity (STO) - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/21985/security-through-obscurity-sto.

What is Server Message Block (SMB) - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/5470/server-message-block-smb.

What is Stress Testing - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/15310/stress-testing.

What is Structured Query Language (SQL) - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/1245/structured-query-language-sql.

What is Virtual Machine (VM) - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/4805/virtual-machine-vm.

What is Windows NT LAN Manager (NTLM) - Definition from Techopedia. Techopedia dictionary. Referenced 7.3.2015. http://www.techopedia.com/definition/4169/windows-nt-lan-manager-ntlm.

APPENDICES

APPENDIX 1. ORIGINAL ASSIGNMENT

This appendix contains the original written assignment of the thesis. Rest of the requirements and objectives were formed during the development and production process based on the conversations with JYVSECTEC.

Toimeksianto

Toimeksiantaja

Jyväskylän ammattikorkeakoulu Oy / JYVSECTEC projekti.

Tehtävän kuvaus

Opiskelija tuottaa harjoitusympäristön tietoturvatestaustyökalujen opetusta varten. Harjoitusympäristö sisältää useita testikohteita kuten web-sovelluksia ja palvelimia, joihin tehdään sekä konfiguroidaan haavoittuvuuksia. Harjoitusympäristö tulee dokumentoida tarkasti.

Tavoitteet

Tutkia, suunnitella ja toteuttaa harjoitusympäristö tietoturvatestausohjelmistolle. Dokumentoi harjoitusympäristöt tarkasti (tunnetut haavoittuvuudet, luodut haavoittuvuudet).

Appendix 2. Attack path inside the laboratory environment

This appendix contains confidential information and is not included in the public version of the thesis.

Appendix 3 Creating a Moodle-course with flag-software integration.

This appendix contains confidential information and is not included in the public version of the thesis.

Appendix 4. Laboratory virtual machine specifications.

This appendix contains confidential information and is not included in the public version of the thesis.

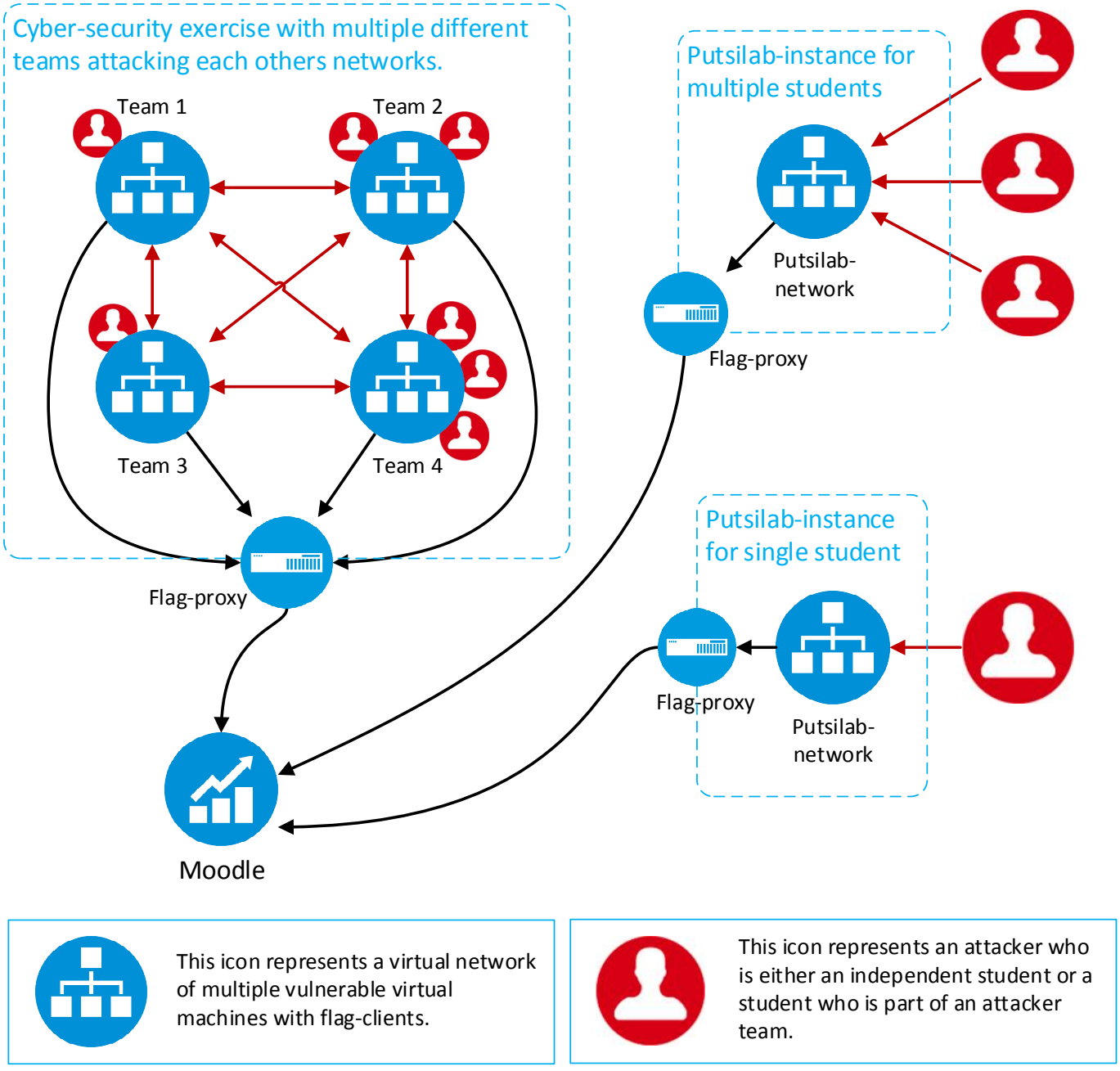
Appendix 5. Scenario walkthrough

This appendix contains confidential information and is not included in the public version of the thesis.

Appendix 6. Python exploit-script for created chat-server.

This appendix contains confidential information and is not included in the public version of the thesis.

Appendix 7. Example of an extended environment



Appendix 8. Architecture, security measures and usage of the tracking software

This appendix contains confidential information and is not included in the public version of the thesis.

Appendix 9. Architecture, security measures, vulnerabilities and usage of the laboratory environment

This appendix contains confidential information and is not included in the public version of the thesis.

Appendix 10. Architecture, security measures and usage of the virtual learning platform

This appendix contains confidential information and is not included in the public version of the thesis.