Towards a Unified Qualitative Usable Security Assessment Framework for Cloud Computing

Master's Thesis of

Oliver Rudzinski

at the HECTOR School of Engineering and Management
Information Systems Engineering and Management

for the Institute for Program Structures and Data Organization (IPD)

in collaboration with Hewlett Packard Enterprise

Reviewer:	Prof. Dr. Ralf Reussner
Second reviewer:	Dr. Robert Heinrich

01. October 2022 – 31. March 2023

HECTOR School of Engineering & Management
℅ International Department of the KIT gGmbH
Schlossplatz 19
76131 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself. I have submitted neither parts of nor the complete thesis as an examination elsewhere. I have not used any other than the aids that I have mentioned. I have marked all parts of the thesis that I have included from referenced literature, either in their original wording or paraphrasing their contents. This also applies to figures, sketches, images and similar depictions, as well as sources from the internet.

Berlin, October 18, 2022

. . . .

(Oliver Rudzinski)

Abstract

English abstract.

Zusammenfassung

Deutsche Zusammenfassung.

Table of Contents

List of Acronyms

API	Application Programming Interface
AWS	Amazon Web Services
CC	Cloud Computing
CLI	Command Line Interface
GCP	Google Cloud Platform
HCD	Human-Centered Design
HPE	Hewlett Packard Enterprise
IaaS	Infrastructure as a Service
IAM	Identity & Access Management
ISO	International Organization for Standardization
IT	Information Technology
NIST	National Institute of Standards and Technology
OWASP	Open Web Application Security Project
PaaS	Platform as a Service
SaaS	Software as a Service
UI	User Interface

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Cybersecurity has been and still remains *the* most crucial discipline in the domain of Information Technology (IT): According to Mordor Intelligence, the global cybersecurity market was valued at approx. $150 billion in 2021. This trend is expected to continue, rising up to approx. $317 billion by the year of 2027 with a constant annual growth rate of 13.4% [12]. This can be linked to the average cost of a single data breach equaling $4.24 million in the same year and describing the highest value on record so far, as per IBM [11]. These rises are proportional to the increase in cyber attacks: Malware attacks increased by 358% in 2020, Ransomeware attacks rose by 435% [30].The current landscape of cybersecurity can be summarized as an arms race between the attackers of valuable resources and their protectors.

Looking at the domain of Cloud Computing (CC), 27% of organizations report having experienced security incidents within one year [48]. Overall, the fear of security challenges is the main concern why some businesses do not introduce CC at all [1, 2].

Businesses investing in sophisticated security technology and services might not be enough, as 95% of cybersecurity issues can be linked to human error [32]. One reason for this might be the increased complexity of security systems and settings, leading to misconfigurations that in turn lead to security vulnerabilities. In CC, misconfiguration in the cloud infrastructure accounted for 23% of security incidents [48]. It can be assumed that the increased complexity of IT security features reduces its usability, resulting in a decrease of their efficiency.

The research discipline of *usable security* has recognized this issue and aims at bringing effective IT security together with information system usability [21]. Considered separately, there exist several design and assessment frameworks for security aspects within information systems, e.g., the Security Design Principles by Saltzer and Schröder (1975) [46], the Security By Design Principles according to the Open Web Application Security Project (OWASP), or the Cybersecurity Framework by the National Institute of Standards and Technology (NIST) [20]. On the other hand, usability assessments, e.g., the Usability Heuristics by Nielsen (1990) [39], or usability principles and guidelines support the evaluation of the user friendliness of User Interfaces (UIs) [35]. These existing frameworks will hereinafter be referred to as *current frameworks*.

However, usable security research as a whole is considered complicated due to interdisciplinary factors, technological velocity, etc. [21]. That is why it is currently conducted based on individual definition and execution of experiments rather than supported by guidelines or frameworks. This can lead to slower examination of the problems at hand, consequently leading to fewer insights on how to potentially address them.

This is where this proposed Master's thesis ties in: It suggests a unified framework for qualitative usable security assessment that will be developed and evaluated experimentally within the security-crucial domain of CC. This will hereinafter be referred to as the *unified framework*.

1.2 Problem Statements

The previous motivation to the topic (cf. Section 1.1) as well as further demonstration of current research in that area (cf. Chapter 3) recognize the following problems:

P1 Usable security research is currently conducted on a case-by-case basis which limits efficiency and quantity of insights in order to address problems within that domain.

P2 In terms of CC, security configuration aspects are currently complicated to consider in terms of usable security, which visibly leads to a significant amount of security incidents.

1.3 Contribution

The problems stated in Section 1.2 are aimed at being addressed through the following aspired contributions. While these contributions are build on one another, they are also designed to individually undertake issues within current usable security research.

C1 Evaluate the applicability of present usability and security design frameworks to usable security individually to the domain of CC in order to deduct limitations and improvements.

This contribution will partly address P1, as there is limited research present regarding the explicit applicability of usability and security design frameworks to usable security. Thus, it cannot be justified that these frameworks are not applicable per se. This contribution's output will generate the input for C2 in order to have a foundation for the design of the unified framework. As this assessment will be pracitically performed within a CC environment, this contribution also partly covers P2.

C2 Design a unified qualitative usable security assessment framework based on the findings from C1.

This contribution will mainly address P1 by applying the findings from C1 in order to develop a unified framework for usable security evaluation. It is expected that certain aspects of the frameworks under test in C1 will appear in the unified framework, while others might be refined or newly introduced.

C3 Validate the unified framework regarding improvements in terms of applicability, conclusivity, and efficiency compared to C1.

This contribution, again, addresses P1 and P2 in that the actual benefit of the new framework needs to be validated within a CC security feature scope.

All in all, P2 will be addressed across all contributions as each practical application will be done in terms of CC security features. The actual methodology for the achievement of the contributions will be outlined in Chapter 4.

1.4 Thesis Structure

Add correct thesis structure.

2 Foundations

This chapter aims at providing an overview on the necessary background information and technical foundations required to perform the desired research. It is expected that these foundations will be elaborated in the actual thesis as certain examined aspects of the research might require additional background. The design of this chapter is not meant to be exhaustive to each area but shall only provide the information that is strictly required for the sake of this work.

2.1 Information Security

The U.S. National Institute of Standards and Technology (NIST) defines information security as "the protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to ensure confidentiality, integrity, and availability" [36]. The last three characteristics can be further defined:

Confidentiality Unauthorized entities are restricted from access and disclosure of information for the purpose of protecting personal privacy and proprietary information [36].

Integrity Improper modification or destruction of information, including the compromitation of data non-repudiation and authenticity, is protected against. Furthermore, *data integrity* means that data has not been improperly altered, whereas *system integrity* goes beyond unautorized manipulation, defining system function quality when performing as intended and unimpaired [36].

Availability Accessibility and usability of information is reliable and timely [36]. This also includes the prevention of intended, unauthorized withholding of data [19, p. 36].

This *CIA triad* can be extended by additional characteristics, e.g. *accountability* (user actions can be traced back to them), *functionality* (system's behavior is as intended and expected) and *reliability* (system always performs under equal conditions) [19, p. 36 sq.].

2.1.1 Distinction from Cyber Security

The terms *information security* and *cyber security* (i.a.), are often used interchangeably in literature. However, this work requires a specific distinction between those terms. As defined by the NIST Cybersecurity Framework, cyber security means "the ability to protect or defend the use of cyberspace from cyber attacks" [20]. By that, cyber security is a disciplinary subset of information security that especially considers the electronic protection of data and resources within its technical perimeter. Consequently, this excludes activities like physical access or compliance control that are inherent information security disciplines. Nevertheless, both areas focus on achieving the goal of the information security definition by ensuring the application of the CIA tiad.

2.1.2 Security Design

As motivated in Chapter 1, cyberattacks are still on the rise, which leads to assume that the current state of the art of security design and implementation cannot systematically exclude security flaws. This has already been recognized by Saltzer and Schröder (1975) who summarized general security design guidelines that address general security problems during the design of information systems [46]. They include best practices that aim at considering the CIA triad from the software perspective.

It appears to support security design aspects during the actual design and implementation. There exist similar design frameworks in that domain, e.g., the OWASP Security By Design Principles.

Finally, there exist distinct security assessment and improvement frameworks for information systems that are already in place, such as the NIST Cybersecurity Framework. It defines a process model for the evaluation of the current state, a tiered readiness assessment, as well as transition profiles to reach a desired state of cybersecurity [20].

2.2 Usability of Information Systems

Information system usability is part of ergonomics of human-system interaction, standardized by the International Organization for Standardization (ISO) [14]. It defines *usability* as the "extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" [14].

The interaction between the user and the system is done through a User Interface (UI), defined by ISO as the "set of all the components of an interactive system that provide information and controls for the user to accomplish specific tasks with the interactive system" [14].

While usability design and evaluation are often applied to UIs, these processes could also apply to less end-user-focussed interfaces, e.g., Application Programming Interfaces (APIs) or Command Line Interfaces (CLIs).

2.2.1 Usability Design Guidelines

The process of providing *good* system usability to users is aided by several guidelines and frameworks: The ISO defines the Human-Centered Design (HCD) process, which describes an "approach to systems design and development that aims to make interactive systems more usable by focusing on the use of the system and applying human factors/ergonomics and usability knowledge and techniques" [15].

HCD is an iterative approach and includes the understanding and requirement of the context, as well as the cooperative development and evaluation of design solutions [15]. The latter steps are aided by potential users of the system under design.

HCD is particularized through further guidelines developed by notable researchers that define general principles on usability design. These include, but are not limited to, Shneiderman's Eight Golden Rules of Interface Design [50], Norman's Seven Design Principles [40], and Nielsen's Ten Usability Heuristics for User Interface Design [39]. All of those frameworks follow a similar structure in that they specifically ask for the existence or absence of characteristics within the interface, e.g., visibility of things, consistency of design, etc.

2.2.2 Usability Evaluation

The evaluation of usability is often referred to *usability testing* [35]. Its goals are to identify problems, uncover opportunities as well as learn about the target user's behavior and preferences [35]. Elements of usability testing include participants (i.e., realistic users of the system), specific tasks or activities performed by the participant, as well as a facilitator (i.e., moderator and creator of the specific activity) [35]. After performing the activities, the participants are expected to provide feedback to the moderator, whereas the moderator notes insights to the behavior of the participant and the system.

Usability evaluation can be either qualitative (i.e., focussing on distinct insights and findings with regards to user behavior) or quantitative (i.e., focussing on benchmarks, e.g., task success or elapsed time). They are usually conducted in a remote setting, however there also exists in-person usability testing. Finally, it can be performed in a moderated (i.e., interactive) or unmoderated manner (i.e., no interaction between the participant and the facilitator) [35].

The definition of tasks for a usability evaluation highly depends on the specific area under test. However, it can be expected that the usability design guidelines (cf. Subsection 2.2.1) can be taken into account.

2.3 Usable Security

Bringing it all together, *usable security* deals with the design, implementation, and evaluation of the usability of security features and aspects within information systems [45]. Chronologically, it originates from the already mentioned Security Design Principles by Saltzer and Schröder (1975), specifically the final one considering the "psychological acceptability" of the mechanisms [45, 46]. Later, Zurko and Simon (1996) novelly suggested to apply usability aspects of software design to security features [63]. Shortly after, Adams and Sasse (1999) researched usability flaws in terms of password protection from a user perspective and provided recommendations on remedial measures [4]. Since then, the majority of usable security research is focussed on end-user security in terms of password protection and email encryption [45, 61]. More recently, this field also started to consider developers and IT professionals as their usability issues in terms of security can lead to severe security vulnerabilities [9].

Other than with security *or* usability evaluation, there do not exist established unified frameworks or guidelines for usable security assessment. Some more recent studies suggest certain approaches that will be discussed in Chapter 3. Other than that, usable security is currently performed by means of individual experiments [21].

2.4 Cloud Computing

The NIST defines CC as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (...) that can be rapidly provisioned and released with minimal management effort (...)" [33]. It can be seen as the latest form of infrastructure consolidation after the introduction of client-server computing and the rise of datacenter hardware virtualization [49, p. 1].

In general, CC is described by a vendor-consumer relationship, with the vendor being the provider of the IT resources, and the consumer being the receiver and user of those resources. Depending on the deployment model of the CC infrastructure at hand, the vendor can be

- internal to an organization or providing isolated resources exclusively to that organization (i.e., *private cloud*),
- external, providing resources to multiple entities (i.e., *public cloud*),
- or a mix of both [33].

The service level defines the level of abstraction of the resources to the consumer and which party is responsible for which layer of application provisioning and deployment. Typically, this includes Infrastructure as a Service (IaaS) (vendor provides virtualized infrastructure), Platform as a Service (PaaS) (vendor provides virtualized infrastructure as well as the specific operating system, runtime environment, etc. to that platform), and Software as a Service (SaaS) (vendor provides infrastructure, platform, and specific software deployment) [33].

The NIST laid out several aspects that further define CC characteristics, service, and deployment models.

On-demand self-service Provisioning of (more) computing, storage, network resources, etc. is done by the consumer without interaction with the CC service provider [33]. This implies a form of automation on the CC vendor's end as well as the existence of control and monitoring interfaces for the consumer [49, p. 51].

Broad network access The CC vendor's services and capabilities are accessible over the network and can be accessed via mechanisms that are already in place for consuming clients [33, 49, p. 45].

Resource pooling Resource pooling allows CC vendor's to introduce the concept of *multi-tenancy*, where the physical infrastructure that is made available is virtualized and dynamically assigned to distinct consumers, or *tenants*, based on demand or contractual agreements [33, 49, p. 45]. In other words, a *pool* of total resources present is made available to multiple consumers based on their *current* needs and re-evaluated over the course of the usage.

Rapid elasticity The consumer can dynamically scale their resource capacity up or down as needed. The consumer is not aware of any global resource limitations which make the CC resources appear unlimited [33].

Measured service The consumed services are quantitatively measured by the CC vendor for billing, controlling and optimization purposes. This is based on appropriate forms of metering with regard to the service (e.g., number of terabytes of storage consumed within a month). Thus, billing of utilized services is often done on a pay-per-use basis [33]. Consequently, the more resources are utilized by the vendor, the higher the cost. This process is transparent to both vendor and consumer, allowing for consumption and billing monitoring, control, and reporting [33].

3 Related Work

As mentioned in Section 2.3, the discipline of usable security has recently gained in importance. Therefore, it is of interest, which aspects of this underlying work are already present in current research, and which aspects of interest have currently not been addressed. The discussion of related work will prove valuable for the identification of current research limitations that need to be addressed by the contributions of this work. As with Chapter 2, it is expected that the consideration of related work might be enlarged during the actual research.

3.1 Present Research

3.1.1 General Research on Usable Security

Starting with a critical look at usable security research, Theofanos (2020) points out that, even though this discipline has been around for a quarter of a decade, there have been little to no achievements to the underlying problem of so-called *security fatigue* observed in user behavior. She points out the major areas where usable security might make a difference (e.g., authentication, encryption, user dialogs, etc.) and appeals to developers and researchers to make this field of study more applicable and practical [55]. The arguments and appeals support this proposed work.

More concretely, Acar et al. (2016) recognize the problem of this research field being almost entirely end-user-centric up until now. They suggest specific measures and key research questions in order to consider developers' and IT professionals' points of view in that domain. This includes the evaluation of interfaces outside of end user scope, introducing secure usability research towards APIs, as well as providing relevant security resources for the development process [3]. Similar claims are made by Green and Smith (2016). They, again, demand more emphasis on secure and usable APIs [22]. These claims support the validity of this work's research topic and will be of great guidance pursuing the goal of developing a qualitative usable security assessment framework.

Chiasson et al. (2007) furthermore propose general design principles to make administrator interfaces more psychologically graspable [10]. These were not validated by the authors during research, but can be used as knowledgable input to the design of this work's assessment framework.

Finally, Lennartsson et al. (2020) analyzed current research directions of usable security more thorougly than possible in this work. Apart from other findings, they recognized the urge of "developing concrete guidelines for implementation of user-centric security" [29], which, again, validates this work's purpose.

3.1.2 Towards Usable Security Frameworks

Parkin et al. (2010) recognized the psychological distance between technology-driven password policies in contrast to their users and what negative effect this could have on businesses' security. They essentially suggest an HCD-driven approach that includes users of the security mechanism and their feedback throughout its entire design phase [42]. However, this approach ties in with how usable security design and evaluation is currently conducted in terms of individual experiments. Plus, the evaluation and enhancement of features potentially in place is missing here.

Similarly, Feth et al. (2017) extended the individual steps of the HCD process to include security awareness from the start. They applied their model exemplarily to a smart home use case. Apart from mentioning the models abstractness, they realize their uncertainty regarding the model's applicability to other use case domains [18]. Furthermore, the research appears to combine security and privacy-related aspects. It also leads to assume that it is rather end-user-focussed which ties in with the majority of current usable security research.

This looks differently when considering research performed by Caputo et al. (2016) who manufactured interview questionnaires for different stages of security feature development and introduction (i.e., development, product, and management level). They then performed three case studies validating the applicability of their methodology and falsified some hypotheses from the beginning (e.g., that complicated security measures are *always* more secure, that usability is common sense, and that there *must be* a tradeoff between usability and security) [8]. Although very valuable for further research, the questionnaires provided are very abstract in that their development does not appear to be based on specific domains nor on current usability or security evaluation frameworks. It also puts a large emphasis on management perception as well as benchmarks in the *process* of development and introduction, rather than on the development or evaluation itself.

Conversely, Ambore et al. (2021) present an evaluation approach that is inspired by usability evaluation heuristics. Specifically, they derive heuristics from general usability assessment and refine as well as apply them to the domain of financial technology. They validate their approach by performing usable security assessment in financial organizations and conclude that such evaluation, in fact, improved usable security of the evaluated applications [6]. This research can be considered as a starting point for the development of this work's unified framework for the different application domain of cloud computing.

Another approach of usable security evaluation is presented by Hausawi and Allen (2015). They introduce a general quantitative evaluation approach in terms of a risk assessment for usable security. The output of the assessment is a numerical value that aims at classifying the overall risk of present or non-present usable security in terms of an application under test [24]. While this approach appears very general and thus applicable to various domains of interest, it might add an additional burden to the process as it is solely quantitative and does not take individual aspects of usable security into account.

3.1.3 Research on Cloud Security

Understanding current security-related CC research and its underlying or discovered problems might aid in moving this work's practical approach in a relevant direction.

A variety of researchers claims that the aspects of multi-tenancy [23, 52, 43, 31], extensive virtualization [17, 23], and Identity & Access Management (IAM) [5] present the most critical security concerns across CC infrastructure management, regardless of their deployment of service models [31]. This leads to assume that general features implementing inherent characteristics of CC are generally implemented similarly. Consequently, they also share similar security issues.

In slight contrast, Yandong and Yongsheng (2012) mention that public cloud vendors might be more susceptible to cyberattacks as it is known that they store a vast amount of data that could be leveraged for malicious activities [62]. While it can be argued, that many cloud security issues cannot solely be tackled by improving their security feature usability, an understanding of the criticality of those features can support prioritization of validation criteria when designing a unified usable security assessment framework.

3.1.4 Towards Usable Security in the Cloud

Research on usable security in the domain of CC is strongly limited. However, Fahl et al. (2012) propose a way to introduce usable security in terms of CC management. They argue that the current way of securing CC resources is done via public key infrastructures that are perceived to be very complicated from an end-users point of view. They propose a novel form of user-friendly security mechanism that removes the perceived burden in the form of a distinct security service [16]. While it appears valuable that entirely new mechanisms are researched and developed with the primary goal of usability, this research does not consider the improvement of current established systems by enhancing their usability. The reinvention of security mechanisms should be the last resort, as it merely shifts responsibility from one end to another. This, in turn, might lead to entirely new security concerns.

3.2 Achievements & Limitations

Based on the findings of the evaluation of the related work to that topic, it can be seen that a lot of effort has been put into discussing and partially solving usable security issues. Specifically, the following general research achievements can be summarized:

A1 Recognition of the need for more usable security consideration across all human-facing security features, especially outside of end-user applications.

A2 Following from A1, the specific usable security consideration under the domain of CC.

A3 Recognition of prime security issues within the domain of CC outside of the scope of usable security.

A4 Recognition of more efficient usable security assessment.

A5 Following from A3, initial development and partial evaluation of such frameworks under limited scope and generalization.

Following a process of exclusion, the following limitations to that research can be derived:

L1 Despite A1, there is currently little effort to focus on developers and IT professionals in terms of usable security enablement.

L2 Despite A2 and A3, the efforts in terms of usable security in the domain of CC are limited to reinventing security features rather than assessing and improving them from a usable security point of view.

L3 Despite of A4 and A5, there currently does not exist a unified, qualitative usable security assessment framework that CC could benefit from.

With that, the derived limitations can be linked to the problem statements (cf. Section 1.2) and proposed contributions (cf. Section 1.3) of this work, which will be done in the following section.

3.3 Application of Contributions

The following relationship of problems, contributions, as well as current research's achievements and limitations shall finally justify the direction and scope of this work:

- **P1** [tedious usable security evaluation] might benefit from addressing
  L3 [lack of unified assessment framework],
  which will be achieved by sequentially performing
  C1 [applicability evaluation of current frameworks],
  C2 [design of unified framework], and
  C3 [validation of unified framework], aided by input given by
  A5 [suggested frameworks from current research].

- **P2** [CC security incidents due to misconfiguration] might benefit from addressing
  L2 [lack of usable security assessment of CC security features], and inherently
  L1 [little consideration of IT professionals within usable security research],
  which will be achieved by the CC scope inside
  C1 [see above] and
  C3 [see above], aided by input given by
  A2 [initial consideration of CC usable security] and
  A3 [prime security challenges within CC].

The specific approach of solving the mentioned problems through the aspired contributions will be outlined in Chapter 4.

4 Approach

This chapter outlines the necessary steps and methodology that is applied in order to achieve the aspired contributions in terms of the desired research goal of this work.

In essence, the underlying research methodology will follow an experimental and empirical approach that aims to (i) find limitations and challenges in current frameworks, (ii) develop a unified framework that addresses these issues, and finally (iii) validate its applicability and overall benefit after the fact. Each aspect is prepared and explained in the following, whereas the explicit evaluation approach will be introduced in Chapter 7.

Optional: Add Design Science Research Approach.

4.1 Sampling Approach

Before pursuing the first contribution of this work, the research environment and scope need to be prepared. Chapters 1 and 3 already established the overall domain under this research will take place, which is Cloud Computing. For that matter, a sample of appropriate CC offerings and services must be derived in order to have a finite scope of consideration. Similarly, this is also the case for the choice of currently used frameworks that solely consider *either* usability *or* security design or evaluation.

4.1.1 CC Vendor and Services Sampling

There exists a large number of independent CC vendors, all offering another vast landscape of numerous CC services. Due to limitations in time and scope, the goal is to have three CC offerings, each providing five comparable, thus similar, CC services. This results in up to 15 independent service executions that can be put under test for current framework assessment as well as for validation of the unified framework.

The choice of the vendor and service samples is based on quantitative indicators and will be defined and practically applied in the following. Additionally, in preparation of the subsequent contributions of this work, the total sample set is split into subsamples for distinct assessment (cf. C1) and validation (cf. C3) purposes. This will allow for an independent assessment of current frameworks as well as a controllable validation of the unified framework to be designed.

4.1.1.1 Generalized Approach & Rationale

The utmost goal of the practical research environment is to be as generalizable and close to actual CC usage as possible. Therefore, the choice of CC vendors for this work is based on their individual market share in their area of business. This ensures that the the considered vendors are relevant to the industry, thus having a significant number of users that might benefit from usable security assessment. Plus, from a technical point of view, it can be expected that major vendors offer a variety of services, ensuring that popular services across these vendors are comparable in functionality.

With that, the actual selection of CC candidates can commence. Here, the respective criteria for a justified acceptance or rejection are as follows:

Popularity across CC Vendors The service candidate must be frequently used within its particular vendor. The general underlying service must be popular across all considered CC vendors.

Comparability across CC Vendors The underlying CC service of a candidate must be available across all considered vendors.

Significant Security Relevance The configuration of the service candidate must be presented with significant security relevance such that the user has an influence on its particular security settings. This can include, but is not limited to, multi-tenancy, IAM, or virtualization.

Free Tier Availability The research to this work is not financially supported by any means. The usage of the service candidate, potentially with a limited feature set, must be available free of charge at least for the duration of this research.

The latter criteria can be atomically answered. However, there does not exist a general popularity index to each vendor, its services, or CC services in general. Therefore, the popularity analysis is done manually and enriched by the acceptance criteria from above as follows:

1. Determine n articles that list most popular CC services for each considered vendor. Disregard lists that are not ordered by assumed popularity.

2. For each article, list the position of every rated service. Only consider the top ten listings, and disregard services that build on top of each other, as well as consolidated services.

3. Link ranked services across the considered vendors. Disregard all services that do not appear in every ranking.

4. Determine average service popularity for each vendor. Begin with services that appear in all n articles, then in $(n-1)$ articles, $(n-2)$, and so on.

5. Determine overall popularity, weighing individual averages by the corresponding vendor's market share overhead.

6. Disqualify each remaining service that does not offer usage free of charge.

7. Disqualify each remaining service that does not present itself with significant security relevance.

8. Choose top five services from the remaining services by their overall popularity ranking.

Referring to the previously mentioned number of three vendors to be considered, it must be stated that this work requires the consideration of a particular cloud services vendor, namely Hewlett Packard Enterprise (HPE) and their as-a-service offering *HPE GreenLake*. It can be expected that this particular offering will not withstand the criterion of highest marketshare within CC. However, this deviation from the sampling approach is not expected to significantly distort the research, as the goal of sampling comparable services across all vendors remains. Additionally, the choice of particular services will solely be done based on the vendors that hold significant marketshare and linked to HPE GreenLake after the fact.

4.1.1.2 Explicit Sample Determination

The generalized CC vendor and service sampling approach is now applied explicitly.

CC Vendor Determination by Market Share The CC vendor market share determination is provided by Synergy Research Group [27]: With 33% of the entire market share in the first quarter of 2022, *Amazon Web Services (AWS)* present themselves as the most relevant cloud provider, followed by *Microsoft Azure* with 22% and *Google Cloud Platform (GCP)* with 10% for the same period. This concludes the vendor choice, resulting in AWS and Azure for following consideration, in addition to HPE GreenLake, which was predetermined. AWS and Azure will now be assessed in terms of potential CC service candidates.

Service Popularity Ranking Articles Table 4.1 lists all articles respective to the individual CC vendor that will be analyzed in terms of service popularity. The variety of articles mitigates the occasional lack of objective quality within individual articles. Articles [7] and [28] cannot be considered as they do not sort services according to their perceived popularity.

Amazon Web Services		**Microsoft Azure**	
Reference	Article Provider	Reference	Article Provider
[34]	SSI Insights	[58]	IntelliPaat
[13]	Jefferson Frank	[41]	RLogical
[26]	All Code	[47]	WhizLabs
[60]	Mindmajix	[59]	TestPrepTraining
[54]	Boxpiper	[57]	EcoCloud
[56]	ClickIT Tech	*[28]*	*DevOpsSchool*
[7]	*IntelliPaat*		

Table 4.1: CC Service Popularity Ranking Articles by Vendor

Nominal Service Popularity Ranking by Vendor Table 4.2 lists each CC service by vendor, alongside its nominal popularity ranking according to the corresponding article rating. Each table only considers the top ten services per article. For Microsoft Azure, each article considers the Azure *DevOps* service within their top ten. However, this service must be disregarded as it is a consolidated service, containing multiple services that can also be consumed independently. This is why the Azure table only counts up to nine services per rating article. Furthermore, [13] and [26] list AWS *EC2 Auto Scaling* as a separate service. Again, this is disregarded as it builds on top of the regular EC2 service.

	Rank /10 in Article					
AWS Service	[34]	[13]	[26]	[60]	[54]	[56]
EC2	1	1	1	1	1	1
RDS	2	5	2	5	2	2
S3	3	3	3	2	5	3
CloudFront	4		5	8	9	4
VPC	5		8	7		5
SNS	6	7	6			6
Beanstalk	7			9	6	7
Lambda	8	6	4	6		8
Autoscaling	9			10		9
IAM	10					
EBS		2	7			
CloudWatch		4				
Cloud-Formation		8				
ELB		9				
DynamoDB		10		4		
Kinesis			9			
Aurora				3		
ElastiCache				10	4	10
ECS					3	
SQS					7	
Route 53					10	

(a) AWS Service Popularity Ranking

	Rank /9 in Article				
Azure Service	[58]	[41]	[47]	[59]	[57]
Blob Storage	1				
Virtual Machines	2	1	9	1	1
Backup	3	5	7	6	6
Cosmos DB	4	2	6	2	2
Logic Apps	5	5	8	5	7
Active Directory	6	3	1	3	3
API Management	7	4			4
Content Delivery Network	8	6	2	4	5
Site Recovery	9	9		9	8
Bots		8		7	9
Data Factory			3		
SQL			4		
Function			5		
Automation				8	

(b) Azure Service Popularity Ranking

Table 4.2: Nominal Popularity Ranking of CC Services by Vendor

Service Mapping Across Vendors The individual services from Table 4.2 are now linked in terms of their general CC service purpose. This eliminates a majority of services mentioned previously because vendor-specific services either do not have a counterpart with a different vendor or do not appear in both popularity listings. This results in Table 4.3.

General Service Term	*AWS* Service	*Azure* Service
Virtual Machines	EC2	Virtual Machines
Relational Database	RDS	SQL
Object Storage	S3	Blob Storage
Content Delivery	CloudFront	Content Delivery Network
Serverless Computing	Lambda	Functions
Access Management	IAM	Active Directory *(role-based)*
Process Automation	CloudFormation	Automation
NoSQL Database	DynamoDB	CosmosDB
Block Storage	*EBS*	*Blob Storage*

Table 4.3: CC Service Mapping between AWS and Azure

Additionally, is needs to be mentioned that AWS *Elastic Block Storage (EBS)* can only be partly applied to Azure *Blob Storage*, as the latter actually provides object storage capabilities. For AWS, this is already given by *S3*, which is why EBS is also disregarded.

Average Service Popularity Determination by Vendor For the remaining services, the average service popularity is now determined based on the popularity rating within the corresponding articles. The total vendor-specific rank is determined by the calculation of the average rank and the number of articles the service appeared in. Services that appear in all articles are considered first, followed by services that appear in all but one article, and so forth. This results in the ranking depicted in Table 4.4 on the next page.

	Rank /10 in Article						average rank calculation			
AWS Service	[34]	[13]	[26]	[60]	[54]	[56]	#	average	group	**rank**
EC2	1	1	1	1	1	1	6	1.00	1	**1**
RDS	2	5	2	5	2	2	6	3.00	1	**2**
S3	3	3	3	2	5	3	6	3.17	1	**3**
CloudFront	4		5	8	9	4	5	6.00	2	**4**
Lambda	8	6	4	6		8	5	6.40	2	**5**
DynamoDB		10		4			2	7.00	3	**6**
CloudFormation		8					1	8.00	4	**7**
IAM	10						1	10.00	4	**8**

(a) Average AWS Cloud Service Popularity

	Rank /9 in Article					average rank calculation			
Azure Service	[58]	[41]	[47]	[59]	[57]	#	average	group	**rank**
Virtual Machines	2	1	9	1	1	5	2.80	1	**1**
Active Directory	6	3	1	3	3	6	3.20	1	**2-3**
Cosmos DB	4	2	6	2	2	6	3.20	1	**2-3**
Content Delivery Network	8	6	2	4	5	5	5.00	1	**4**
Blob Storage	1					5	1.40	2	**5**
SQL			4			2	4.00	2	**6**
Functions			5			1	5.00	2	**7**
Automation				8		1	8.00	2	**8**

(b) Average Azure Cloud Service Popularity

Table 4.4: Average CC Service Popularity by Vendor

Total Service Popularity Determination The vendor-specific average popularity rankings are now used to calculate the general services' overall popularity. This is enriched by considering the larger marketshare of AWS, which is around 50% higher than Azure's. As a lower rank value represents a better rating, this is accounted for by multiplying the AWS rating by $\frac{2}{3}$ (i.e., the reciprocal of 1.5). This results in the rating table shown in Table 4.5 on the next page.

	AWS		Azure		total calc.	
Service	name	rank	name	rank	WAVG	**rank**
Virtual Machines	EC2	1	Virtual Machines	1	0.83	**1**
NoSQL Database	DynamoDB	6	Cosmos DB	2.5	3.25	**2**
Content Delivery	CloudFront	4	Content Delivery Network	4	3.33	**3**
Object Storage	S3	3	Blob Storage	5	3.50	**4**
Relational Database	RDS	2	SQL	6	3.67	**5**
Access Management	IAM	8	Active Directory	2.5	3.92	**6**
Serverless Computing	Lambda	5	Functions	7	5.17	**7**
Process Automation	CloudFormation	7	Automation	8	6.33	**8**

Table 4.5: Total Average CC Service Popularity

Service Disqualification Finally, the preliminary top five services by ranking need to be assessed in terms of free-tier availability and security relevance. All services but Azure *Content Delivery Network* are available for free, which is why content delivery in total cannot be considered further. This results in the following allocation, depicted in Table 4.6:

	AWS		Azure		total calc.	
Service	name	rank	name	rank	WAVG	**rank**
Virtual Machines	EC2	1	Virtual Machines	1	0.83	**1**
NoSQL Database	DynamoDB	6	Cosmos DB	2.5	3.25	**2**
Object Storage	S3	3	Blob Storage	5	3.50	**3**
Relational Database	RDS	2	SQL	6	3.67	**4**
Access Management	IAM	8	Active Directory	2.5	3.92	**5**
Serverless Computing	Lambda	5	Functions	7	5.17	**6**
Process Automation	CloudFormation	7	Automation	8	6.33	**7**
Content Delivery	CloudFront	4	Content Delivery Network	4	3.33	**DSQ**

Table 4.6: CC Service Ranking after Disqualification

From a security relevance perspective, all remaining top-five services present themselves with the necessity of security configuration in terms of multi-tenancy, access control, etc., which will be of value when assessing current frameworks and validating the unified framework. Therefore, no adaptation to this rating is required.

Link with HPE GreenLake services After choosing the overall services based on the two most relevant CC vendors by market share, these services can now be linked to available services within HPE GreenLake. This concludes the overall choice and mapping of CC vendors and services and is represented in Table 4.7.

Service	AWS	Azure	HPE GreenLake
Virtual Machines	EC2	Virtual Machines	Virtual Machines
Object Storage	S3	Blob Storage	Object Storage
Access Management	IAM	Active Directory	IAM
Relational Database	RDS	SQL	Microsoft SQL
NoSQL Database	DynamoDB	Cosmos DB	N/A

Table 4.7: Final Sample of CC Vendors & Services

It needs to be mentioned that, being a relatively new vendor in the area of CC, HPE GreenLake cannot provide every service that was determined in the previous steps. Specifically, HPE GreenLake does not offer any kind of NoSQL database service. This inconvenience will be accounted for when allocating vendor-service combinations into assessment and validation subsets.

4.1.1.3 Assessment & Validation Subset Allocation

The definition and allocation of subsets for assessment of current frameworks and validation of the unified framework is required to enable both disciplines with the available sample of CC vendor-service combinations. Specifically, the subsets fulfil the following goals:

Assessment Subset allow for a holistic, generalized assessment of current usability and security frameworks in order to gain insights for the design of the unified framework.

Validation Subset allow for a controlled validation of the unified framework within and outside its design space in terms of applicability, consistency, efficiency, as well as external robustness.

The allocation of the assessment subset should account for the entire service sample such that service-specific aspects are considered. In order to reduce the risk of overfitting to a specific vendor, this subset should additionally consider multiple vendors.

The validation subset should account for three types of evaluations, namely

- the validation of applicability and consistency within the design set and space (i.e., a control set),
- the validation of fidelity with real-world phenomena and robustness outside the design set, but still within the design space, and
- the validation of effectiveness, efficiency, and external consistency outside the design space.

AWS		Azure		HPE GreenLake	
service	purpose	service	purpose	service	purpose
EC2	A_1	Virtual Machines	$K_1 = A_6 \vee V_1$	Virtual Machines	V_6
S3	A_2	Blob Storage	$K_2 = A_7 \vee V_2$	Object Storage	V_7
IAM	A_3	Active Directory	$K_3 = A_8 \vee V_3$	IAM	V_8
RDS	A_4	SQL	V_4	Microsoft SQL	V_9
DynamoDB	A_5	Cosmos DB	V_5		

Table 4.8: Definition of Assessment, Control, and Validation Subsets

Considering these criteria in combination with the limited vendor-service combinations, the following subset allocation is defined in Table 4.8.

By designing the unified framework based on findings from all AWS services, enhanced with the two most-used Azure services alongside the security-critical service of IAM within Azure, the assessment considers every individual service across multiple vendors in a justified manner. The assessment vendor-service combinations are referred to with A_i in Table 4.8. These already mentioned Azure services then function as the control set for in-space validation (referred to with K_i), while the remaining Azure services and the entire service set of HPE GreenLake define the rest of the validation set elements (referred to with V_i).

Referring to the lack of NoSQL service availability for HPE GreenLake, this can now be neglected, as an out-of-design validation of NoSQL services will nevertheless be possible.

4.1.2 Current Assessment Framework Sampling

Other than with the CC vendor and services sample, the sample of current frameworks is not limited in its sole number. Rather, it is important to deduce which general kinds of usability and security frameworks are suited for the matter of this work, i.e., which frameworks are already tangible to be rewritten for the purpose of usable security assessment. This implies a general applicability of the framework within its scope rather than the need of designing individual experiments and corresponding performance indicators. This limitation is justified as the goal of this work is to provide a generally applicable framework for usable security itself, reducing the necessity of performing evaluations on a case-by-case basis.

Apart from that, again, the degree of establishment of the framework within its corresponding area of application further justifies the consideration of particular current frameworks. These information will be analyzed by means of available literature in the following.

4.1.2.1 Usability Inspection Framework Sampling

Nielsen (1994) summarized a list of general usability inspection methods [38] that has been re-validated by Hollingsed and Novick (2007) [25]. While these sources might appear dated, it appears that there were no further general usability inspection methods created and used in more recent times. Therefore, the corresponding findings are expected to still hold today and will be enriched by further justification.

Nielsen presents the following usability inspection methods that are explained and contextualized for the purpose of this work in the following:

Heuristic evaluation informal method performed by usability specialists to validate usability elements by means of a set of established guidelines [38].

This process is still recognized as being the most effective and most widely used method in terms of usability inspection [25]. It is based on general guidelines that can be used for different kinds of usability inspections, regardless of the underlying interface.

Cognitive walkthrough more detailed procedure simulating a user solving a problem, understanding if user's goals and memory lead to a correct action [38].

This method is less generally applicable compared to heuristic evaluation. It is rather more dependent on the actual task at hand, requiring a specific experiment design and corresponding passing and failing criteria. It can therefore hardly be supported by a framework. Plus, it is less widely used in practice than heuristics [25].

Pluralistic usability walkthrough meetings in which different stakeholders discuss individual dialogue elements of a usability scenario [38].

These walkthroughs present themselves with similar shortcomings as cognitive walkthroughs. In addition, they require the consideration of different users (developers, designers, end users, etc.) [38], which again leads to assume that the design of such experiment is highly individual and dependent on the actual use case.

Formal usability inspection combination of heuristic evaluation and cognitive walkthroughs [38].

Again, similar downsides to those from individual cognitive walkthroughs appear. Apart from being highly user and task-dependent, the experiments can produce biased insights as developers of the interface under test assume the role of a user [25]. It has also been stated that this form of usability inspection has hardly been used after the 1990s [25].

Feature inspection Individual assessment of features that are evaluated regarding standard tasks, potentially long sequences or unintuitive steps [38].

Similarly to what has already been said, feature inspections require concrete task and performance criteria definitions.

Consistency and standards inspection isolated validation of design consistency with regards to predefined standards and compliance [38].

While this discipline is important for sole usability purposes, design consistency and standards play a minor role in terms of usable security.

By exclusion, it can be argued that heuristic evaluation is the most suitable usability inspection method that can be used as a starting foundation to develop a usable security assessment framework from it. As mentioned, it is presented as the most widely used and most effective method, which increasingly justifies the approach to design advanced heuristics for usable security purposes. However, other methods (e.g., formal usability inspections and feature inspections) can also benefit from heuristics. It also needs to be mentioned that empirical case-by-case inspections remain most effective outside the scope of frameworks, guidelines and methods [25]. However, this cannot be considered within the scope of that work as the ultimate goal is to provide a framework that reduces the need of individual experiment creation. This ultimately means, that the unified framework will also be developed as a set of usable security heuristics.

Furthermore, it is necessary to further limit the scope to particular sets of current heuristics. For that matter, the most popular heuristics are deduced from literature. Pierre (2015) conducted a literature analysis against 33 distinct sets of usability inspection sets in order to find similarities and differences among them [44]. In essence, it is observed that all usability heuristics follow the common goal of providing general guidelines for usability design and assessment. It can also be expected that more recent creations of heuristics are more specialized in terms of a limited, but more focussed, area of application.

The choice of particular heuristic sets should therefore be based on the general applicability of these heuristics which is expected to appear in older publications. Based on its popularity in both academia and the industry, the choice falls on

- *The Eight Golden Rules of Interface Design* by Ben Shneiderman (most recently in [51], originally in [50], overall more than 18,000 citations)

- *10 Usability Heuristics for User Interface Design* by Jakob Nielsen (most recently in [37], originally in [39], overall more than 10,000 citations)

Both sets of heuristics were adapted and updated by the original authors over time, which makes them both established and recent. The number of overall citations underlines their popularity and justifies the consideration of those heuristics going forward. It needs to be mentioned that similarities in the heuristics across both sets cannot be excluded. However, this is accepted as vague differences might prove valuable in terms of findings for the creation of the unified framework.

4.1.2.2 Security Design and Assessment Framework Sampling

The corresponding sampling of security design and assessment frameworks follows the same criteria, being general applicability as well as establishment and popularity in the field.

The sampling process starts with the work produced by Saltzer and Schröder (1975) [46]. As mentioned, their Security Design Principles are considered the starting point of usable security research, especially the final principle of *psychological acceptability* (cf. Section 2.3). Initially, it appears reasonable to consider their principles within this sample. However, the datedness of this source needs to be addressed.

For that matter, Smith (2012) aimed at validating these principles [53]. In summary, these security design principles still hold, with the nuance that some of those principles remain more important or critical than other. Furthermore, more recent developments of such principles are strongly influenced by Saltzer's and Schröder's original work which leads to assume that their principles remain relevant [53]. Further literature research yielded no results that contained notable differences in such design principles. As with the heuristics sampling, there exist specialized sets of principles that violate the criterion of general applicability. The popularity is validated by observing over 3,000 citations of their work, which is significantly higher than any other work presenting a security design or assessment framework.

This leads to the consequence that the Security Design Principles by Saltzer and Schröder (1975) represents the only current security framework that will be considered in this work going forward.

4.1.3 Sampling Summary

In conclusion, the following current frameworks will be considered:

Usability I *The Eight Golden Rules of Interface Design* by Ben Shneiderman

Usability II *10 Usability Heuristics for User Interface Design* by Jakob Nielsen

Security *Security Design Principles* by Saltzer & Schröder

These will be applied onto the CC service assessment set from Table 4.8. The explicit approach to that process is described in the next section.

4.2 Assessment Approach

Refine current framework assessment approach.

In order to evaluate the applicability of current usability and security frameworks, the chosen frameworks will be applied on the eight CC assessment services A_i from Table 4.8. Specifically, every CC assessment service-vendor combination will be individually evaluated by means of all three mentioned frameworks.

This yields a total of 24 assessments. Each assessment will be performed manually, noting findings in terms of the usable security applicability for the particular case. It is expected to have the following categories of findings:

- pure usability aspects directly related to the application of a usability framework,
- pure usability aspects derived from applying a security framework,
- pure security aspects directly related to the application of a security framework,
- pure security aspects derived from applying a usability framework,

- usable security aspects derived from a usability framework, and
- usable security aspects derived from a security framework.

These categories then yield three applicability degrees:

1. aspects that are applicable without change,
2. aspects that are applicable with change, and
3. aspects that are not applicable at all.

Furthermore, each assessment is expected to aid the subsequent design process of the unified framework (cf. C2) such that relevant aspects that are missing from the current framework can be derived by justified assumption. After performing each evaluation, the findings and insights can be prioritized in terms of their relevance, how often they appeared, etc.

The results from the control group evaluation will be most thoroughly documented as they will serve as additional input to the validation process of the unified framework (cf. C3).

It is expected that certain cloud services' interfaces do not solely include security aspects, which is why the evaluation will only focus on aspects within the usability that is mostly or entirely security-related.

The ultimate design of this proposed assessment will be revised after deciding on the specific CC offerings, services, as well as current frameworks, if necessary. In any case, the output of the total assessment will be subsequently used in the next contribution C2.

4.3 Design Approach

Refine unified framework design approach.

The findings from C1 are now applied in order to design the unified qualitative usable security assessment framework, as part of the second contribution.

The design will consider the following findings in order:

1. High-priority aspects from current frameworks that can be taken over without change.
2. High-priority aspects from current frameworks that can be taken over with change.
3. Low-priority aspects from current frameworks that can be taken over without change.
4. Low-priority aspects from current frameworks that can be taken over with change.
5. High-priority aspects derived by justified assumption.
6. Low-priority aspects derived by justified assumption.

After each step, it is validated that the included steps are not redundant to the previously undertaken step. Finally, after performing all steps based on all findings from C1, all aspects are considered in total in terms of consolidation or generalization, if applicable.

Again, the ultimate design of this proposed framework development process will be revised after having the actual assessment of the current frameworks. In any case, this contribution is expected to yield a unified framework that is similar to the mode of application to the previously chosen, current frameworks. This framework is now subject to validation in terms of the third and last contribution C3, outlined in its separate Chapter 7.

5 Current Framework Assessment

6 Unified Framework Design

7 Evaluation

The validation of the unified framework is expected to assess its validity and benefits compared to current frameworks as well as overall evaluate the success of this research. It also defines the final contribution C3 of this work. This will be done in two steps; (i) the validation by means of the control set K_{CC} as well as by means of the remaining feature set V_{CC}.

As the control set features have already been assessed by means of the current frameworks, this form of validation can evaluate efficiency, conclusivity, and applicability in comparison to those frameworks. Since these features have been used to design the framework, there is the risk that it is rather tailored to the particular feature set rather than generally applicable (i.e., *overfitting*). For that, the remaining validation set is used. Even though it cannot compare its outcome with current frameworks, it can validate if the unified framework is generally applicable, efficient and conclusive outside of its design space (i.e., specific cloud offerings *and* services. In short:

Comparison of Operation $\mathcal{K}_c := K_{CC} \times (C_{FW_{sec}} \cup C_{FW_{use}}) = \{FW_{sec_1}(B2), \ldots, FW_{use_2}(B3)\}$ compared with $\mathcal{K}_u := K_{CC} \times FW_u = \{FW_u(B2), FW_u(B3)\}$

Validation of General Applicability $\mathcal{V} := (V_{CC} \backslash K_{CC}) \times FW_u = \{FW_u(B4), FW_u(B5), FW_u(C2), FW_u(C3), FW_u(C4), FW_u(C5)\}$,

with $\mathcal{K}_c$ being the control set assessment of current frameworks, $\mathcal{K}_u$ being the control set assessment of the unified framework, and FW_u being the unified framework application.

Again, as with previous contributions, the ultimate design of this proposed evaluation process will be revised after having the actual framework design in place. This is because the nature of a qualitative framework complicates performing a quantifiable evaluation. In any case, this evaluation will conclude the work by answering whether this unified framework actually solves (or, at least, partially addresses) the underlying problems of this thesis (i.e., slow case-by-case approach for usable security research as well as little consideration of non-end-users), critically consider shortcomings of the developed framework, and finally suggest further research in that particular area.

8 Outlook & Conclusion

Bibliography

[1] *2021 State of the Cloud Report.* Tech. rep. Flexera, 2021. URL: https://resources.flexera.com/web/pdf/report-cm-state-of-the-cloud-2021.pdf (visited on 09/24/2022).

[2] *2022 Cloud Security Report.* Tech. rep. ISC2, 2022. (Visited on 09/24/2022).

[3] Yasemin Acar, Sascha Fahl, and Michelle L. Mazurek. "You are Not Your Developer, Either: A Research Agenda for Usable Security and Privacy Research Beyond End Users". In: *2016 IEEE Cybersecurity Development (SecDev).* Nov. 2016, pp. 3–8. DOI: 10.1109/SecDev.2016.013.

[4] Anne Adams and Martina Angela Sasse. "Users are not the enemy". In: *Communications of the ACM* 42.12 (Dec. 1999), pp. 40–46. ISSN: 0001-0782. DOI: 10.1145/322796.322806. (Visited on 09/23/2022).

[5] Sameera Abdulrahman Almulla and Chan Yeob Yeun. "Cloud computing security management". In: *2010 Second International Conference on Engineering System Management and Applications.* Mar. 2010, pp. 1–7.

[6] Stephen Ambore, Huseyin Zekeriya Dogan, and Edward Apeh. "Development of Usable Security Heuristics for Fintech". In: (July 2021). Publisher: BCS Learning & Development. DOI: 10.14236/ewic/HCI2021.12. URL: https://www.scienceopen.com/hosted-document?doi=10.14236/ewic/HCI2021.12 (visited on 09/23/2022).

[7] *AWS Services List: Top 25 Amazon Web Services & Products list.* en-US. Section: Cloud Computing. Sept. 2022. URL: https://intellipaat.com/blog/aws-services-list-and-products/ (visited on 10/17/2022).

[8] Deanna D. Caputo et al. "Barriers to Usable Security? Three Organizational Case Studies". In: *IEEE Security & Privacy* 14.5 (Sept. 2016). Conference Name: IEEE Security & Privacy, pp. 22–32. ISSN: 1558-4046. DOI: 10.1109/MSP.2016.95.

[9] Sonia Chiasson, Robert Biddle, and Anil Somayaji. "Even experts deserve usable security: Design guidelines for security management systems". In: *In SOUPS Workshop on Usable IT Security Management (USM.* 2007.

[10] Sonia Chiasson, Robert Biddle, and Anil Somayaji. "Even experts deserve usable security: Design guidelines for security management systems". In: *SOUPS Workshop on Usable IT Security Management (USM).* 2007. URL: https://cups.cs.cmu.edu/soups/2007/workshop/Design_Guidelines.pdf (visited on 09/26/2022).

[11] *Cost of a Data Breach. Report 2022.* Tech. rep. Armonk, NY: IBM Corporation, July 2022. URL: https://www.ibm.com/security/data-breach (visited on 09/24/2022).

[12] *Cybersecurity Market - Growth, Trends, Covid-19 Impact, and Forecasts (2022 - 2027).* Tech. rep. Mordor Intelligence, 2022. URL: `https://www.mordorintelligence.com/industry-reports/cyber-security-market` (visited on 09/24/2022).

[13] Kelly Dent. *Revealed: the top 43 AWS products and services.* en. Section: Ask the Expert. URL: `https://www.jeffersonfrank.com/insights/aws-products` (visited on 10/17/2022).

[14] *Ergonomics of human-system interaction — Part 110: Interaction principles.* Tech. rep. ISO 9241-110:2020(en). International Organization for Standardization, 2020. URL: `https://www.iso.org/obp/ui/#iso:std:iso:9241:-110:ed-2:v1:en` (visited on 09/25/2022).

[15] *Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems.* Tech. rep. ISO 9241-210:2019(en). International Organization for Standardization, 2019. URL: `https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-2:v1:en` (visited on 09/25/2022).

[16] Sascha Fahl et al. "Confidentiality as a Service – Usable Security for the Cloud". In: *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications.* ISSN: 2324-9013. June 2012, pp. 153–162. DOI: `10.1109/TrustCom.2012.112`.

[17] Dennis Felsch et al. "How Private is Your Private Cloud? Security Analysis of Cloud Control Interfaces". In: *Proceedings of the 2015 ACM Workshop on Cloud Computing Security Workshop.* CCSW '15. New York, NY: Association for Computing Machinery, Oct. 2015, pp. 5–16. ISBN: 978-1-4503-3825-7. DOI: `10.1145/2808425.2808432`. URL: `https://doi.org/10.1145/2808425.2808432` (visited on 09/26/2022).

[18] Denis Feth, Andreas Maier, and Svenja Polst. "A User-Centered Model for Usable Security and Privacy". In: *Human Aspects of Information Security, Privacy and Trust.* Ed. by Theo Tryfonas. Lecture Notes in Computer Science. Cham, CH: Springer International Publishing, 2017, pp. 74–89. ISBN: 978-3-319-58460-7. DOI: `10.1007/978-3-319-58460-7_6`.

[19] Simone Fischer-Hübner. *IT-Security and Privacy.* Lecture Notes in Computer Science 1958. Berlin, DE. Heidelberg, DE. New York, NY.: Springer, 2001. ISBN: 3-540-42142-4.

[20] *Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1.* Tech. rep. NIST CSWP 04162018. Gaithersburg, MD: National Institute of Standards and Technology, Apr. 2018. DOI: `10.6028/NIST.CSWP.04162018`. URL: `http://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf` (visited on 09/25/2022).

[21] Simson Garfinkel and Heather Richter Lipford. *Usable Security: History, Themes, and Challenges.* Cham, CH: Springer International Publishing, 2014. ISBN: 978-3-031-01215-0 978-3-031-02343-9. DOI: `10.1007/978-3-031-02343-9`. URL: `https://link.springer.com/10.1007/978-3-031-02343-9` (visited on 09/24/2022).

[22] Matthew Green and Matthew Smith. "Developers are Not the Enemy!: The Need for Usable Security APIs". In: *IEEE Security & Privacy* 14.5 (Sept. 2016). Conference Name: IEEE Security & Privacy, pp. 40–46. ISSN: 1558-4046. DOI: `10.1109/MSP.2016.111`.

[23] Keiko Hashizume et al. “An analysis of security issues for cloud computing”. In: *Journal of Internet Services and Applications* 4.1 (Feb. 2013), p. 5. ISSN: 1869-0238. DOI: 10.1186/1869-0238-4-5. URL: https://doi.org/10.1186/1869-0238-4-5 (visited on 09/26/2022).

[24] Yasser M. Hausawi and William H. Allen. “Usable-Security Evaluation”. In: *Human Aspects of Information Security, Privacy, and Trust*. Ed. by Theo Tryfonas and Ioannis Askoxylakis. Vol. 9190. Series Title: Lecture Notes in Computer Science. Cham, CH: Springer International Publishing, 2015, pp. 335–346. ISBN: 978-3-319-20375-1 978-3-319-20376-8. DOI: 10.1007/978-3-319-20376-8_30. URL: http://link.springer.com/10.1007/978-3-319-20376-8_30 (visited on 09/25/2022).

[25] Tasha Hollingsed and David G. Novick. “Usability inspection methods after 15 years of research and practice”. In: *Proceedings of the 25th annual ACM international conference on Design of communication*. SIGDOC ’07. New York, NY, USA: Association for Computing Machinery, Oct. 2007, pp. 249–255. ISBN: 978-1-59593-588-5. DOI: 10.1145/1297144.1297200. URL: https://doi.org/10.1145/1297144.1297200 (visited on 10/04/2022).

[26] Travor House. *Top 25 AWS Services List 2021*. en-US. July 2021. URL: https://allcode.com/top-aws-services/ (visited on 10/17/2022).

[27] *Huge Cloud Market Still Growing at 34% Per Year; Amazon, Microsoft & Google Now Account for 65% of the Total | Synergy Research Group*. Tech. rep. Reno, NV: Synergy Research Group, Apr. 2022. URL: https://www.srgresearch.com/articles/huge-cloud-market-is-still-growing-at-34-per-year-amazon-microsoft-and-google-now-account-for-65-of-all-cloud-revenues (visited on 10/17/2022).

[28] Rajesh Kumar. *List of Top Microsoft Azure Services*. en-US. Aug. 2021. URL: https://www.devopsschool.com/blog/list-of-top-microsoft-azure-services/ (visited on 10/17/2022).

[29] Markus Lennartsson, Joakim Kävrestad, and Marcus Nohlberg. “Exploring the Meaning of “Usable Security””. In: *Human Aspects of Information Security and Assurance*. Ed. by Nathan Clarke and Steven Furnell. IFIP Advances in Information and Communication Technology. Cham, CH: Springer International Publishing, 2020, pp. 247–258. ISBN: 978-3-030-57404-8. DOI: 10.1007/978-3-030-57404-8_19.

[30] *Malware increased by 358% in 2020*. Help Net Security. Feb. 2021. URL: https://www.helpnetsecurity.com/2021/02/17/malware-2020/ (visited on 09/24/2022).

[31] Sheeja Shaji Manakattu, Shivakumar Murugesh, and Rajashekhar Ningappa Hirekurabar. “Security Landscape for Private Cloud”. In: *Inventive Computation Technologies*. Ed. by S. Smys, Robert Bestak, and Álvaro Rocha. Lecture Notes in Networks and Systems. Cham, CH: Springer International Publishing, 2020, pp. 67–78. ISBN: 978-3-030-33846-6. DOI: 10.1007/978-3-030-33846-6_8.

[32] Paul Mee and Rico Brandenburg. *After reading, writing and arithmetic, the 4th ’r’ of literacy is cyber-risk*. World Economic Forum. Dec. 2020. URL: https://www.weforum.org/agenda/2020/12/cyber-risk-cyber-security-education/ (visited on 09/24/2022).

[33] Peter Mell and Timothy Grance. *The NIST Definition of Cloud Computing*. Tech. rep. NIST SP 800-145. Gaithersburg, MD: National Institute of Standards and Technology, Sept. 2011, p. 7.

[34] Madison Miner. *The Top 10 Most Used AWS Services*. en. Aug. 2020. URL: `https://insider.ssi-net.com/insights/the-top-10-most-used-aws-services` (visited on 10/17/2022).

[35] Kate Moran. *Usability Testing 101*. Nielsen Norman Group. Dec. 2019. URL: `https://www.nngroup.com/articles/usability-testing-101/` (visited on 09/24/2022).

[36] Michael Nieles, Kelley Dempsey, and Victoria Yan Pillitteri. *An introduction to information security*. Tech. rep. NIST SP 800-12r1. Gaithersburg, MD: National Institute of Standards and Technology, June 2017. DOI: `10.6028/NIST.SP.800-12r1`. URL: `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-12r1.pdf` (visited on 09/25/2022).

[37] Jakob Nielsen. *10 Usability Heuristics for User Interface Design*. en. Nov. 2020. URL: `https://www.nngroup.com/articles/ten-usability-heuristics/` (visited on 09/24/2022).

[38] Jakob Nielsen. "Usability inspection methods". In: *Conference Companion on Human Factors in Computing Systems*. CHI '94. New York, NY, USA: Association for Computing Machinery, Apr. 1994, pp. 413–414. ISBN: 978-0-89791-651-6. DOI: `10.1145/259963.260531`. URL: `https://doi.org/10.1145/259963.260531` (visited on 10/04/2022).

[39] Jakob Nielsen and Rolf Molich. "Heuristic evaluation of user interfaces". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '90. New York, NY, USA: Association for Computing Machinery, Mar. 1990, pp. 249–256. ISBN: 978-0-201-50932-8. DOI: `10.1145/97243.97281`. URL: `https://doi.org/10.1145/97243.97281` (visited on 10/04/2022).

[40] Donald A. Norman. *The design of everyday things*. 3rd ed. Philadelphia, PA: Basic Books, 2013. ISBN: 978-0-465-05065-9.

[41] Jatin Panchal. *List of Top 10 Azure App Services*. en. Aug. 2020. URL: `https://www.rlogical.com/blog/list-of-top-10-most-popular-azure-services/` (visited on 10/17/2022).

[42] Simon Parkin et al. "A stealth approach to usable security: helping IT security managers to identify workable security solutions". In: *Proceedings of the 2010 New Security Paradigms Workshop*. NSPW '10. New York, NY: Association for Computing Machinery, Sept. 2010, pp. 33–50. ISBN: 978-1-4503-0415-3. DOI: `10.1145/1900546.1900553`. URL: `https://doi.org/10.1145/1900546.1900553` (visited on 09/23/2022).

[43] Napoleon C. Paxton. "Cloud Security: A Review of Current Issues and Proposed Solutions". In: *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*. Nov. 2016, pp. 452–455. DOI: `10.1109/CIC.2016.066`.

[44] Raisa da Silva de Queiroz Pierre. “Heuristics in Design: A Literature Review”. en. In: *Procedia Manufacturing*. 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015 3 (Jan. 2015), pp. 6571–6578. ISSN: 2351-9789. DOI: 10.1016/j.promfg.2015.07.961. URL: https://www.sciencedirect.com/science/article/pii/S2351978915009622 (visited on 10/04/2022).

[45] Christian Reuter, Luigi Lo Iacono, and Alexander Benlian. “A quarter century of usable security and privacy research: transparency, tailorability, and the road ahead”. In: *Behaviour & Information Technology* 41.10 (July 2022), pp. 2035–2048. ISSN: 0144-929X, 1362-3001. DOI: 10.1080/0144929X.2022.2080908. URL: https://www.tandfonline.com/doi/full/10.1080/0144929X.2022.2080908 (visited on 09/25/2022).

[46] J.H. Saltzer and M.D. Schroeder. “The protection of information in computer systems”. In: *Proceedings of the IEEE* 63.9 (Sept. 1975). Conference Name: Proceedings of the IEEE, pp. 1278–1308. ISSN: 1558-2256. DOI: 10.1109/PROC.1975.9939.

[47] Girdharee Saran. *List of Top 10 Azure Services*. en-US. Section: Microsoft Azure. Sept. 2021. URL: https://www.whizlabs.com/blog/top-azure-services/ (visited on 10/17/2022).

[48] Holger Schulze. *Cloud Security Report 2022*. Tech. rep. Cybersecurity Insiders, 2022. URL: https://www.checkpoint.com/cyber-hub/cloud-security/what-is-cloud-security/the-biggest-cloud-security-challenges-in-2022/ (visited on 09/24/2022).

[49] Naresh Kumar Sehgal and Pramod Chandra P. Bhatt. *Cloud Computing: Concepts and Practices*. Cham, CH: Springer International Publishing, 2018. ISBN: 978-3-319-77838-9 978-3-319-77839-6. DOI: 10.1007/978-3-319-77839-6. URL: https://link.springer.com/10.1007/978-3-319-77839-6 (visited on 09/25/2022).

[50] Ben Shneiderman. *Designing The User Interface: Strategies for Effective Human-Computer Interaction*. en. 1st ed. Pearson Education, 1987. ISBN: 978-81-317-2163-6.

[51] Ben Shneiderman. *The Eight Golden Rules of Interface Design*. URL: https://www.cs.umd.edu/users/ben/goldenrules.html (visited on 10/18/2022).

[52] Ashish Singh and Kakali Chatterjee. “Cloud security issues and challenges: A survey”. In: *Journal of Network and Computer Applications* 79 (Feb. 2017), pp. 88–115. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2016.11.027. URL: https://www.sciencedirect.com/science/article/pii/S1084804516302983 (visited on 09/26/2022).

[53] Richard E. Smith. “A Contemporary Look at Saltzer and Schroeder’s 1975 Design Principles”. In: *IEEE Security & Privacy* 10.6 (Nov. 2012). Conference Name: IEEE Security & Privacy, pp. 20–25. ISSN: 1558-4046. DOI: 10.1109/MSP.2012.85.

[54] *The Top 15 Most Commonly Used AWS Services You Should Know About in 2022*. Oct. 2022. URL: https://www.boxpiper.com/posts/the-top-15-most-commonly-used-aws-services-you-should-know-about (visited on 10/17/2022).

[55] Mary Theofanos. “Is Usable Security an Oxymoron?” In: *Computer* 53.2 (Feb. 2020), pp. 71–74. ISSN: 1558-0814. DOI: 10.1109/MC.2019.2954075. (Visited on 09/25/2022).

[56] *Top 10 AWS services list.* en-US. 2022. URL: https://www.clickittech.com/aws/aws-services-list/ (visited on 10/17/2022).

[57] *Top 10 Azure Services.* en. Section: Uncategorized. Mar. 2020. URL: https://www.ecocloudservices.com/top-10-azure-services/ (visited on 10/17/2022).

[58] *Top 10 Azure Services and Products in 2022.* en-US. Section: Cloud Computing. Sept. 2022. URL: https://intellipaat.com/blog/top-azure-services/ (visited on 10/17/2022).

[59] *Top 10 Microsoft Azure Services in 2021.* en-US. Oct. 2021. URL: https://www.testpreptraining.com/blog/top-10-microsoft-azure-services-in-2021/ (visited on 10/17/2022).

[60] *Top 30 AWS Services List to Choose in 2022.* en. Section: Looker. Apr. 2021. URL: https://mindmajix.com/top-aws-services (visited on 10/17/2022).

[61] Alma Whitten and J. D. Tygar. “Why Johnny Can’t Encrypt: A Usability Evaluation of {PGP} 5.0”. en. In: 1999. URL: https://www.usenix.org/conference/8th-usenix-security-symposium/why-johnny-cant-encrypt-usability-evaluation-pgp-50 (visited on 09/23/2022).

[62] Zhang Yandong and Zhang Yongsheng. “Cloud computing and cloud security challenges”. In: *2012 International Symposium on Information Technologies in Medicine and Education.* Vol. 2. Aug. 2012, pp. 1084–1088. DOI: 10.1109/ITiME.2012.6291488.

[63] Mary Ellen Zurko and Richard T. Simon. “User-centered security”. In: *1996 workshop on New security paradigms - NSPW ’96.* Lake Arrowhead, CA: ACM Press, 1996, pp. 27–33. ISBN: 978-0-89791-944-9. DOI: 10.1145/304851.304859. URL: http://portal.acm.org/citation.cfm?doid=304851.304859 (visited on 09/23/2022).