TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Daniel Golubkov

Vulnerability Assessment of Modern Authentication and Authorization Protocols

Literature Review Report

Supervisor: Edmund Laugasson,
MSc

Tallinn 2024

Abstract

Authentication and authorization have become necessary access control concepts of most systems. The increase of new cyberthreat dangers creates a need to analyse how secure are the existing authentication and authorization protocols, which types of vulnerability assessments were conducted on these protocols.

Throughout this literature review, various vulnerabilities of authentication and authorization protocols, their mitigation strategies, vulnerability assessment and penetration testing tools, techniques were identified. Newly proposed process improvements for existing frameworks and in-development protocols were also reviewed. Research gaps were found, which include a lack of vulnerability assessment of in-development protocols. Additionally, missing long-term testing for proposed novel solutions, absent qualification analysis of the VAPT process, insufficient hands-on evaluation of vulnerability assessment tools was also identified during the literature review.

Further research on the vulnerability assessment of in-development authentication and authorization protocols can aid in finding possible vulnerabilities of these protocols and identify their safety for usage by service providers.

List of abbreviations and terms

AD	Active Directory
CA	Certificate Authority
CSRF	Cross Site Request Forgery
DDoS	Distributed Denial of Service
DQN	Deep Q-Network
DoS	Denial of Service
GNAP	Grant Negotiation and Authorization Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IAM	Identity and Access Management
IP	Internet Protocol
IT	Information Technology
IdP	Identity Provider
JSON	JavaScript Object Notation
JWT	JSON Web Token
LDAP	Lightweight Directory Access Protocol
MDQ	Metadata Query Protocol
MITM	Man In The Middle
NVD	National Vulnerability Database
OAuth	Open Authorization
OIDC	OpenID Connect
PKCE	Proof of Key Code Exchange
REST API	RESTful Application Programming Interface
SAML	Security Assertion Markup Language
SDK	Software Development Kit
SP	Service Provider
SSI	Self-Sovereign Identity
SSO	Single Sign-On
UMA	User-Managed Access
URL	Uniform Resource Locator
VAPT	Vulnerability Assessment and Penetration Testing
VAS	Vulnerability Assessment Scanner
WAF	Web Application Firewall
XXE	Extensible Markup Language External Entity
XML	Extensible Markup Language
XSS	Cross Site Scripting

XSW	Extensible Markup Language Signature Wrapping
ZAP	Zed Attack Proxy

Table of contents

List of tables

1 Introduction

This chapter gives way to the introduction of the research problem, research goal definition, construction of research questions, scope and the description of the research methods used.

1.1 Research Problem Statement

Authentication and authorization have become core identification and access control concepts of most information technology systems with existing user management processes. However, with new cyberthreat dangers increasing every year a question arises on which modern authentication and authorization protocols are currently implemented into the access control processes of various service providers. Another inquiry is to examine whether there are any in-development authentication and authorization protocols, which could potentially further increase access control security.

It becomes a necessity to conduct vulnerability assessments of these protocols to evaluate possible attack vectors and identify unsecure implementation issues. Remediation of identified security issues would in turn enhance security of service providers, protecting from cyberattacks and preventing reputational, monetary and structural damages.

1.2 Research Goal

The main goal of this literature review is to identify and review published studies and literature related to modern authentication and authorization protocols, their possible security threats, implementation issues. Additionally, it is necessary to discover possible ways of conducting vulnerability assessments of these protocols through various tools and frameworks.

1.3 Research Questions

To make proper analysis of the current state of the art, a concrete research question and sub questions were stated to aid in conducting a more detailed level of the research process:

- **[RQ1]** What is the current state of the art in conducting vulnerability assessment of modern authentication and authorization protocols?
- **[RQ1.1]** What are the current authentication and authorization protocols?
- **[RQ1.2]** What are the known vulnerabilities related to authentication and authorization protocols and what are the possible mitigations of these vulnerabilities?
- **[RQ1.3]** What are the possible methods and tools for conducting vulnerability assessment and penetration testing of authentication and authorization protocols?

1.4 Research Scope

The research will focus on the analysis of literature related to authentication and authorization protocols. More specifically, this research is aimed to identifying the possible vulnerabilities and ways to conduct vulnerability assessment of these protocols. If no known vulnerabilities are identified, then identification of overall possibilities in conducting vulnerability assessment of these protocols is provided.

At its current state, this research is only limited to a literature review, hence it is purely theoretical - the limitations are:

1. There will be no actual implementation of the researched protocols in a testing environment.
2. There will be no practical vulnerability assessment of these protocols from individual perspective.

The number of articles or research papers considered for review during this research is limited to twenty (scope of literature review) – corresponding analysis and conclusions will be based on the results of the literature review.

1.5 Research Methodology

For this literature review, a combination of both the analytical and qualitative research methods is used, specifically:

• Analytical research method – analysis of current authentication and authorization protocols: evaluation and comparison, research of known vulnerabilities. Analysis of possible ways to conduct vulnerability assessment of these protocols.

• Qualitative research method – research of types, security enhancements, short-comings of various authentication and authorization protocols. Research of types of vulnerability assessments.

2 Literature Review Structure

This chapter describes the actions and processes executed for finding relevant literature through various sources, the search, selection and data extraction strategies.

2.1 Literature Sources

The following recommended literature databases were used for searching for relevant documents:

- SCOPUS
- ACM DL (Association for Computing Machinery Digital Library)
- IEEE Xplore (Institute of Electrical and Electronics Engineers digital library)
- ScienceDirect

These databases were accessed through the proxy of Tallinn Technical University for an increased access level.

2.2 Search Strategy

For searching of relevant literature, a systematic search was conducted using the following search query string through the stated literature databases:

(("Authentication protocols" OR "Authorization protocols") AND ("Vulnerability assessment" OR "Vulnerability analysis" OR "Security assessment" OR "Security analysis" OR "Security evaluation" OR "Penetration testing")) AND attendance 2017…2025

Such query states that keywords like “Authentication protocols”, “Security assessment”, etc. should exist within the research documents. This is needed to narrow down the search scope for better relevance to the research goal.

The oldest publication year of a research document is set to 2017, this is set to give way for reviewing of new research on the topic while also leaving the possibility for older, more refined research to create additional interest.

Table 1. shows that, depending on the digital library/database, results of the query and paper selection for review were various.

Table 1. Document-related decisions based on the digital database.

Database	**Number of found related documents**	**Initial inclusion**	**Number of documents excluded due to exclusion criteria and manual evaluation**	**Number of documents chosen for the literature review**
SCOPUS	929	23	23	0
ACM DL	207	18	16	2
IEEE Xplore	291	35	19	16
ScienceDirect	1265	15	13	2

2.3 Selection Strategy

A specific selection strategy consisting of inclusion and exclusion criterions was developed for aiding in choosing the research documents. Overall, it is necessary to consider only relevant research documents while avoiding non-recent and inaccessible documents. Additionally, manual selection was conducted for individual research documents based on the title, abstract and conclusion relevance to the research goal.

2.3.1 Inclusion Criteria

Overall, the inclusion criteria for the research document selection involved:

- Documents matching the search query.
- Documents with matching topic, relevant to the research goal.
- Research journals, articles, volumes.

2.3.2 Exclusion Criteria

Overall, the exclusion criteria for the research document selection involved:

- Documents not written in English.
- Documents not closely related to the topic (abstract, title, conclusion) and/or not relevant to the research goal.
- Documents with closed access.

2.4 Selected Literature

Number of selected research documents for the literature review is twenty, as stated before. The documents were chosen by carefully evaluating the selection strategy and their relevance to the research goal. Chapter **3.4 Related Work** will give further insight into the selected literature and discuss potential shortcomings and developments.

2.5 Data collection strategy

For the scope of this literature review the data collection from the research documents was manual. Each selected research document was read through and analysed using the stated research methodology. Relevant data and highlights of each document was collected and added to this literature review report.

3 Background

This chapter describes the brief background of researched topic and the related work in assessing vulnerabilities of authentication and authorization protocols.

3.1 Authentication

Authentication governs the verification of user identities in various IT systems and applications. Typical authentication through username and password has become prone to brute force attacks, and because of this, implementation of various authentication protocols is becoming a new standard in the industry.

3.2 Authorization

Authorization is often linked with authentication, but as a separate process it handles access to various resources for authenticated users based on their attributes. Privileged users like administrators usually have full access to manage a system, while standard users have no such access. Authorization can be handled by separate mechanisms like attribute-based access control procedures, or by protocols. Sometimes, a protocol can support both user authentication and authorization.

3.3 Vulnerability Assessment

Vulnerabilities are the direct or indirect security flaws of a system. Vulnerabilities consequently create security risks, which when exploited, can cause damage to the system and monetary, reputational, structural damages to the service provider hosting that system. New vulnerabilities are found every day, organizations like OWASP provide insights and updates on discoveries related to vulnerabilities.

Vulnerability assessment is a process of gathering information about a system, scanning that system for vulnerabilities, identification of those vulnerabilities, and consequent reporting to corresponding parties.

3.4 Related Work

Pöhn et al., "New Directions and Challenges within Identity and Access Management" [1] describes current and future authentication and authorization protocols. The research paper states that new authentication protocols are in development, which could potentially increase the security of IAM processes in enterprises and SPs.

Current authentication and authorization protocols:

1. SAML 2.0 – protocol, which supports both authentication and authorization processes through metadata documents in XML format.

2. OAuth 2.0 – protocol used mainly for web authorization, does not natively support authentication. Is simpler in integrating between SPs when compared to SAML 2.0.

3. OIDC – extends the OAuth 2.0 protocol by adding authentication. Uses cryptographic digital signature JWT documents as metadata instead of XML when compared to SAML 2.0, which allows for higher level of security.

Future authentication and authorization protocols are described in Table 2.

Table 2. Future authentication and authorization protocols.

Protocol	**Builds on top of**	**Description**
OAuth 2.1	OAuth 2.0	Introduces additional security methodologies like the omitting of resource owner credentials grant.
GNAP	OAuth 2.0	Introduces additional security mechanisms like parameter encoding with JSON, for increased data confidentiality requirements.
MDQ	SAML 2.0	Implements a simple query language for protocol management, has

		reduced runtime load, implements REST API for metadata management.
UMA SSI	OAuth 2.0, UCIM	Allows for interaction between the UMA and SSI technologies for decentralized access management.

The research paper describes the current and future authentication and authorization protocols in a quite informative and intuitive way. However, it was found that it doesn't specifically state any concrete possibilities of increased attack surface regarding the new protocols, no vulnerability assessment is conducted either. Future authentication and authorization protocols are still in development, which means that they are not yet actively used in enterprise/SP environments.

Sadqi et al., "Web OAuth-based SSO Systems Security" [2] conducts a security analysis of the OAuth 2.0 authorization protocol. It is explained how the protocol works, security properties of the protocol are analysed, possible security threats are provided. Properties of the protocol include confidentiality, authentication and session integrity.

1. Confidentiality – this property relies on secure data transmission during the authorization grant flow.

2. Authentication – this property relies on genuine identity of the authenticated user.

3. Session integrity – this property relies on the creation of a session only for a specific user, who explicitly gave their consent for access to a specific resource.

It is important to note that if all these properties are fulfilled through correct implementation on the SP's side, then the OAuth 2.0 protocol can provide usage flexibility and security.

Possible security threats are divided based on the affected area into two groups: web-based threats and network-based threats. Examples of the threats include session impersonation, user IdP account hijacking, unsafe token generation, etc. The security of the protocol depends heavily on how exactly it was implemented. As an example, providing an authorization service with OAuth 2.0 through an insecure connection

(HTTP) fully voids the confidentiality property of the protocol. As in such case the potential attacker can retrieve access and refresh tokens through a network-based attack.

Overall, the paper gives a sufficient overview of OAuth 2.0 and its possible security threats. Possible research gap relevant to this paper is that the research is limited only to a single protocol. Additionally, no practical examples of any of the stated attacks are provided.

Naik et al., "Securing digital identities in the cloud by selecting an apposite Federated Identity Management from SAML, OAuth and OpenID Connect" [3] when compared to [1] and [2] this research paper provides a more throughout and technical description of the SAML 2.0, OAuth 2.0 and OIDC protocols. Similarly to [2] it is stated that the protocols can be considered as secure when implemented properly. A vulnerability assessment is also conducted for the protocols, and possible security vulnerabilities in the example of DoS attacks, MITM and XSS attacks are provided. In the example of a DoS attack on SAML – instead of targeting the protocol itself, the attacker exploits the vulnerabilities of an IdP system, possibly breaking the SAML authentication flow. Another example - DoS attack targets an OAuth authorization server to make it unavailable for processing the OAuth authorization grant flow. XSS attacks are more relevant to social engineering, where an attacker can mimic a legitimate SP website to trigger an unwanted authentication or authorization data leak. More technical are the MITM attacks – OAuth can possibly be exploited through a maliciously registered OAuth SP by sending user data to a malicious endpoint instead of a legitimate one.

This paper is more descriptive than the others in technical aspects, but it doesn't explicitly provide a hands-on example of execution of one, or many of the stated security vulnerabilities.

Indu et al., "Identity and access management in cloud environment: Mechanisms and challenges" [4] describes current trends in cloud IAM technologies, including digital security mechanisms and access control mechanisms. Different security aspects, vulnerabilities and possible mitigations are provided. What is relevant to the topic is that the paper states possible ways of attacking SAML 2.0, OIDC and OAuth 2.0 protocols – including conducting CSRF, replay attacks, XSS attacks. For mitigating the stated threats,

it is recommended to use the "encrypted communication" functionality of OIDC atop of OAuth 2.0 and to sign plus encrypt SAML 2.0 protocol communication.

The possible threats for the stated protocols are provided in a more extensive manner when compared to [3], but again, practical/hands-on examples of the stated attacks are not provided.

Maidine et al., "Cloud Identity Management Mechanisms and Issues" [5] also describes OAuth 2.0, SAML 2.0 and OIDC – similarly to [2][3] it is stated that the security of the protocols depends on how exactly they were implemented. Additionally, security issues and possible attacks for each protocol are simulated in a very descriptive way through diagrams and pictures, possible vulnerability mitigations are also described in the study.

Attacks against the stated authentication and authorization protocols are stated in Table 3.

Table 3. Various attacks against authentication and authorization protocols.

Attack	**Affected protocol(s)**	**Mitigation**
DoS	OAuth 2.0, SAML 2.0, OIDC	IP address filtering, request limitation, WAF implementation
CSRF	OAuth 2.0, SAML 2.0, OIDC	Usage of state property with user-linked information
307 redirect attack	OAuth 2.0, OIDC	Usage of HTTP 303 instead of HTTP 307 for user redirection
IdP mix-up	OAuth 2.0, OIDC	Usage of HTTPS, IdP whitelisting

XSS	OAuth 2.0, OIDC	Input validation, sanitization, usage of the content security policy
XSW	SAML 2.0	Either element validity checks or structural fixation
XXE	SAML 2.0	Disable external entities

This research paper gives quite a well-defined description of the protocols, their related issues, security threats and possible mitigations of those threats. Hands-on approach to the any of the stated attacks is not present.

Sharma et al., "Security Analysis of OAuth 2.0 Implementation" [6] takes more consideration of the structure of OAuth 2.0 URLs while also describing the general protocol flow and possible security issues related to it. Similarly to [2] [3] and [5] the importance of securing query parameters like "client_id", "state", "scope" and using HTTPS (overall proper implementation) are highlighted for enhancing security of the protocol.

The following protocol flow vulnerabilities are described which can be present due to improper protocol implementation:

1. Authorization code leakage – can happen when the attacker is able to skip the authorization step and directly retrieve an access token during code exchange process. Can be mitigated by implementing PKCE for the authorization code flow.

2. Unauthorized code injection – can happen by executing malicious code on a compromised website and retrieve user data and/or access token if they are stored in the browser (implementation issue).

3. Linking misconfiguration – can happen when there is no cross-verification between the IdP and SP and the user identities are not correctly linked (implementation issue).

The paper discusses a browser extension that was developed by the authors. This extension notifies the end-user if the website they are visiting has OAuth 2.0 protocol incorrectly implemented and could possibly have one (or many) of the stated security vulnerabilities. Out of the analysed 75 websites, 30% were vulnerable to CSRF attacks, 13% had incorrectly implemented the authorization grant flow, which reveals "id_token" or "access_token" parameters in the OAuth URL. Additionally, 2% of websites had secret parameters publicly accessible through analysing the URL during the OAuth authorization grant flow.

This paper described more technical implementation issues of the protocol, which can happen due to improper configuration on SP's side. The description of the attacks was done through a hands-on approach, which is commendable when comparing to previous papers.

Sharif et al., "Best current practices for OAuth/OIDC Native Apps: A study of their adoption in popular providers and top-ranked Android clients" [7] – this study describes the best modern practices of implementing OAuth 2.0 protocol for minimizing the possibility of attacks and security vulnerabilities. It also analyses how well the most popular IdPs and Android applications follow these practices. The authors also introduce an approach to correctly implementing OAuth 2.0 protocol based on the previously stated best practices. This is done by using a plugin built on top of an SDK for verifying application source code correctness.

The static analysis of the source code of specific applications supporting OAuth 2.0 is conducted by using open-source and commercial source-code analysis tools, like Super and MobSF. Additional testing services like OAuth Fuzzer were previously developed to also test SSO implementations of various applications.

For the OAuth 2.0 protocol, vulnerabilities are identified by finding flaws or security issues in the source code of either an Android application or in the source code of a web service. Similar points were previously stated in [2] [3] [5]. Analysis of other protocols is not present.

Thapa et al., "Security Analysis of User Authentication and Methods" [8] describes some of the possible attack methods against some of the stated authentication and authorization protocols, including OAuth 2.0, LDAP, SAML 2.0.

Similarly to [4] and [5] the authors describe the vulnerability of OAuth 2.0 protocol to CSRF and redirect attacks.

Compared to other research papers, this article additionally describes the possible vulnerabilities of LDAP authentication. LDAP allows for user authentication and data storage on the LDAP directory – the data can range from user information to authorization information about various IT-related services and infrastructure. Vulnerabilities described in the article related to LDAP are the following:

1. DDoS – an attack where the LDAP server is overflown with maliciously distributed authentication requests, which therefore results in the server being unable to process legitimate authentication requests, locking users out. Can be mitigated by filtering IP addresses and configuring a firewall.

2. LDAP injection attack – an attack which occurs when malicious code or string manipulation is added to otherwise legitimate queries for the LDAP server, resulting in the server possibly leaking otherwise inaccessible information. Can be mitigated using input sanitization.

Additional SAML 2.0 attack vectors are described, including:

1. SAML message timestamp manipulation – occurs when due to improper implementation SAML authentication messages never expire, when retrieved, such messages can be used to authenticate a threat actor.

2. Invalid signature attack – occurs when due to improper implementation the signature on SAML requests is not checked for CA validity, which can lead to threat actor authentication.

The possible gap of this paper is that the attacks are not described in a hands-on manner, some attacks are difficult to understand due to how the sentences are structured.

Motero et al., "On Attacking Kerberos Authentication Protocol in Windows Active Directory Services: A Practical Survey" [9] conducts a hands-on vulnerability assessment of the Kerberos authentication protocol, which is usually used for on-prem ADs.

The use of the ‘Mimikatz’, ‘Rubeus’ and ‘Hashcat’ tools are highlighted in all the attacks and can be considered as quite useful when there are other system vulnerabilities present.

Some attacks on hard-to-mitigate (due to limitations of the protocol design) vulnerabilities were identified, including:

1. ‘Overpass-the-hash’ – possible when a local machine (and user) on the domain is compromised, the threat actor can conduct injection attacks without knowing the username and password of the affected user.

2. ‘Pass-the-ticket’ – possible when the threat actor can steal identification data from the memory of a compromised machine in the domain, allowing them to impersonate a legitimate user.

3. ‘Silver ticket’ – possible when the threat actor can steal a service account’s password hash by being present in the compromised domain – this hash can then be brute forced and used to attack servers or client machines.

4. ‘Kerberoasting’ - possible when the threat actor has access to machine in the domain, when querying the domain controller, it is possible to capture a hash of a user’s password from the controller’s memory, which can later be brute forced.

‘Future work’ chapter of this article states that a similar assessment can be conducted on a Kerberos instance implemented on a Linux operating system. Because the performed work was limited only to the Windows operating system. Additionally, other scenarios of delegation attacks can be analysed, which would not be limited only to printer servers.

It can also be concluded that the protocol is vulnerable only if a machine or a user on the relevant domain is already compromised.

Chaturvedi et al., “A Comprehensive Vulnerability Tools Analysis for Security and Control in IT Environment and Organizations” [10] describes various vulnerability identification tools. These tools are not specifically made for assessing vulnerabilities of authentication and/or authorization protocols. Though it was revealed previously in [3] [4] [5] [6] [7] that often security vulnerabilities of authentication and authorization protocols are introduced through incorrect implementation (code logic, not using HTTPS, etc.). Because of that, it can be assumed that these tools could be beneficial in revealing

possible security vulnerabilities in the implementation of the protocols by analysing relevant systems.

Various analysed vulnerability assessment/analysis tools include:

1. N-Map – network scanning tool, can potentially be used for revealing possible misconfigurations on the network level of the IdP/SP infrastructure.

2. Wireshark – tool for network protocol analysis, can potentially help in revealing malicious data transmissions within a network.

3. Metasploit framework – a framework for exploiting possible vulnerabilities on various operating systems and machines.

4. Open VAS – a vulnerability scanner capable of multi-machine scanning and management.

This paper is beneficial for the research of the stated topic. It gives insight on the possible software which can be used for conducting vulnerability assessments, though not strictly to authentication and/or authorization protocols.

Shebli et al., "A study on penetration testing process and tools" [11] describes the penetration testing processes and its standards, methods and their importance. It is beneficial to note that a vulnerability is mitigated only if it is patched/fixed when discovered during a penetration test.

A penetration test process consists of three steps, including:

1. Information gathering – scanning of all possible accessible assets for vulnerabilities.

2. Vulnerability analysis – analysis and information gathering about the revealed vulnerabilities.

3. Vulnerability exploit – either manual or automated exploitation of the revealed vulnerabilities.

In Addition to the tools described previously in [9] this paper states three more additional tools:

1. Beef – framework, which allows for conducting browser exploitation techniques, can be used for revealing XSS or CSRF vulnerabilities of a target website.

2. Nessus – vulnerability scanner, used for penetration testing various services running on the target machine.

3. Cain and Abel – framework mostly used for password cracking.

This paper is relevant to the topic because it gives insight on the methodology and processes, which can be used for conducting penetration testing. Though, the paper is not directly related to assessing vulnerabilities of authentication and authorization protocols.

Pareek et al., "Performance Analysis of Vulnerability Detection Tools and Techniques" [12] discusses various penetration testing strategies, including:

1. White box penetration testing – all information about the relevant system is disclosed to the tester, additionally the tester has full access to the system and network.

2. Black box penetration testing – no information about the target system is disclosed to the tester, simulates an 'unauthenticated' type of testing. Tester must conduct full reconnaissance of the target system themselves.

3. Gray box penetration testing – only some information about the target system is disclosed to the tester, requires more time for reconnaissance.

There are also present different types of penetration testing, like network-based penetration testing, web application penetration testing, wireless penetration testing and social engineering penetration testing.

Various tools not stated before are discussed in the article:

1. Burp Suite – toolkit for conducting assessment of vulnerabilities of websites. Highlighted by the author as the most effective tool when compared to the other stated tools.

2. SQLmap – tools used for penetration testing database servers.

3. DirBuster – tool used to brute force directory and file names on web servers.

4. cURL – command-line tool used for making web requests.

Only theoretical aspects of penetration testing are discussed, no real penetration testing was conducted.

Yurtseven et al., "A Review of Penetration Testing and Vulnerability Assessment in Cloud Environment" [13] discusses possible tools for conducting vulnerability assessments in the changeable and always developing cloud infrastructure. The research paper describes the Vulcan framework as a good method for conducting vulnerability assessments in the cloud. The framework uses NVD as its source of information on vulnerabilities and in turn generates the list of identified vulnerabilities to the testers.

Additionally, penetration testing in the cloud is discussed. However, it is important to note the recommendation provided in the research paper - penetration testing in a cloud environment should be conducted on isolated systems. When performed on live systems – they may get overflown with traffic from the testing process and become unable to process legitimate requests.

The discussed service for penetration testing in the cloud is Potassium, which provides results from penetration testing processes with built-in with protection against service deterioration and avoids availability issues. Additionally, the service is tightly linked to the Metasploit framework discussed in [9].

Most authentication and authorization protocols are usually implemented into cloud-based SPs. This research paper brought up relevant information about the current frameworks for assessing vulnerabilities in the cloud and is beneficial for the topic at hand. No hands-on examples are provided for the discussed frameworks, unfortunately. Not related directly to authentication or authorization protocols.

Lachkov et al., "Vulnerability Assessment for Applications Security Through Penetration Simulation and Testing" [14] provides some hands-on examples on conducting penetration testing and vulnerability assessment using the tools described in [9]. Additionally, the paper discusses phases of the penetration testing process, which are often overlooked. The phases include:

1. Pre-Engagement Scoping – test scope definition, which assets need to be tested and how exactly should they be tested.

2. Reconnaissance – passive information gathering of selected assets.

3. Threat Modelling/Vulnerability Identification – identifies vulnerabilities which can be possibly exploited along with design of the threat model relevant to the selected assets.

4. Exploitation – execution of exploitation techniques on the selected assets.

5. Post-Exploitation – documentation of the outcomes of the exploitation phase.

6. Reporting – creation of an official report stating the outcomes of previous phases, methodology, technologies used, etc.

7. Resolution and Re-Testing – can happen when the client wants to re-test their system after the identified vulnerabilities have been fixed.

Examples of hands-on penetration testing are also provided, though they are not relevant to authentication or authorization protocols.

Jiayan et al., "Research on penetration testing procedures based on Kali system" [15] describes the general penetration testing procedures and processes, analyses the toolkit for penetration testing which comes bundled with the Kali Linux operating system. The authors conducted practical experimentation and analysis of the operating system and concluded that it offers the required flexibility, capability and functionality needed for conducting penetration testing. Additionally, the operating system has the tools described in [8] [10] [11] [12] bundled together.

Possible gap here is that these tools are not sufficiently analysed or described.

Rane et al., "Comparative Analysis of Automated Scanning and Manual Penetration Testing for Enhanced Cybersecurity" [16] conducts a vulnerability analysis along with penetration testing on a sample website, tools like Nikto and Netsparker are introduced as suitable for conducting vulnerability assessment. Types of penetration testing methodologies and types of vulnerabilities of web applications are discussed. In the example of Netsparker it was discovered that the automated vulnerability scanning tools can sometimes give out inaccurate or false-positive results. Because of this, the authors

propose to combine manual checking along with automated tools in cases where the result of automated assessment is uncertain.

This paper seems to rely on the idea of combining manual work with automated procedures, but it lacks in-depth analysis of false positives, false negatives and how often they occur in real world scenarios. Additionally, no practical recommendations are provided in combing manual work and automated processes.

Garmabi et al., "Automatic Detection and Risk Assessment of Session Management Vulnerabilities in Web Applications" [17] proposes a solution to identifying session management vulnerabilities discussed in [2]. The statement of the authors is that the current tools for detecting vulnerabilities in session management have certain limitations. Because of this, the authors have developed their own tool, which analyses the traffic between the client browser and the server running the web application. Additionally, the automation can conduct computation of the risk score associated with the scanned traffic.

The proposed tool is limited in the sense that it conducts only unencrypted traffic analysis and ignores more dynamic session management mechanisms. The tool also seems to lack long-term testing in live environments.

Patel, "A Survey on Vulnerability Assessment & Penetration Testing for Secure Communication" [18] highlights the importance of conducting vulnerability assessments and penetration testing procedures to ensure security of systems and software through identification and mitigation. Vulnerability assessment is more relevant to identifying vulnerabilities and loopholes in a system, while penetration testing is aimed at exploiting those vulnerabilities. Most common system vulnerabilities are provided, and some mitigation techniques are also recommended.

Similarly to [2] [3] [5] [17] it is stated that functions related to authentication and session management are usually not implemented correctly in web applications. This in turn can lead to issues ranging from sensitive data exposure to full system compromise. Vulnerability assessment is divided into three subphases:

1. Information gathering – gathering of information about the selected system.
2. Scanning – scanning for possible loopholes and vulnerabilities in the selected system.

3. Result analysis – reporting and analysis of the results of the scanning phase

The authors propose to combine both vulnerability assessment techniques and penetration testing into one whole process: ‘VAPT’.

When compared to separate processes of vulnerability assessment and penetration testing VAPT provides the following benefits:

1. A detailed view of vulnerabilities and threats can be provided.
2. Identification of security issues in the implementation process of the system.
3. Development of an improved risk management process.
4. Enhancement of system’s security against external and internal attacks.

VAPT tools discussed by the authors are:

1. Nessus – commercial vulnerability assessment tool used for scanning networks and web applications.
2. Acunetix – commercial vulnerability assessment tool used for scanning web applications.
3. The Harvester – tool, which is used for information gathering about a specific system.
4. ZAP - vulnerability assessment tool used for scanning web applications.

The authors give a comprehensive description of vulnerability assessment procedures, which is lacking in other research papers. Main issue here is that some commercial tools are described, require monetary investment for usage - this limits the potential of the research scope. Possible disadvantages of VAPT in example of limited skill set are not provided.

Tyagi et al., “Efficient Vulnerability Assessment and Penetration Testing: A Framework for Automation” [19] focuses on developing an automated framework for the VAPT process discussed in [18]. The authors state that currently the VAPT process is constrained by manual procedures, which can be automated. The proposed automation

framework is designed to reduce the knowledge required to conduct effective VAPT by automating the whole process as much as possible. This is achieved by additionally implementing decentralized management and monitoring processes. This setup could reduce costs and improve efficiency, making VAPT more accessible for organizations that may lack specialized cybersecurity specialists.

While the research paper does provide a novel framework, no detailed analysis of resource requirements and cost-efficiency of it are provided. No long-term testing in live environments is conducted, either. Additionally, there is no real efficiency comparison between the framework and the existing VAPT tools.

Lv et al., "Research on vulnerability mining of authentication protocol based on fuzzy simulation" [20] – this research is based on developing a verification model for assessing the security of an authentication protocol during operation. The author proposes a protocol logic vulnerability mining methodology based on the improved noisy-duelling DQN algorithm. The proposed model is more efficient in finding protocol authentication mechanism loopholes and bugs, when compared to other testing techniques and the default DQN algorithm. In the case of this research a simulation of a malicious user and client impersonation during the OAuth 2.0 protocol flow was conducted and successfully identified by the proposed model.

This is an interesting piece of research as it goes beyond the default vulnerability assessment methodologies and conducts the assessment by using a fuzzy simulation model. This model can potentially have higher levels of efficiency and possible judgement when compared to a typical automation tool (ex. [19]) or manual penetration testing. It is also stated that this model can be used for finding vulnerabilities of other protocols and is not only limited to the OAuth 2.0 protocol. However, it is not clear how would it be possible to set this model up for analysing other protocols, no long-term analysis in live environments in conducted.

4 Summary of The Literature Review

This chapter discusses the summary of the literature review, identified gaps in the current state of the art, and answers the stated research questions.

4.1 Results

Twenty research articles were discussed in the scope of the literature review. Topics of the articles were various, but all of them were relevant to the proposed research goal and problem statement. State of the art was identified and answers to all stated research questions are specified.

Each research paper either:

1. Overviewed the existing technologies, tools and protocols.
2. Conducted security analysis and/or vulnerability assessment of the existing technologies and protocols.
3. Proposed new solutions to existing processes.

4.2 Gaps in The State of The Art

Identified gaps in the state of the art are the following:

1. In-development protocol vulnerabilities are not identified.
2. Not all penetration testing and vulnerability assessment tools are properly analysed with a hands-on approach.
3. Long-term testing in live environments is not present for proposed frameworks and models.
4. Qualification issues possibly introduced by combining processes are not discussed.

5. Hands-on examples of the provided vulnerabilities and mitigation techniques of current protocols are not provided.

4.3 Answers to Research Questions

4.3.1 Main Research Question

[RQ1] What is the current state of the art in conducting vulnerability assessment of modern authentication and authorization protocols?

Various authentication and authorization protocols like SAML 2.0, OAuth 2.0, OIDC, LDAP, Kerberos are used. It should be noted that most of the vulnerabilities of authentication and authorization protocols occur when they are improperly implemented or when relevant systems have security flaws. For each vulnerability a corresponding mitigation method should be applied. There are various ways of analysing the security of an implementation of a protocol, but not the protocol itself. Implementation analysis can range from tools for application code logic scanning for security issues to using DQN verification models and automated tools for vulnerability assessment. Vulnerability assessment covers the identification and reporting of vulnerabilities, while penetration testing focuses on exploiting the vulnerabilities and reporting the results. Vulnerability assessment and penetration testing methodologies can be combined into a single process called VAPT.

There are also some gaps in existing research – most novel solutions lack long-term testing. It also seems that for future and in-development protocols like OAuth 2.1 lack proper vulnerability assessment. Most penetration testing and vulnerability assessment tools are brought up, but they are not properly discussed or analysed with a hands-on approach. VAPT can bring up qualification issues as the combined processes require higher level of expertise.

4.3.2 Secondary Research Questions

[RQ1.1] What are the current authentication and authorization protocols?

Current authentication and authorization protocols include the following:

- SAML 2.0

- OAuth 2.0
- OIDC
- LDAP
- Kerberos

Protocols currently in development include:

- OAuth 2.1
- GNAP
- MDQ
- UMA SSI

[RQ1.2] What are the known vulnerabilities related to authentication and authorization protocols and what are the possible mitigations of these vulnerabilities?

These are the identified vulnerabilities of the provided authentication and authorization protocols:

- SAML 2.0:
    - Vulnerabilities:
        - DoS
        - CSRF
        - XSW
        - XXE
        - MITM attacks
        - SAML message timestamp manipulation
        - Invalid signature attack

- Mitigations:
    - Described in **Table 3**., additionally proper implementation of the protocol following the current best practices.

- OAuth 2.0, OIDC:
    - Vulnerabilities:
        - DoS
        - CSRF
        - 307 redirect attacks
        - IdP mix-up
        - XSS
        - MITM attacks
        - Authorization code leakage
        - Unauthorized code injection
    - Mitigations:
        - Described in **Table 3**., additionally proper implementation of the protocol following the current best practices.

- LDAP
    - Vulnerabilities:
        - DDoS
        - LDAP injection attack
    - Mitigations:
        - Firewall configuration

        - Input sanitization

- Kerberos
    - Vulnerabilities:
        - ‘Overpass-the-hash’
        - ‘Pass-the-ticket’
        - ‘Silver ticket’
        - ‘Kerberoasting’
    - Mitigations:
        - Implementation of security measures for systems and devices in the domain.

[RQ1.3] What are the possible methods and tools for conducting vulnerability assessment and penetration testing of authentication and authorization protocols?

- Methods include
    - Penetration testing:
        - Testing variations:
            - White box testing
            - Black box testing
            - Gray box testing
        - Testing phases:
            - Pre-Engagement Scoping
            - Reconnaissance
            - Threat Modelling/Vulnerability Identification

        - Exploitation
        - Post-Exploitation
        - Reporting
        - Resolution and Re-Testing
    - Vulnerability assessment:
        - Phases:
            - Information gathering
            - Scanning
            - Result analysis
    - VAPT – combination of vulnerability assessment and penetration testing processes into a single framework.
- Tools include
    - Kali Linux OS with bundled software
    - Mimikatz
    - Rubeus
    - Hashcat
    - N-Map
    - Wireshark
    - Metasploit framework
    - Potassium
    - Open VAS
    - Beef

- Nessus
- Cain and Abel
- Burp Suite
- SQLmap
- DirBuster
- cURL
- Vulcan framework
- Nikto
- Netsparker
- Acunetix
- The Harvester
- ZAP
- DQN Verification models
- Code security analysers
- Automated frameworks for VAPT
- Super
- MobSF

4.4 Further Research

Further research can be based on any of the research gaps identified in **4.2 Gaps in The State of The Art**. However, vulnerability assessment with the possible use of the VAPT process to assess and exploit the vulnerabilities of in-development protocols seems to be the most promising for extensive results.

Alternatively, hands-on examples of various attacks and mitigation techniques (as described in [5]) of current protocols can be provided for their actuality analysis in 2025.

References

[1] D. Pöhn and W. Hommel, “New Directions and Challenges within Identity and Access Management,” *IEEE Communications Standards Magazine,* vol. 7, no. 2, pp. 84-90, 2023.

[2] Y. Sadqi, Y. Belfaik and S. Safi, “Web OAuth-based SSO Systems Security,” *In Proceedings of the 3rd International Conference on Networking, Information Systems & Security (NISS '20),* vol. 69, pp. 1-7, 2020.

[3] N. Naik and P. Jenkins, “Securing digital identities in the cloud by selecting an apposite Federated Identity Management from SAML, OAuth and OpenID Connect,” *2017 11th International Conference on Research Challenges in Information Science (RCIS),* pp. 163-174, 2017.

[4] I. Indu, P. R. Anand and V. Bhaskar, “Identity and access management in cloud environment: Mechanisms and challenges,” *Engineering Science and Technology, an International Journal,* vol. 21, no. 4, pp. 574-588, 2018.

[5] K. Maidine and A. El-Yahyaoui, “Cloud Identity Management Mechanisms and Issues,” *2023 IEEE 6th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech),* pp. 1-9, 2023.

[6] S. Sharma and J. KP, “Security Analysis of OAuth 2.0 Implementation,” *2023 Innovations in Power and Advanced Computing Technologies (i-PACT),* pp. 1-8, 2023.

[7] A. Sharif, R. Carbone, G. Sciarretta and S. Ranise, “Best current practices for OAuth/OIDC Native Apps: A study of their adoption in popular providers and top-ranked Android clients,” *Journal of Information Security and Applications,* vol. 65, 2022.

[8] A. Thapa, C. S. Dhapola and H. Saini, “Security Analysis of User Authentication and Methods,” *Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing (IC3-2022),* p. 564–572, 2022.

[9] C. D. Motero, J. R. B. Higuera, J. B. Higuera, J. A. S. Montalvo and N. G. Gómez, “On Attacking Kerberos Authentication Protocol in Windows Active Directory Services: A Practical Survey,” *IEEE Access,* vol. 9, pp. 109289-109319, 2021.

[10] A. Chaturvedi, B. Lakhani, T. Agarwal, Mohana, M. Moharir and A. K. A. R, “A Comprehensive Vulnerability Tools Analysis for Security and Control in IT Environment and Organizations,” *2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC),* pp. 612-618, 2024.

[11] H. M. Z. A. Shebli and B. D. Beheshti, “A study on penetration testing process and tools,” *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT),* pp. 1-7, 2018.

[12] K. K. Pareek and G. K. Ameta, “Performance Analysis of Vulnerability Detection Tools and Techniques,” *2024 Parul International Conference on Engineering and Technology (PICET),* pp. 1-5, 2024.

[13] I. Yurtseven and S. Bagriyanik, “A Review of Penetration Testing and Vulnerability Assessment in Cloud Environment,” *2020 Turkish National Software Engineering Symposium (UYMS),* pp. 1-6, 2020.

[14] P. Lachkov, L. Tawalbeh and S. Bhatt, “Vulnerability Assessment for Applications Security Through Penetration Simulation and Testing,” *Journal of Web Engineering,* vol. 21, no. 7, pp. 2187-2208, 2022.

[15] Z. Jiayan, M. Haifei and C. Gengjie, “Research on penetration testing procedures based on Kali system,” *2023 4th International Conference on Computers and Artificial Intelligence Technology (CAIT),* pp. 271-276, 2023.

[16] N. Rane and A. Qureshi, “Comparative Analysis of Automated Scanning and Manual Penetration Testing for Enhanced Cybersecurity,” *2024 12th International Symposium on Digital Forensics and Security (ISDFS),* pp. 1-6, 2024.

[17] N. Garmabi and M. A. Hadavi, “Automatic Detection and Risk Assessment of Session Management Vulnerabilities in Web Applications,” *2021 11th International Conference on Computer Engineering and Knowledge (ICCKE),* pp. 41-47, 2021.

[18] K. Patel, “A Survey on Vulnerability Assessment & Penetration Testing for Secure Communication,” *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI),* pp. 320-325, 2019.

[19] Y. Tyagi, S. Bhardwaj, S. Shekhar and A. P, “Efficient Vulnerability Assessment and Penetration Testing: A Framework for Automation,” *2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES),* pp. 553-557, 2023.

[20] L. Lv, Z. Feng, W. Dong, Y. Zhao and X. Gai, “Research on vulnerability mining of authentication protocol based on fuzzy simulation,” *2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB),* pp. 1-4, 2023.