UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Gregor Eesmaa

Account Existence Leaks in Estonian Online Services

Master's Thesis (30 ECTS)

Supervisor:
Arnis Paršovs, PhD

Tartu 2025

Account Existence Leaks in Estonian Online Services

Abstract:

An account existence leak is a security vulnerability that allows an attacker to determine if a user is registered with an online service. This flaw can be exploited for both large-scale account enumeration and targeted attacks, posing a significant privacy risk and representing a likely violation of the EU's General Data Protection Regulation (GDPR). This thesis investigates the prevalence of this vulnerability across popular Estonian online services that use email addresses as account identifiers. A systematic security analysis of 55 popular Estonian websites and mobile applications was conducted, testing functionalities such as sign-in, password reset, account registration, and email change forms for the vulnerability. This technical analysis was coupled with a multi-stage responsible disclosure process, which included personalised vulnerability reports and formal data subject requests under GDPR for non-compliant service providers. The study found that all tested services were vulnerable to account existence leaks through at least one vector, with account registration and email change forms being the most commonly vulnerable. A majority of tested services, 31 (56%), facilitated stealthy leaks and seemingly lacked basic anti-automation countermeasures, thereby enabling stealthy account enumeration on a massive scale. Both the initial voluntary disclosure and the subsequent formal GDPR requests proved to be effective mechanisms for compelling remediation, ultimately resulting in a 15 (27%) of tested services fully fixing the issue. The findings indicate a systemic cultural and regulatory neglect in preventing account existence leaks. This research provides the first comprehensive large-scale analysis of account existence leaks in the Estonian context, validates a GDPR-based disclosure strategy as a tool for driving compliance, and offers actionable recommendations for service providers, policymakers, and users to mitigate this pervasive privacy risk.

Keywords: Cybersecurity, privacy, GDPR, account existence leak, account enumeration, targeted phishing, large-scale vulnerability testing, large-scale vulnerability disclosure

CERCS: P170 Computer science, numerical analysis, systems, control

Konto olemasolu lekked Eesti veebiteenustes

Kokkuvõte:

Konto olemasolu leke on turvanõrkus, mis võimaldab ründajal tuvastada, kas kasutaja on veebiteenuses registreeritud. Seda viga saab ära kasutada nii laiaulatuslikuks kontode loendamiseks kui ka sihitud rünneteks, mis kujutab endast märkimisväärset privaatsusriski ja on tõenäoliselt vastuolus EL-i isikuandmete kaitse üldmäärusega (IKÜM). Käesolev magistritöö uurib selle haavatavuse levimust populaarsetes Eesti veebiteenustes, mis kasutavad kontotunnusena meiliaadresse. Töös viidi läbi süstemaatiline turvaanalüüs 55 populaarsel Eesti veebilehel ja mobiilirakenduses, testides haavatavuse suhtes selliseid funktsionaalsusi nagu sisselogimine, parooli lähtestamine, kontode registreerimine ja e-posti aadressi muutmine. Tehnilisele analüüsile järgnes mitmeetapiline vastutustundliku avalikustamise protsess, mis hõlmas kohandatud haavatavusaruandeid ja ametlikke IKÜM-i andmesubjekti päringuid nõuetele mittevastavatele teenusepakkujatele. Uuringust selgus, et kõik testitud teenused olid haavatavad konto olemasolu leketele vähemalt ühe vektori kaudu, kusjuures konto registreerimise ja e-posti muutmise vormid olid kõige sagedamini haavatavad. Enamus testitud teenuseid, 31 (56%), võimaldas varjatud lekkeid ja neil nähtavasti puudusid elementaarsed automatiseerimisvastased meetmed, võimaldades seeläbi varjatud ja massilist kontode loendamist. Nii esialgne vabatahtlik avalikustamine kui ka ametlikud IKÜM-i päringud osutusid tõhusateks vahenditeks paranduste esilekutsumisel, mille tulemusel 15 (27%) testitud teenustest vea täielikult parandasid. Tulemused viitavad süsteemsele kultuurilisele ja regulatiivsele hooletusele konto olemasolu lekete ennetamisel. See uurimus on esimene põhjalik ja laiaulatuslik konto olemasolu lekete analüüs Eesti kontekstis, mis kinnitab IKÜM-põhise teavitusstrateegia tõhusust nõuetele vastavuse saavutamiseks ning pakub praktilisi soovitusi teenusepakkujatele, poliitikakujundajatele ja kasutajatele selle laialt levinud privaatsusriski maandamiseks.

Võtmesõnad: Küberturve, privaatsus, IKÜM, konto olemasolu leke, kontode loendamine, sihitud andmepüük, laiaulatuslik haavatavuste testimine, laiaulatuslik haavatavustest teavitamine

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Contents

1. Introduction 8
    1.1 Research Questions 8
    1.2 Use of Artificial Intelligence 9
2. Background 10
    2.1 Vulnerable Functionality and Mitigation Patterns 10
        2.1.1 Sign-In 11
        2.1.2 Password Reset 12
        2.1.3 Account Registration 13
        2.1.4 Changing Account Identifier 15
        2.1.5 The Role of Identifier Confirmation 16
    2.2 Factors Affecting the Attack 17
        2.2.1 Side-Channels 17
        2.2.2 Stealthiness 18
        2.2.3 Anti-Bot Countermeasures 19
    2.3 Threat Model 20
        2.3.1 Mass Account Enumeration of a Service 20
        2.3.2 Targeted Profiling of an Individual 20
    2.4 Legal Aspects 20
    2.5 Implications 23
        2.5.1 Risks to Individuals 23
        2.5.2 Risks to Service Providers 23
    2.6 Related Work 24
        2.6.1 History 24
        2.6.2 Prevalence 24
        2.6.3 Prevention 25
3. Methodology 26
    3.1 Selection of Websites and Mobile Apps 26
    3.2 Subjective Additions to the Target List 28
    3.3 The List of Services to be Tested 29
    3.4 General Testing Procedure 30
        3.4.1 Systematic Testing Approach 30
        3.4.2 Mobile Application Testing Specifics 31

3.5 Testing Variations 31
3.5.1 Account Registration Form 31
3.5.2 Sign-In Form 31
3.5.3 Password Reset Form 32
3.5.4 Email Change Form 32
3.5.5 Special Cases 32
3.6 Data Gathering and Analysis 32
3.6.1 Data Recording with HAR Files 33
3.6.2 Vulnerability Detection 33
3.6.3 Stealthiness Detection 33
3.6.4 Side-Channel Timing Analysis 34
3.6.5 Detection of Protective Measures 34
3.6.6 Manual Review and Overrides 34
3.7 Tooling and Automation Exploration 34
3.8 Received Emails Verification 35
3.9 Ethical Considerations and Responsible Disclosure 35
3.10 Methodological Limitations 35
4. Findings 37
4.1 Overview of Vulnerability Landscape 37
4.2 Findings by Vulnerability Vector 39
4.2.1 Sign-in Forms 39
4.2.2 Password Reset Forms 40
4.2.3 Account Registration Forms 40
4.2.4 Email Change Forms 41
4.3 Side-Channel Vulnerability Findings 41
4.4 Platform Patterns 42
4.5 Anti-Bot Measures 45
5. Vulnerability Disclosure 47
5.1 Preliminary AKI Interaction 47
5.2 Phase 1: Initial Vulnerability Reporting 48
5.2.1 Preparation of Disclosure Materials 48
5.2.2 Contact and Notification 49
5.2.3 Responses to Initial Vulnerability Reports 50

5.3 Phase 2: Reassessment Follow-Up 51
5.4 Phase 3: Escalation as a Data Subject 53
5.4.1 Responses to GDPR Requests 55
5.4.2 Justifications and Arguments Against Fixing 58
5.4.3 Other Notable Patterns 60
5.4.4 Notable Individual Responses 61
5.5 Outcomes 63
6. Discussion and Recommendations 66
6.1 A Global Blind Spot 66
6.2 The 80/20 Security Culture 66
6.2.1 Risk of Shared Platforms 67
6.2.2 Effectiveness of Countermeasures 67
6.3 Barriers to Remediation 68
6.4 Recommendations 68
6.4.1 For Service Providers 69
6.4.2 For Data Protection Authorities (DPAs) 69
6.4.3 For Users 70
6.5 Future Work 70
7. Conclusion 71
References 72
Appendices 75
A. Inclusions From Top 150 .ee Websites in Tranco List 75
B. Inclusions From Top 20 Android Apps in Play Store 80
C. Inclusions From Top 20 iOS Apps in App Store 81
D. Example Initial Disclosure Email 82
E. Example Initial Disclosure Report 83
F. Example Reassessment Report Email 90
G. Example Reassessment Report 91
H. Example GDPR Request Email 97
I. Example GDPR Request 98
J. Analysis Results 100
J.1 Results of First Analysis 101
J.2 Results After Reassessment 102

J.3 Results After Partial Second Reassessment..... 103
J.4 Results After Partial Final Assessment 104
K. Correspondence Log..... 105
License 120

1. Introduction

This thesis provides the first focused analysis of the account existence leaks within online services targeted at Estonians that use email addresses as account identifiers. The security flaw allows an attacker to determine if an identifier, such as an email address, is registered with a service [1]. While the issue has been known for decades [2], it remains pervasive [3, 4], often arising from a fundamental conflict between providing a helpful user experience and ensuring robust security. For example, a registration form that explicitly responds "*Email already in use*" on an attempt to register an account using an email that is already registered, helps a legitimate user but also discloses information to an attacker.

The threat manifests in two primary attack scenarios: mapping the user base of a service based on an existing list of emails, or profiling a single individual across multiple services. For an individual, the unauthorised disclosure of their association with a service is likely a privacy violation under the EU's General Data Protection Regulation (GDPR) [5], where personal data is defined very broadly. This exposure can reveal sensitive details, leading to risks such as highly convincing targeted phishing, blackmail based on an individual's association with a sensitive platform (e.g., a dating or gambling service), or discrimination based on inferred political views, financial status or health concerns. For service providers, the vulnerability poses a dual threat: competitors can enumerate customer lists, potentially violating trade secret protections [6], and attackers can perform reconnaissance for sophisticated cyberattacks like credential stuffing and account takeover [7].

To provide a comprehensive overview of this problem in the Estonian context, an investigation was conducted into the prevalence and nature of email-based account existence leaks across 55 popular Estonian online services. The technical vectors were examined, the effectiveness of existing countermeasures was assessed, and the responses of service providers to responsible disclosure efforts were documented.

1.1 Research Questions

This thesis aims to answer the following research questions:

- **RQ1:** How widespread are account existence leaks across popular Estonian online services, and which service functionalities are most commonly vulnerable?
- **RQ2:** What security measures are employed by Estonian online services to mitigate account existence leaks, and how effective are they?

- **RQ3:** How do Estonian service providers respond to the vulnerability reports of these vulnerabilities and formal GDPR data subject requests?

1.2 Use of Artificial Intelligence

In accordance with the University of Tartu guidelines [8], a large language model (LLM), specifically Google's Gemini 2.5 Pro, was used as a supportive tool during the research process. The LLM was used as an aid in editing and formatting thesis draft, generating and debugging automation scripts, formulating search queries for literature reviews, copy-editing reports and requests sent to service providers.

In every instance, the LLM-generated content was treated as a preliminary suggestion. All outputs were critically evaluated, fact-checked, and significantly revised by the author to ensure the accuracy, validity, and originality of the final work. Full responsibility for the content presented in this thesis rests solely with the author, thereby upholding the principles of academic integrity.

2. Background

This section provides an overview of account existence leaks. The section describes the common technical vectors through which this vulnerability is exploited, the concept of stealthiness, and the potential implications for individuals and service providers. The section also situates this study within the context of existing research and industry guidelines.

2.1 Vulnerable Functionality and Mitigation Patterns

Any functionality within an online service that accepts an account identifier (e.g., email address, phone number or username) as input is potentially vulnerable to leaking account existence. However, in this thesis, the main focus is on the use of email addresses as account identifiers, as this identifier is commonly used by modern services and uniquely links to an individual.

At its core, an account existence leak is caused by an observable response discrepancy that reveals whether a provided identifier matches an existing account [2]. The discrepancy can manifest in various ways, for example: visible error messages ("*user not found*" vs. "*incorrect password*"), different server responses (HTTP status codes, HTML structure, JSON data), or even subtle timing side-channels. Effective mitigation requires ensuring that responses are indistinguishable, regardless of whether the account exists [9].

Such vulnerabilities commonly appear in authentication and account management features, including:

- Sign-in
- Password reset
- Account registration
- Account identifier change (e.g., email change)[1]

However, any functionality that processes account identifiers could also be susceptible. The following sections illustrate how this vulnerability manifests in these common forms and describe the secure design patterns that prevent it.

[1]This vector is often overlooked, even in security guidelines like OWASP ASVS 5.0 [10], where requirement 6.3.8 fails to mention identifier change functionality.

2.1.1 Sign-In

A vulnerable sign-in form discloses account existence by having distinguishable responses depending on whether an account with the given identifier exists or not. For example, a vulnerable sign-in form might display "*No account found with this email*" for an unregistered identifier but "*The given password is incorrect*" for a registered identifier with an incorrect password (see Figure 1). In contrast, a secure sign-in form provides an indistinguishable response for any type of failure (see Figure 2).

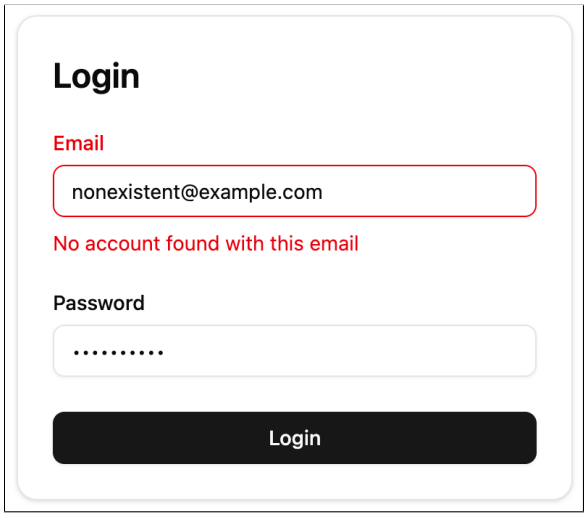

Figure 1. Vulnerable sign-in form

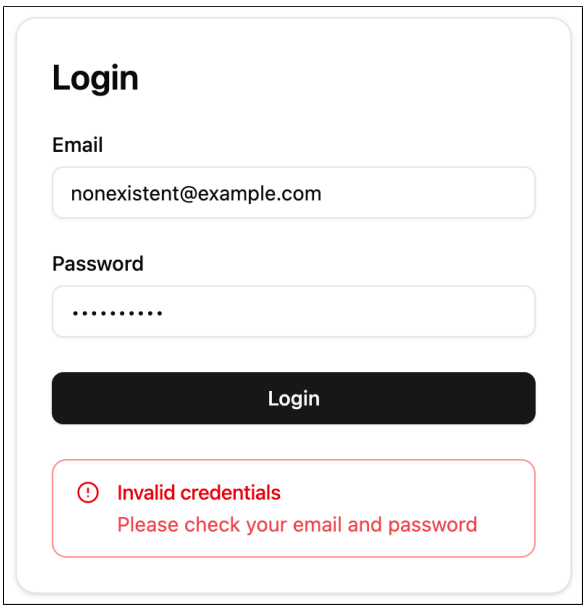

Figure 2. Secure sign-in form

Recommended pattern for email identifiers. Upon any submission error, the form should always display a generic message such as "*The email or password you entered is incorrect*". The server must provide an indistinguishable, constant-time response (e.g., an HTTP `401 Unauthorized` status with no additional content).

Anti-automation measures, if any, should be enforced before the request is processed and without reading user-provided data. Other conditions preventing sign-in (e.g., an unactivated, locked or deactivated account) should only be revealed after a successful authentication attempt.

Justification of recommendation. The recommended messaging aims to conceal which of the inputs was incorrect. The indistinguishable response should be designed to reduce the risk of the vulnerability being reintroduced by later changes (e.g., by strictly requiring the API error response to contain nothing beyond the status code).

To prevent timing side-channel leaks, great care must be taken to ensure the server performs an equivalent amount of processing for both valid and invalid attempts. Account lookups, for instance, must always ensure uniform execution time – especially when querying multiple sources, all responses must be awaited even if an account is found early. Furthermore, the password matching process must execute dummy cryptographic computations for non-existent accounts, making its duration indistinguishable from that of a valid attempt. Rate limits, CAPTCHA enforcement, or other logic that prevents sign-in (e.g., unconfirmed emails, temporarily locked out or deactivated accounts) must not produce observably different responses for existing and non-existent accounts to an unauthorised party.

2.1.2 Password Reset

A vulnerable password reset form discloses account existence by having distinguishable responses depending on whether an account with the given identifier exists or not. For example, a vulnerable password reset form might display "*No account found with this email*" for an unregistered identifier (see Figure 3). In contrast, a secure password reset form provides an indistinguishable response regardless of whether the identifier exists (see Figure 4).

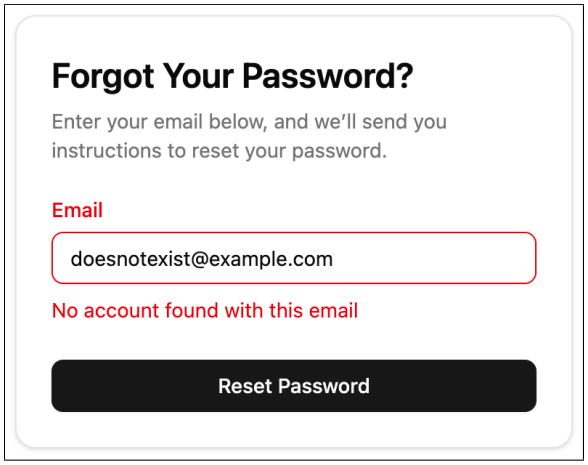

Figure 3. Vulnerable password reset form

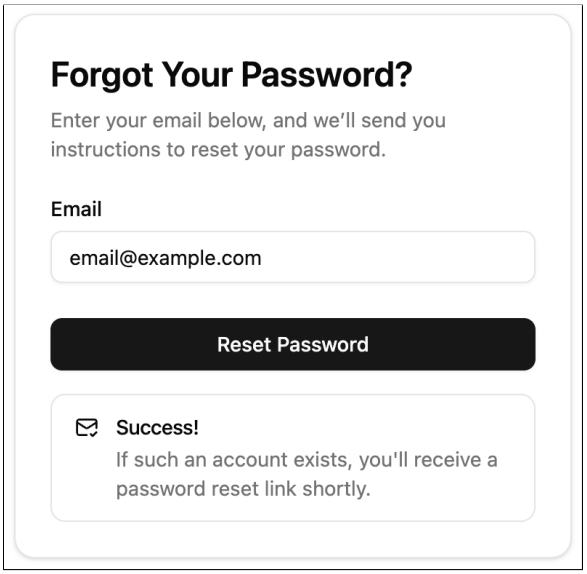

Figure 4. Secure password reset form

Recommended pattern for email identifiers. Upon submission, the form should display "*Please check your email for instructions. If you do not receive an email, please double-check your email address*". The server should always respond immediately (e.g., with HTTP status `202 Accepted`) before starting to process the request. An email should always be sent to the provided address. For an existing user, the email should contain a password reset link. For

a non-existent account, the email should explain that an account does not exist, while providing a registration confirmation link.

Anti-automation measures (if any) should be enforced prior to handling the request without reading the user-provided data. Any other reason an account cannot have its password reset (e.g., it is unactivated, locked or deactivated) should also be explained only over email.

Justification of recommendation. The recommended messaging aims to conceal whether the reset was successful, while acknowledging that typos can result in not receiving an email. To ensure both uniform response times and that no details about the account existence or status could be exposed in the response, the processing of the requested operation should be deferred to after returning a response. This deferred request handling also reduces the risk of the vulnerability being inadvertently reintroduced, as it would require a more significant change to check for account existence or status before responding.

Sending an email for non-existent accounts aims to provide a good user experience by ensuring clarity, especially as email non-receipt would then more strongly indicate input errors. Sending emails in both cases also prevents a side-channel where email delivery or non-delivery could be monitored by an attacker (e.g., email receipt sound in close proximity).

Rate limits, CAPTCHA enforcement or any other logic preventing password resets (e.g., unconfirmed emails, temporarily locked out or deactivated accounts) must not result in observably different responses for existing and non-existent accounts.

2.1.3 Account Registration

A vulnerable account registration form discloses account existence by having distinguishable responses depending on whether an account with the given identifier exists or not. For example, a vulnerable account registration form might display "*Email already in use*" for a registered identifier (see Figure 5). In contrast, a secure registration form provides an indistinguishable response, preventing an attacker from confirming if an email is already registered (see Figure 6).

Recommended pattern for email identifiers. The recommended approach is a two-step process where the email address is confirmed first. Upon submission, the form should display "*Please check your email for further instructions. If you do not receive an email, please double-check your email address*". The server should always respond immediately (e.g., with HTTP status `202 Accepted`) before starting to process the request.

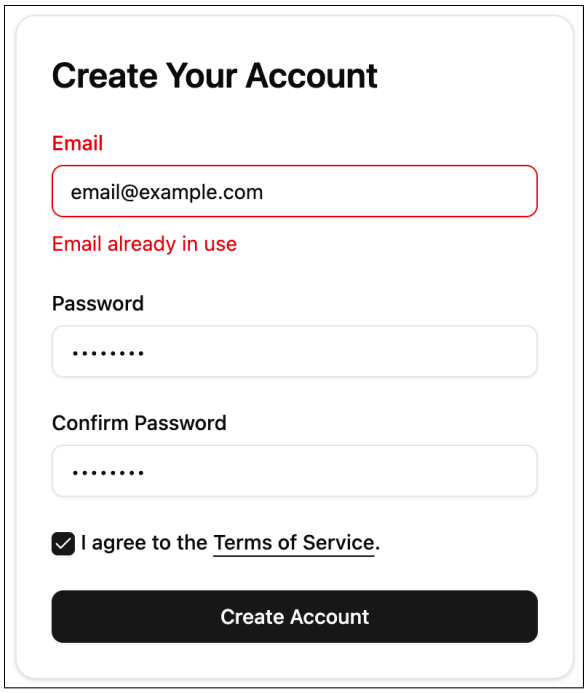

Figure 5. Vulnerable registration form

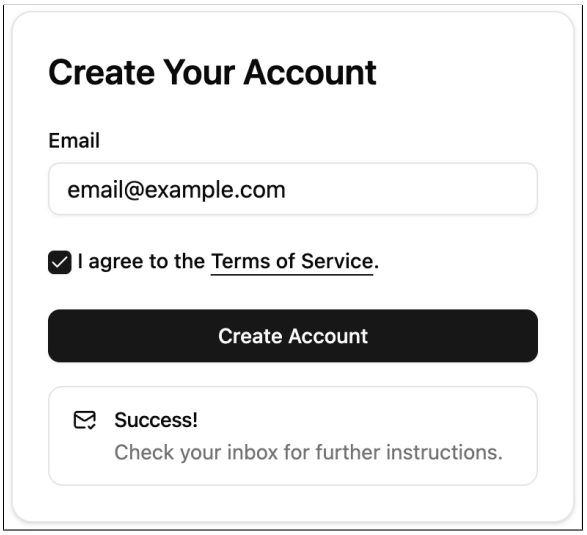

Figure 6. Secure registration form

An email should always be sent to the provided address. If the email address is new, the email should contain a link to confirm the email address and complete the registration (e.g., to fill out other details). For an existing account, the email should explain that an account already exists and provide a password reset link.

Any other reason an account cannot be registered (e.g., it is locked or deactivated) should also be explained only over email.

Justification of recommendation. Confirming a user's email address before they fill out the rest of the registration form is the recommended approach for three key reasons. Firstly, it improves the user experience, as it prevents time wasted filling out a form that will be discarded if the email is already in use (confirmation is required anyway, see Section 2.1.5). Secondly, this pattern is easier to secure against account existence leaks because unconfirmed accounts either need to be kept separate from registered accounts or need special handling to avoid accidentally leaking information elsewhere on the service (e.g., login page, user counters, etc.). Finally, having a distinct first step for email confirmation is more difficult to change, reducing the risk that the vulnerability is inadvertently reintroduced.

The recommended messaging aims to conceal whether the registration was successful, while acknowledging that typos can result in not receiving an email. To ensure both uniform response times and that no details about the account existence or status could be exposed in the response,

the processing of the requested operation should be deferred to after returning a response. This deferred request handling also reduces the risk of the vulnerability being inadvertently reintroduced, as it would require a more significant change to check for account existence or status before responding.

Sending an email for existing accounts aims to provide a good user experience by ensuring clarity, especially as email non-receipt would then more strongly indicate input errors. Sending emails in both cases also prevents a side-channel where email delivery or non-delivery could be monitored by an attacker (e.g., email receipt sound in close proximity). Any other situations preventing registration (e.g., temporarily locked out or deactivated accounts) must not result in observably different responses for existing and non-existent accounts, but may be explained over email, following the standard submission flow up until the email is composed.

2.1.4 Changing Account Identifier

A vulnerable account identifier change form discloses account existence by having distinguishable responses depending on whether an account with the given identifier exists or not. For example, a vulnerable account identifier change form might display "*Email already in use*" for a registered identifier (see Figure 7). In contrast, a secure account identifier change form provides an indistinguishable response (see Figure 8).

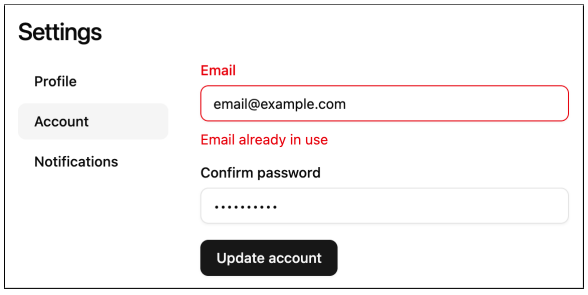

Figure 7. Vulnerable email change form

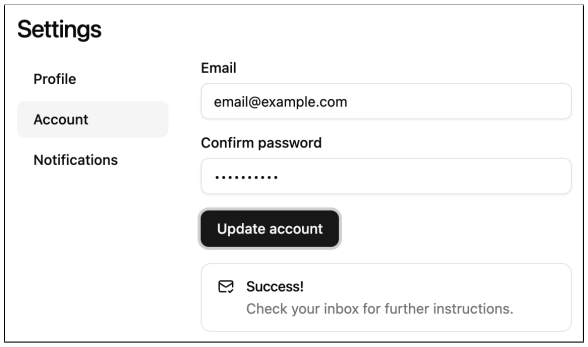

Figure 8. Secure email change form

Recommended pattern for email identifiers. Upon submission, the form should display "*Please check your new email for further instructions. If you did not receive an email, please double-check your new email address*". The server should always respond immediately (e.g., with HTTP status `202 Accepted`) before starting to process the request.

An email should always be sent to the provided new address. If the new email address has no associated account, the email should contain a link to confirm the email address change. If the new email address already has an existing account, the email should explain the situation.

Any emails sent to the old email address as per OWASP guidelines [9] should have indistinguishable content regardless of the status of the new email address.

Any other reason an account is unable to have its email changed (e.g., it is locked or deactivated) should also be explained only over email.

Justification of recommendation. The recommended messaging aims to conceal whether the email change request is successful, while acknowledging that typos can result in not receiving an email. To ensure both uniform response times and that no details about the account existence or status could be exposed in the response, the processing of the requested operation should be deferred to after returning a response. This deferred request handling also reduces the risk of the vulnerability being inadvertently reintroduced, as it would require a more significant change to check for account existence or status before responding.

Sending an email for existing accounts aims to provide a good user experience by ensuring clarity, especially as email non-receipt would then more strongly indicate input errors. Sending emails in both cases also prevents a side-channel where email delivery or non-delivery could be monitored by an attacker (e.g., email receipt sound in close proximity). OWASP guidelines [9] require a notification or a confirmation sent to the old email address as well, but those emails must not leak information about account existence associated with the new email, since it cannot be guaranteed that both email addresses are controlled by the same person.

Any other situations preventing an email change (e.g., temporarily locked out or deactivated accounts) must not result in observably different responses for existing and non-existent accounts, but may be explained over email (only to the new email address), following the standard submission flow up until the email is composed.

2.1.5 The Role of Identifier Confirmation

Services commonly require users to confirm their email addresses to ensure that account-specific communication is received by the legitimate holder of the email. In practice, this confirmation is implemented during registration by emailing a confirmation link or confirmation code.

If email confirmation is not enforced by the service and the user specifies an email address they have no access to, then there is no guarantee that any account-management-related communication (including password resets) could be received by the account holder.

Additionally, a lack of identifier confirmation could be interpreted as a failure to ensure data accuracy under GDPR Article 5(1)(d), which states that ("*Personal data shall be:*") "*accurate*

and, where necessary, kept up to date; every reasonable step must be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay ('accuracy')". GDPR Article 5(2) additionally states that "*The controller shall be responsible for, and be able to demonstrate compliance with, paragraph 1 ('accountability')*".

Furthermore, the lack of email confirmation inherently enables account existence leaks by forcing one of two possible behaviours on registration and email change forms. Firstly, the forms can let the user know that the email is already registered, leading to an obvious account existence leak. Secondly, the forms can return success message without actually creating the account (or changing the email), but an attacker can detect this by checking whether the supposedly created account can be logged into.

Consequently, a prerequisite for preventing account existence leaks on registration or email change forms is to send a confirmation email or a notification of account existence.

Similarly, control of a phone number can also be confirmed by, e.g., sending a verification code over SMS. However, confirming usernames is more problematic, as there is no way to claim ownership. Therefore, if a service wants to offer user-chosen usernames, there is no way to prevent account existence leaks during registration or username changes.

2.2 Factors Affecting the Attack

The impact of an account existence leak and the ease of its exploitation vary based on several key factors. These factors include the availability of subtle attack vectors, the ability to remain undetected, and the presence of anti-automation measures.

2.2.1 Side-Channels

Beyond explicit error messages, the potential for exploitation is expanded by the existence of side-channels – indirect information disclosures. For example, a timing attack can be successful even if a service provides generic error messages. If an attacker can measure a consistent difference in server response time between a request for an existing user and a non-existent one, they can still effectively determine account existence. Other side-channels include differences in HTTP response headers or variations in non-visible page content.

Such implicit disclosures can be more difficult to find than explicit statements about account existence, requiring specialised tools for analysing responses. However, they can also be more

difficult to fully mitigate, potentially requiring revisions in multiple layers of authentication-related logic to ensure responses are indistinguishable in both content and timing.

Different messaging for locked out (i.e., too many failed sign-in attempts), banned, or otherwise deactivated accounts could also enable account existence leaks, when the messages appear only for registered accounts.

2.2.2 Stealthiness

The effectiveness and impact of an attack are significantly increased if account existence can be checked with stealth, thereby avoiding detection by the user. An attack is considered stealthy if the account owner does not receive any notification about the check being performed.

The existence of an account can usually be checked stealthily on sign-in forms, as users are generally not notified about singular failed sign-in attempts.

On other vulnerable forms, stealthiness can be asymmetrical, depending on whether the tested account exists. For example, on an email-based password reset form, checking for an existing account is typically non-stealthy because the user receives a password reset email. In contrast, checking for a non-existent account on the same form is typically stealthy, given no notification is sent.

Account registration and identifier change forms can exhibit the opposite asymmetry. Checking for a non-existent account (i.e., a successful registration attempt) is often non-stealthy, as it triggers a welcome or confirmation email. Conversely, checking for an existing account might be stealthy when it returns only an error message like "*Email already in use*" without sending any emails. However, this behaviour is not universal, as some services might not enforce email confirmation at all.

However, a fully stealthy check – one that avoids notifications regardless of account status – can be achieved through a technique overlooked by the earlier Swiss study by Maceiras et al. [3]. This method is possible if a server validates an identifier's existence before other form fields. An attacker can submit an identifier to a vulnerable registration form while intentionally failing a secondary validation step, such as a CAPTCHA or password confirmation check (see Figure 9). If the server's validation logic prioritises the identifier check, it will return an error like "*Email already in use*" for an existing account, while returning a different error (e.g., "*Passwords do not match*") for a non-existent account. In either case, the overall form submission fails, preventing any notification email from being sent and thus guaranteeing a stealthy check.

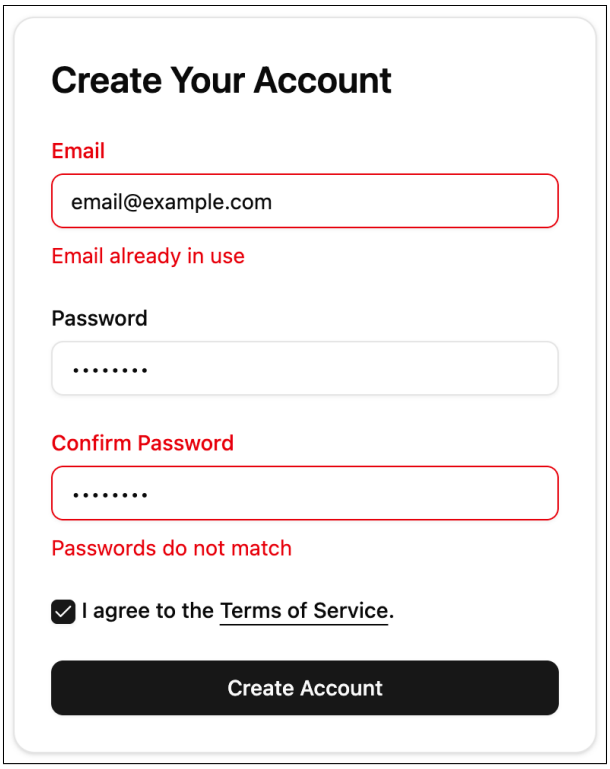

Figure 9. A registration form with two validation errors, guaranteeing a stealthy account existence leak

Even when an attack is not stealthy, the primary leak has already occurred by the time the user is notified. However, such notifications are not without consequence for the attacker. They can alert the target to being under surveillance and, if occurring at scale, can signal the presence of an attack to the service provider. The service provider could then employ countermeasures such as blocking the attacker's IP address, enforcing rate limits or introducing CAPTCHAs, and potentially remediate the vulnerability in the long term.

2.2.3 Anti-Bot Countermeasures

To combat automated attempts to discover account existence (account enumeration), services can employ several defences. The most common is the use of CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart), which is designed to distinguish human users from automated scripts. Another key defence is rate limiting, which restricts the number of requests a single actor can make in a given period. If implemented correctly, these measures can make large-scale, automated attacks to discover account existence more difficult and costly for an attacker to carry out, forcing the attacker to check for accounts manually.

2.3 Threat Model

This section defines two primary threat models related to account existence leaks, distinguished by their motivations and methods. The analysis of risks and countermeasures throughout this thesis is contextualised by the capabilities and goals associated with these two models.

2.3.1 Mass Account Enumeration of a Service

This threat model involves an attacker performing large-scale harvesting of user lists from a service. The motivation may be commercial (e.g., building marketing lists, competitive intelligence) or malicious (e.g., creating a database of targets for widespread phishing or credential stuffing attacks). The attack is executed using automated scripts to test large dictionaries of identifiers (emails, mobile numbers or usernames) against a single service, with a focus on the efficiency and scalability of the enumeration method.

2.3.2 Targeted Profiling of an Individual

This threat model focuses on an attacker targeting a specific individual or a small group. The motivation is often personal, such as stalking, blackmail, social engineering or intelligence gathering (OSINT). The objective is to determine whether a specific person uses a particular service to build a profile of their activities, associations and vulnerabilities. For this purpose, even a slow or manual method of checking for account existence is valuable if it confirms a single, critical piece of information.

2.4 Legal Aspects

The General Data Protection Regulation (GDPR) [5] mandates the protection of any personal data. An email address, phone number or other unique identifier, and the fact of its association (or lack thereof) with a specific service, are personal data under GDPR Article 4(1): "*'personal data' means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person*".

GDPR applies to any operations ("*processing*") performed on personal data. Entities, that control the purposes and means of processing are referred to as "*data controllers*". GDPR specifically requires data controllers to ensure data protection by design and by default, which is particularly highlighted in GDPR Article 25(2)(1): "*The controller shall implement appropriate*

technical and organisational measures for ensuring that, by default, only personal data which are necessary for each specific purpose of the processing are processed" and GDPR Article 25(2)(3) "*In particular, such measures shall ensure that by default personal data are not made accessible without the individual's intervention to an indefinite number of natural persons*".

Unauthorised disclosure via an account existence leak in most cases likely constitutes a data processing activity without a valid legal basis (violating GDPR Article 6(1): "*Processing shall be lawful only if and to the extent that at least one of the following applies: [...]*") and potentially a failure to ensure appropriate security (violating GDPR Article 32 "*Taking into account the state of the art, the costs of implementation and the nature, scope, context and purposes of processing as well as the risk of varying likelihood and severity for the rights and freedoms of natural persons, the controller and the processor shall implement appropriate technical and organisational measures to ensure a level of security appropriate to the risk [...]*"), thus constituting a data breach as defined by GDPR Article 4(12) "*'personal data breach' means a breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorised disclosure of, or access to, personal data transmitted, stored or otherwise processed*". Enforcement lies with the national Data Protection Authorities in the European Economic Area. In Estonia, that authority is the Estonian Data Protection Inspectorate (Andmekaitse Inspektsioon, AKI). As will be detailed in Section 5.1, the inspectorate confirmed that leaking account existence is an issue data controllers must address.

For reference, the legal bases for processing personal data as per GDPR Article 6(1) are as follows:

a "*the **data subject has given consent** to the processing of his or her personal data for one or more specific purposes*"

b "*processing is **necessary for the performance of a contract** to which the data subject is party or in order to take steps at the request of the data subject prior to entering into a contract*"

c "*processing is **necessary for compliance with a legal obligation** to which the controller is subject*"

d "*processing is **necessary in order to protect the vital interests** of the data subject or of another natural person*"

e "*processing is **necessary for the performance of a task carried out in the public interest** or in the exercise of official authority vested in the controller*"

f "*processing is **necessary for the purposes of the legitimate interests** pursued by the controller or by a third party, except where such interests are overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data, in particular where the data subject is a child*"

Regarding consent, GDPR defines it as a "*freely given, specific, informed and unambiguous decision*" (Article 4(11)) and also states that the "*data subject shall have the right to withdraw his or her consent at any time*", adding that "*It shall be as easy to withdraw as to give consent*" (Article 7(3)). Getting user consent specifically for receiving feedback on authentication-related forms, while keeping them informed of the consequences that someone else could verify their existence, is likely not practical. It is also noteworthy that consent would apply only to the account identifiers of the consenting users. This limitation would render the feedback functionality much less useful, as it could not lawfully provide feedback on mistyped identifiers.

The rest of the legal bases (b-f) explicitly require necessity. For most services, it is likely not necessary to expose account existence to any unauthorised third party to perform a contract, comply with a legal obligation, protect vital interests, perform a public task or pursue legitimate interests. Given that mitigation strategies exist (as detailed in Section 2.1), these bases would generally not be applicable.

A notable exception may apply to platforms, such as social networks, for which user discovery is an integral feature. In these specific contexts, the data processing that facilitates this functionality could plausibly be argued as a contractual necessity. However, given that discoverability is often an optional feature rather than a prerequisite for using the service, a more appropriate legal basis would be the user's explicit consent. Relying on consent aligns with the principle of data protection by design, as it provides users with granular control over whether their presence on the service is public.

As a side note, confirming an account does not exist is likely still considered data processing under GDPR. Since a non-user has no relationship with the service, there is arguably no legal basis for revealing their lack of an account. This situation presents a likely legal grey area for services that rely on user discovery as a core function, especially where contractual necessity might be used for handling such inquiries.

As later detailed in Section 5.4.2 and Section 5.5, many service providers still attempted to claim some legal bases for their implementations that are vulnerable to account existence leaks.

2.5 Implications

Account existence leaks enable various malicious activities and carry significant implications. The focus of this work is on the Estonian context, acknowledging that legal frameworks and user expectations may differ elsewhere.

2.5.1 Risks to Individuals

Targeted Attacks. Knowing which services a person uses enables highly targeted attacks. An attacker can craft convincing phishing campaigns (spear phishing), execute social engineering attempts, or try to compromise accounts through methods like credential stuffing and password spraying, which can lead to identity theft. Indeed, people report being significantly more likely to click on links in phishing emails that impersonate services they actually use [3].

Privacy Violation and Profiling. Revealing associations with specific services can expose sensitive personal information, and even confirming non-association can be informative for profiling purposes. For example, an attacker could infer an individual's political leanings (e.g., news portals), financial situation (e.g., credit services), sensitive interests (e.g., dating sites), addictions (e.g., gambling sites), family status (e.g., school portals) or general consumer behaviour (e.g., e-commerce sites).

Blackmail and Bullying. Information about sensitive service usage, such as activity on dating-, gambling- or health-related sites, can be used for extortion or harassment.

Open-Source Intelligence (OSINT). Law enforcement or intelligence agencies might check for account existence as part of open-source intelligence gathering, which is concerning for its privacy implications [3].

2.5.2 Risks to Service Providers

Competitive Disadvantage. Competitors could discover user lists to gain market intelligence, target users with advertising, poach customers, or analyse business initiatives. Such activities could potentially violate unfair competition and trade secret regulations [6]. Detecting and attributing exploitation of this vulnerability is challenging, making proactive prevention by service providers the most effective approach.

Reputational Damage. Public disclosure of such vulnerabilities can damage user trust and brand reputation.

Legal Consequences. Service providers may face investigations and sanctions from the relevant Data Protection Authority (DPA) if breaches are confirmed and not adequately addressed. DPAs can initiate proceedings based on complaints or on their own initiative.

However, there is no known precedent of a DPA taking action against a service provider for vulnerabilities that leak account existence.

2.6 Related Work

The vulnerability of leaking account existence is not a new phenomenon; its history is deeply intertwined with the evolution of networked computing and the shifting norms of digital privacy. This section situates the current study within this broader context by tracing the vulnerability's history, examining key academic and industry research, and highlighting how this thesis builds upon and contributes to existing knowledge.

2.6.1 History

Early entries in the Common Vulnerabilities and Exposures (CVE) system document a recurring pattern of account existence leaks through distinguishable system responses. These vulnerabilities ranged from leaking existence "*by design*" (CVE-1999-0656) to subtler variations in outputs and error messages across common software (CVE-2001-1483, CVE-2001-1013, CVE-2001-1528, CVE-2004-2150) [11–15]. This historical pattern highlights the tension between user experience and security, where a seemingly helpful error message like "*Username not found*" can be weaponised.

This class of vulnerability was formally grouped by Mitre Corporation in 2005 as "*Response discrepancy infoleaks*" under the PLOVER (Preliminary List Of Vulnerability Examples for Researchers) initiative [16, 17]. This effort evolved into the Common Weakness Enumeration (CWE) system, which now classifies this weakness as CWE-204, "*Observable Response Discrepancy*" [2].

2.6.2 Prevalence

The most significant recent overview on account existence leaks is the work of Maceiras et al. [3], who conducted a multifaceted study of online services popular in Switzerland. Their research found that 59 (94%) out of 63 services tested were vulnerable. Similarly, a less recent study

by Hasegawa et al. [4] found that 86 (99%) out of 87 services tested leaked account existence information through inconsistent messages. They also framed the problem as a privacy issue in the context of insiders, i.e., family, friends or other acquaintances.

While this study focuses primarily on discovering account existence via email addresses, some earlier research has demonstrated a similar set of problems with numeric identifiers, albeit exposing even more than just account existence. For example, case studies on KakaoTalk messenger and Facebook have shown how phone numbers could be incrementally enumerated in user-search functionalities to harvest various personal data [18, 19]. Similar enumeration techniques have been shown to be effective in package tracking systems [20], leaking addresses, and dating apps [21], leaking identities. A notable real-world incident in the Baltic region was a 2013 Swedbank data leak, where a list of user IDs and names were exposed [22].

While these studies establish the international prevalence of enumeration and account existence leaks, this thesis provides the first analysis focused specifically on the Estonian digital ecosystem, examining the issue through the unique lens of its advanced digital society and its implementation of GDPR.

2.6.3 Prevention

Industry best practices for mitigating account existence leaks are now well-established, with the Open Worldwide Application Security Project (OWASP) being a common reference. Its Authentication Cheat Sheet [9] provides clear guidance on using generic, indistinguishable responses to prevent an attacker from differentiating between existing and non-existent accounts. However, as noted by this thesis, these guidelines can have blind spots, such as overlooking the email change functionality as a potential vector for leaking account existence.

Conversely, Ambreen et al. [23] note that Small and Medium Enterprises (SMEs) are often vulnerable to cyberattacks due to a lack of resources and expertise, which may help explain the apparent prevalence of account existence leaks. Also, Gorski et al. [24] found that developers often skip security-related documentation to find code examples that help them solve their problems. These two works indicate that any proposals to be made in this thesis should be easy to read and comprehend, even for developers with limited security expertise.

3. Methodology

The methodology used to assess vulnerabilities that leak account existence in popular Estonian online services is detailed in this section. The process involved selecting relevant services, systematically testing for the vulnerability, collecting data using appropriate tooling, analysing the results, and preparing for responsible disclosure.

3.1 Selection of Websites and Mobile Apps

The study focused on the most popular websites and mobile apps used in Estonia to ensure the relevance of the research. While no single, universally accepted metric for measuring website popularity exists, especially on a regional level, Alexa's Top Sites list was a common resource historically but was retired in December 2022 [25].

Several alternative ranking sources (Similarweb, Ahrefs, Semrush) exist, but present methodological biases, limitations in regional specificity, or access restrictions. Local directories (`neti.ee`, `infoweb.ee`, `www.ee`, `1182.ee`) could identify Estonian services, but not rank them by popularity. Consequently, the Tranco list [26] was chosen as it provides a robust, consensus-based ranking. For website selection, the Tranco list[2] generated on October 24, 2024, was used as the primary starting point, from which websites with the top-level domain (TLD) of `.ee` were considered.

To define *Estonian* online services (including websites and mobile apps), the following criteria were considered:

- Services targeted primarily at the Estonian population (e.g., language, content).
- Services operated by companies registered in Estonia, or offering significant physical services within Estonia (indicating a strong local presence/audience).

A service was excluded from the list if it matched any of the following criteria:

- It did not offer creation of user accounts.
- It did not use email-based user identifiers (had only e.g., arbitrary usernames, mobile numbers, eID authentication, or SSO without separate registration).

[2]Available at https://tranco-list.eu/list/249P9.

- Its user identifiers could be only email addresses issued by the service itself (e.g., `mail.ee`).
- It was clearly not targeted at Estonians, despite the TLD (e.g., `linktr.ee`, `sci-hub.ee`).
- It did not load as a website in a browser.
- It redirected to or had shared authentication with another service in the list.

Specific websites included and excluded, along with justifications, are detailed in Appendix A. Of the 150 websites reviewed, 46 (31%) were included in the study. The remaining websites were excluded for various reasons: 26 (17%) did not use email-based identifiers, 28 (19%) did not have user accounts or did not allow registration, and the rest were either not Estonian, not working, duplicates of other services, or email services (see Figure 10).

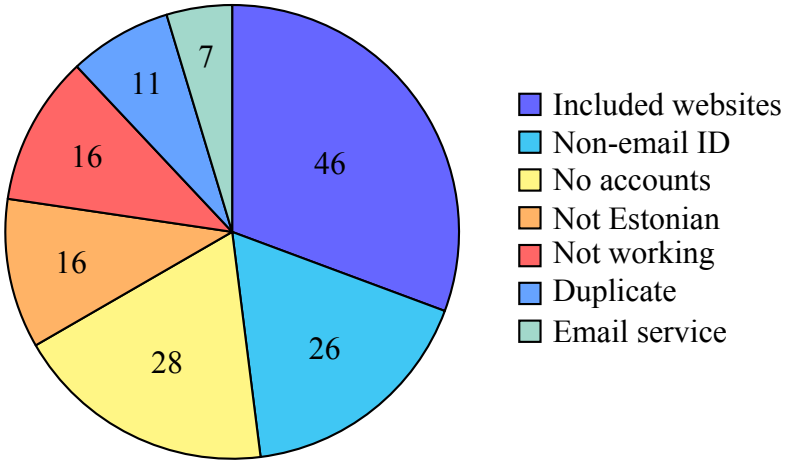

Figure 10. Categorisation of the top 150 `.ee` websites considered

For mobile applications, rankings from the Google Play Store (see Appendix B) and Apple App Store (see Appendix C) for the Estonian region were used. The inclusion criteria for mobile apps mirrored those for websites, focusing on Estonian relevance and email-based accounts.

The inclusion of mobile apps served a secondary role: to demonstrate that leaking account existence is not limited to websites, but online services in general. The sample size of apps was smaller due to a less precise analysis methodology compared to that used for websites (see Section 3.4.2).

Of the 29 apps reviewed, 5 (17%) were included in the study, although a majority, 16 (55%), were not targeted at Estonians. The rest did not support email authentication or relied on WebView

using same authentication flows as their websites, which were already included from the Tranco list (see Figure 11).

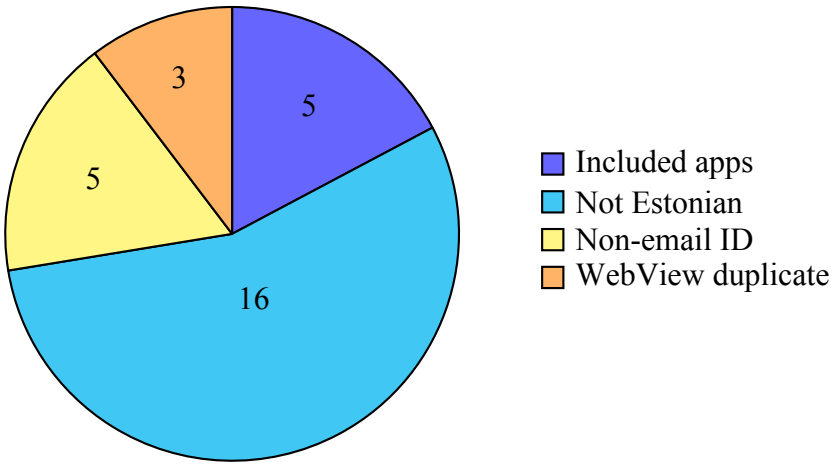

Figure 11. Categorisation of the top 29 apps considered

3.2 Subjective Additions to the Target List

To reach a target list with 50 websites, the initial selection based on popularity rankings was supplemented with websites chosen for their societal importance. This subjective supplementation ensured the analysis would be comprehensive, covering services that, despite lower popularity rankings, represent a relatively high risk.

Specifically, several educational and health-related services were included due to the sensitive nature of their user data. Educational platforms (ekool.eu, eliis.eu, jpg.ope.ee) were included because they are widely used in primary and secondary education and thus process the data of minors. The mental health service peaasi.ee was included as it is a widely promoted resource for mental health education and support, indicating its user base is seeking sensitive health information.

The presence of vulnerabilities that leak account existence on such platforms would be particularly concerning from a data protection perspective. Recital 38 of the GDPR states that "*Children merit specific protection with regard to their personal data, as they may be less aware of the risks, consequences and safeguards concerned and their rights in relation to the processing of personal data*" [5]. A vulnerability that leaks account existence on platforms used by minors could be interpreted as a failure to provide such specific protection. The vulnerability enables the unauthorised mapping of which child uses which educational service, potentially revealing

school affiliation. This sensitive data could be exploited for social engineering, bullying, or other malicious activities targeting a uniquely vulnerable demographic.

Similarly, the potential ability to confirm that an individual has an account on a mental health platform like peaasi.ee constitutes a significant privacy breach. Such information can imply an individual's interest in or need for mental health support, which is sensitive personal data. The exposure of this information could lead to stigma, discrimination, or targeted manipulation.

3.3 The List of Services to be Tested

The complete list of 55 services included in the research, categorised by what accounts on those sites are likely to indicate, is as follows:

- Financial capacity or risk tolerance: bet365.ee (betting), olybet.ee (betting), kv.ee (real estate), city24.ee (real estate), cvkeskus.ee (job seeking), hind.ee (price comparison), ResQ Club app (leftover food discounts), okidoki.ee (marketplace), kuldnebors.ee (marketplace), soov.ee (marketplace), paavlikaltsukas.ee (second-hand store)
- Goods and grocery preferences: kaup24.ee (general), euronics.ee (electronics), ikea.ee (furniture), jysk.ee (furniture), aeromotors.ee (car parts), k-rauta.ee (construction), zalando.ee (fashion), notino.ee (fashion), aboutyou.ee (fashion), rimi.ee (groceries), barbora.ee (groceries), Lidl Plus app (groceries), Maxima Estonia app (groceries)
- World view (news): postimees.ee, geenius.ee, delfi.ee, ohtuleht.ee, äripäev.ee (business)
- Entertainment preferences or hobbies: jupiter.err.ee (online streaming), play.tv3.ee (later rebranded to go3.tv, online streaming), piletilevi.ee (event tickets), apolloklubi.ee (SSO for cinema and restaurant group), Elisa Raamat app (e-books), nami-nami.ee (cooking recipes)
- Educational background: opiq.ee, moodle.edu.ee, harid.ee, kae.edu.ee, solnet.ee, ekool.eu, eliis.eu, ope.ee (via jpg.ope.ee)
- Sex life: iha.ee (adult content), amoremi.ee (dating), flirtic.ee (dating)
- Technical aptitude: upload.ee (file hosting), fv.ee (server hosting)

- Means of transport: elron.ee (train tickets), Circle K app (gas station)
- Information sources: inforegister.ee (business registry), kava.ee (TV schedules)
- Community belonging (Russian community): forum.ee, stena.ee
- Mental health: peaasi.ee

3.4 General Testing Procedure

To detect vulnerabilities that leak account existence, a systematic approach was applied to each authentication-related functionality (typically sign-in, password reset, account registration and sometimes email change) on each selected website and app. The core concept is that a vulnerable service reveals, through some observable difference in its responses, whether an identifier (e.g., email address) is associated with an existing account.

3.4.1 Systematic Testing Approach

An initial test account was created using an accessible email address (specifically created for this research to avoid using personal or third-party data without consent) to establish a baseline for an existing account. For testing the email change vector, a second test account with a different email was also created. The vulnerability detection process generally followed these steps across approximately 8–14 distinct interactions per service:

1. **Scenario of existing email:** Interact with the target functionality using the email address from the initially created test account (or the second account's email for email change tests). Record the request and the response.
2. **Scenario of non-existent email:** Interact with the same functionality using an unused email address. Record the request and the response.
3. **Stealthiness test (if applicable):** If the form involved multiple input fields (password, CAPTCHA, Terms of Service checkbox, etc.), additional tests were performed. These tests combined an existing or non-existing email with deliberately incorrect values for other fields (e.g., invalid password, unaccepted ToS, deliberately incorrect CAPTCHA if possible). Record the request and the response. These tests determined if the vulnerability could be exploited without triggering noisy server-side actions like sending confirmation emails.

The responses from steps 1 and 2 were compared (see Section 3.6). Any discernible, consistent difference indicated a vulnerability that leaks account existence. Step 3 established if such information disclosure could potentially occur *stealthily*, meaning without triggering expected notifications to the legitimate account owner.

3.4.2 Mobile Application Testing Specifics

While the general principles remained the same, testing mobile applications presented unique challenges. Deep network traffic analysis (e.g., using Wireshark with a MITM proxy) was not performed for mobile apps due to its technical complexity. Instead, testing relied primarily on manual user interface (UI) analysis, observing the app's direct responses within the UI after submitting the forms. This approach acknowledges that unindicated differences in API responses or other side-channels might have been missed. Apps were tested on both iOS and emulated Android platforms.

For purposes of statistical analyses in this thesis, all mobile apps were categorised as having CAPTCHA protection, as they were not explicitly tested for anti-automation measures.

3.5 Testing Variations

While the general procedure was mostly applicable, there were variations based on the specific functionality being tested.

3.5.1 Account Registration Form

The standard procedure (Steps 1, 2, 3) was followed. The stealthiness test (Step 3) was crucial, as successful account registration often sends confirmation emails. If email existence was checked before other validations (like password strength or ToS acceptance), an attacker could potentially disclose existence while ensuring the form submission failed, thus preventing email notifications.

The account registration forms were tested first, as the resulting test account was also necessary to test other forms.

3.5.2 Sign-In Form

Sign-in forms typically do not trigger a failure notification being sent to the email owner. Therefore, only steps 1 (existing email + wrong password) and 2 (non-existent email) were necessary. The responses from these two steps were compared.

Because the logic behind lockouts and bans was opaque and varied across services, and user-requested deactivations would have resulted in the closure of test accounts, the examination of

basic functionalities across a larger number of targets was prioritised over investing resources into analysing these more complex scenarios.

3.5.3 Password Reset Form

The general procedure was followed, including the stealthiness test (Step 3). Password reset forms often trigger sending an email, so the ability to check for account existence without sending an email (e.g., by failing a CAPTCHA or other field validation simultaneously) was investigated.

3.5.4 Email Change Form

The general procedure required a logged-in state and the second test account. The first account's email was tested by changing it to both a non-existent email (Step 2) and the second account's existing email (Step 1). The stealthiness test (Step 3) was crucial, as these forms almost always send notifications or confirmations to old and new addresses. It was checked if triggering other validation errors (e.g., incorrect current password) could disclose existence before submission.

3.5.5 Special Cases

During testing, several special cases were encountered:

- Some sites routed to sign-in or account registration based on initial email input, or had multi-step forms validating email early, making it trivial to check for account existence.
- Some sites lacked password reset or email change forms. Email change sometimes required customer support.
- Notably, `bet365.ee` had functionality blocking the use of Chrome DevTools. Safari's Web Inspector was used to bypass this block.

Handling Front-End Validation and CAPTCHA. Some services had front-end form validation via JavaScript. Similarly, Google's reCAPTCHA might have validated requests invisibly. No attempts were made to submit invalid data by circumventing client-side validation measures, which means more forms might have been stealthy than were observed.

3.6 Data Gathering and Analysis

Manual analysis of responses is error-prone and difficult to scale. To improve accuracy, efficiency, and record-keeping, the HTTP Archive (HAR) format was used as the primary method for data collection and analysis.

3.6.1 Data Recording with HAR Files

For each test interaction, a HAR file was recorded using Google Chrome's developer tools. The HAR file format provided a detailed JSON-formatted record of all HTTP requests and responses, including methods, URLs, headers, cookies, POST data, status codes, response content, and timings. This record enabled systematic comparison and later re-examination. The recording process was started immediately before interacting with a form and stopped just after, capturing only the relevant network traffic. The first request in the HAR file containing the tested email address was considered the primary request for analysis.

Alongside HAR files, evidence collection involved taking screenshots of relevant forms and maintaining qualitative notes on observed behaviours or testing difficulties for each service.

3.6.2 Vulnerability Detection

A vulnerability that leaks account existence (or *information disclosure*) was determined by comparing the primary requests in HAR files from the existing versus non-existent email scenarios. The vulnerability was confirmed if there was any discernible difference in the following:

- **HTTP Status Codes:** For example, a `200 OK` for a non-existent email versus a `409 Conflict` for an existing one.
- **Response Body Length:** Comparing the content length of the response bodies. Comparing length avoids false positives from dynamic content like CSRF tokens.
- **Redirection URL:** A difference in the `Location` header, indicating the user is sent to a different page based on the email's existence.

3.6.3 Stealthiness Detection

Stealthiness—the ability to exploit the vulnerability without notifying the user—was primarily relevant for account registration, password reset, and email change forms. HAR files from the stealthiness tests (Step 3, which used invalid non-email fields) were analysed using the same criteria (status, length, redirect). A discernible difference indicated the vulnerability could be exploited stealthily. Sign-in forms, if found to allow disclosure, were considered inherently stealthy as they typically do not trigger user notifications on failure.

3.6.4 Side-Channel Timing Analysis

While a large-scale statistical analysis was not performed, significant timing differences were assessed as a potential side-channel. The HAR file's `time` property, representing the total response time, was used for this assessment. A one-off difference of 500 ms or more between the 'existing' and 'non-existent' scenarios was considered a potential finding for a timing-based information disclosure. This threshold was chosen as an arbitrary but significant value, intended to distinguish a likely server-side processing difference from minor network jitter. Other minor side-channels, such as different page titles, were captured by the primary information disclosure detection methods.

3.6.5 Detection of Protective Measures

CAPTCHA Detection. Completely Automated Public Turing tests to tell Computers and Humans Apart (CAPTCHAs) were detected by searching the primary HAR request for case-insensitive keywords: `captcha`, `checksec` and `challenge`. The keyword search aimed to identify both visible and invisible server-validated CAPTCHAs based on keywords found during iterative testing.

Rate Limit Detection. Rate limiting was assessed on a per-service basis. A recorded request (e.g., a sign-in attempt or a password reset attempt with an invalid email) was replayed 1000 times over 10 seconds using a script. If the response status code changed from the original during the test, it indicated a rate limit had been triggered. The duration of any imposed block was not systematically tested.

3.6.6 Manual Review and Overrides

The results from the scripted HAR analysis were manually reviewed and, where necessary, overridden. Manual review was crucial for ensuring high confidence in the findings and for correcting issues that automated analysis could miss. Such issues included false negatives for stealthiness caused by separate pre-validation requests, or for CAPTCHA detection where non-standard implementations were used. The review also addressed inconsistencies from dynamic content (e.g., varying headers or CSRF tokens) and cases where different response content had an identical byte length.

3.7 Tooling and Automation Exploration

While the core testing was manual, several tools and scripts were developed or explored to support the research.

Simple helper scripts were developed for repetitive tasks. These included basic HAR parsing scripts to extract key data points for comparison, a script to automate the rate-limit testing by replaying HTTP requests, and a script using the Microsoft Graph API to manage the disclosure email campaign (see Section 5).

Recognising the limitations of manual testing at a larger scale, more advanced automation was also explored, although it was not used for the final data collection. Browser automation tools like Playwright and LLM-based agents such as Skyvern-AI were investigated. Initial tests showed these tools could potentially automate the testing procedure, but they proved inconsistent. Challenges included handling complex DOM structures, anti-bot measures, CAPTCHAs, and the high computational or financial cost of running advanced LLM agents at scale. For instance, using a cloud-based model via Skyvern-AI was estimated to cost tens of cents per test step, making a large-scale study prohibitively expensive. These difficulties led to the decision to rely on the manual, HAR-based methodology to ensure high confidence in the findings for this study.

3.8 Received Emails Verification

The mailbox of the email used for testing was manually reviewed to map received emails (password resets, account registration or email change confirmations, welcome messages) to actions taken. The verification of received emails helped confirm the stealthiness assessments. Delayed promotional emails were disregarded.

3.9 Ethical Considerations and Responsible Disclosure

Ethical conduct was paramount throughout the research. A responsible disclosure protocol was followed, which involved notifying affected service providers of the findings. The Estonian Data Protection Inspectorate (AKI) was consulted before sending the initial notifications. The detailed disclosure process and outcomes are described in Section 5.

Tests with third-party emails were not performed without consent; instead, a new email address was created specifically for the purposes of testing. The responsible disclosure protocol that was followed prioritised communication with service providers to allow adequate time for fixing before any form of public disclosure.

3.10 Methodological Limitations

Several limitations should be considered when interpreting the findings of this study.

Manual Scope. The target list was manually curated and limited to 55 services. While aiming for popular and relevant targets, this selection is not exhaustive, and findings may not generalise to all Estonian web services.

Potential Missed Side-Channels. The analysis focused primarily on explicit differences in status codes, response lengths, redirects, obvious content changes, and basic timing analysis. More subtle side-channels (e.g., complex timing variations requiring statistical analysis, non-HTTP information disclosures) were likely missed.

Anti-bot and Rate-Limiting Tests. The assessment of CAPTCHAs and rate limiting was basic. Sophisticated anti-bot systems that use fingerprinting, behavioral analysis, or probabilistic challenges may not have been fully triggered or accurately assessed. Similarly, complex rate-limiting schemes (e.g., adaptive or credential-based) may have gone undetected by the methods used, as comprehensive testing of these defences was beyond the scope of this work.

Mobile App Analysis Depth. Relying on UI analysis and email verification for mobile apps is less comprehensive than intercepting and analysing direct API traffic. Network-level information disclosures specific to app APIs might have been missed.

Focus on Email. While common, focusing primarily on email addresses as the identifier means vulnerabilities related to leaking account existence via phone numbers or usernames were only noted if encountered incidentally, not systematically tested across all targets.

Edge Case Coverage. While some edge cases were noted (unconfirmed accounts, multiple users per email), systematic testing of all possible account states (banned, locked, unactivated, etc.) and related policies (e.g., account lockout thresholds) was not performed.

Manual Analysis Bias. Despite efforts for systematic analysis using HAR files and clear criteria, manual review and overrides introduce potential for subjective interpretation or missed inconsistencies. While automation was explored, its challenges meant manual analysis remained central, carrying inherent risks of human error or bias, although deemed necessary for accuracy in this context.

4. Findings

This section presents the findings from the analysis of vulnerabilities that leak account existence across 55 popular Estonian online services, including 50 websites and 5 mobile apps. The detailed analysis results for all 55 tested services are presented in Appendix J.1. The initial round of tests was conducted between November 2024 and April 2025.

4.1 Overview of Vulnerability Landscape

The central finding of this study is stark: all 55 tested services were found to be vulnerable to leaking account existence through at least one vector. However, the level of risk was observed to vary significantly based on two key factors: the potential for an attack to be performed with stealth (i.e., without alerting the user) and the presence of anti-automation defences (e.g., CAPTCHA and rate limiting). A concerning majority, 31 (56%) of the 55 tested services, allowed for stealthy checks of account existence and lacked any CAPTCHA protection, posing the most significant risk for automated abuse and making them susceptible to the mass enumeration threat model (see Section 2.3.1). An additional 12 (22%) of the services were also vulnerable and without CAPTCHA, though not in a stealthy manner (see Figure 12).

A further 9 (16%) had CAPTCHA protection, but were still vulnerable to non-automated attacks. The final 3 (5%) services had combinations of different forms, with some vulnerable to mass enumeration (no CAPTCHA, not stealthy) and allowing for stealthy checks (has CAPTCHA, is stealthy).

A further breakdown by user interaction flow reveals that the vulnerability is not evenly distributed across functionalities (see Figure 13). Account registration and email change forms were found to be the most common vulnerable forms; nearly all services that offered these functionalities were vulnerable, often in a way that could be exploited with stealth and without triggering anti-bot measures. In contrast, sign-in forms were the most robustly implemented, with the vast majority, 44 (86%) of 51 tested sign-in forms, found to be secure against leaking account existence. Password reset forms represented a middle ground, being frequently vulnerable, 33 (66%) of 50 tested password reset forms, but less often in a manner that could be exploited with stealth, only 3 (6%) forms.

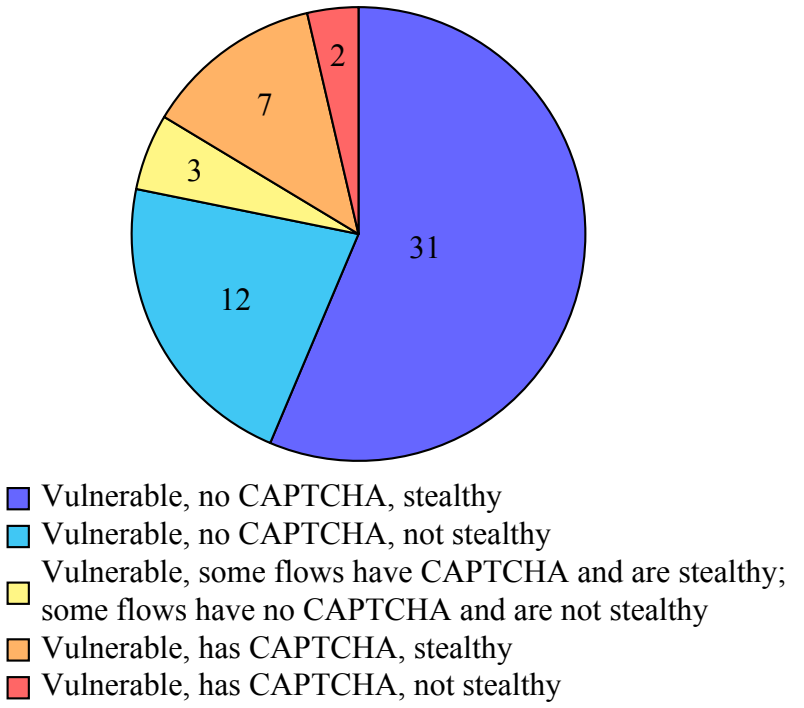

Figure 12. Severity profile of vulnerable services (55 total tested)

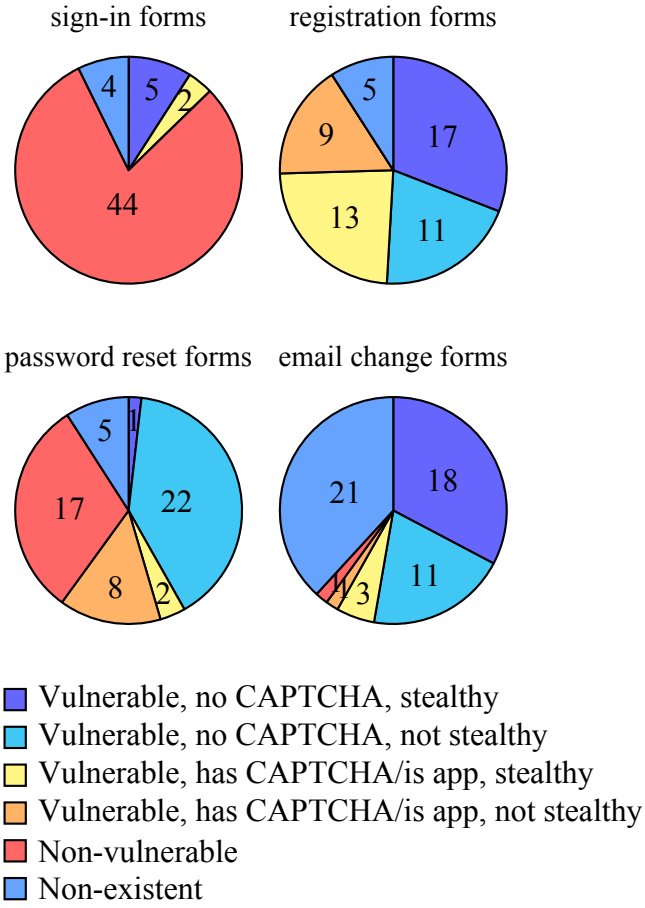

Figure 13. Vulnerability severity across authentication flows

4.2 Findings by Vulnerability Vector

The 55 tested services provided 185 distinct sign-in, password reset, registration or email change forms to be tested, of which 123 (66%) were found to be vulnerable to leaking account existence. Vulnerabilities were identified across common user authentication and management flows.

Information disclosure typically occurred through distinguishable responses, with differences in response content being far more common than obvious timing side-channels. A response content difference was observed in 123 (66%) of 185 tested flows, while a significant timing difference of at least 500ms was found in only 29 (16%) of them (see Figure 14).

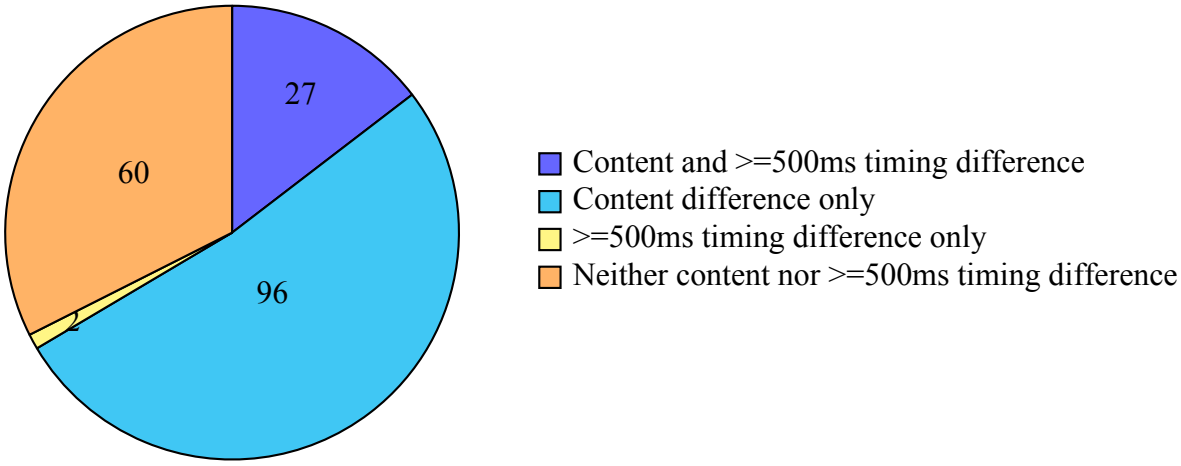

Figure 14. Distribution of response content difference versus response timing difference across all tested flows

Notably, the most frequent type of information disclosure was an explicit error message displayed to the user, which was observed in a majority of the vulnerable flows (see Figure 15). This pattern suggests that many vulnerabilities are not accidental side-channels but rather the result of intentional design choices that prioritise a specific notion of user experience over the principle of data protection by design. Detailed evidence for the claims made in this section is available in Appendix J.1.

4.2.1 Sign-in Forms

Sign-in forms accepting an email address as the user identifier were available on 51 (93%) of the 55 tested services. The remaining services relied on other identifiers such as usernames but still allowed account existence to be checked via email on other forms.

Direct information disclosure confirming account existence during the sign-in process was less common, observed on 7 (14%) of 51 services with a tested sign-in form (cvkeskus.ee,

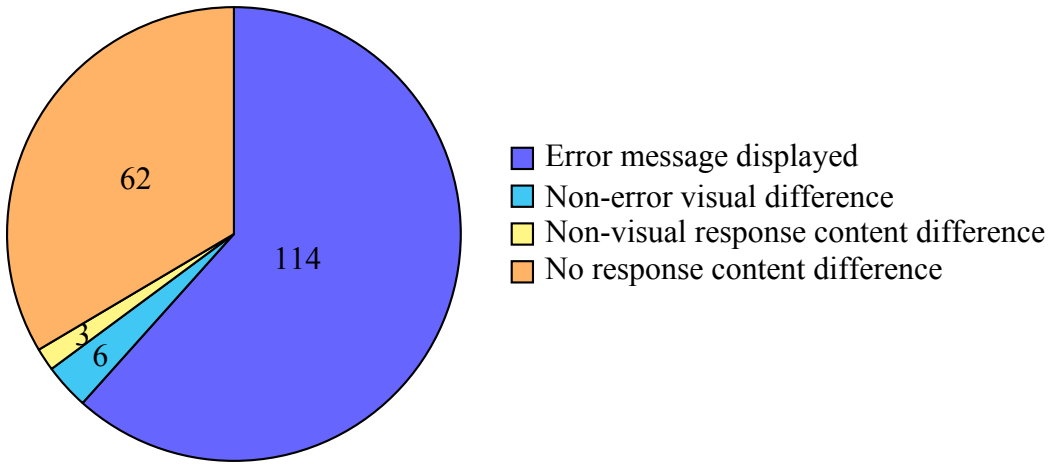

Figure 15. Distribution of response content differences across all vulnerable flows

city24.ee, geenius.ee, jpg.ope.ee, k-rauta.ee, Circle K app, LIDL Plus app). This direct disclosure through sign-in forms is often inherently stealthy, as failed sign-in attempts that reveal existence information were generally observed not to trigger security alerts or notifications to the account holder.

For context, three services (kava.ee, stena.ee and upload.ee) had sign-in form relying on usernames, however email uniqueness or existence was still checked on other forms. Additionally, zalando.ee had a shared first step of the sign-in and registration forms, so it was not counted in the form statistics – it was vulnerable.

4.2.2 Password Reset Forms

Password reset functionality was available on 50 (91%) of the 55 tested services. The password reset forms were also frequently vulnerable, on 33 (66%) of the 50 services with the form available.

Stealthy checks for account existence were possible in only 3 (9%) of the 33 vulnerable flows. This could occur when the form asked for more information than just the email address and validated user's existence first. No CAPTCHA protection was present on 23 (70%) of the 33 vulnerable flows, increasing the risk of automated attacks.

4.2.3 Account Registration Forms

An explicit account registration form was available on 50 (91%) of the 55 tested services. The service harid.ee did not have an explicit registration form, as an account was created automatically upon first sign-in with a national eID. Also, zalando.ee was also excluded from this specific count with its combined sign-in and registration form. Account registration

represented the most frequent vulnerability vector, affecting all 50 (100%) of the 50 services with a registration form.

No CAPTCHA protection was present on 28 (56%) of the 50 vulnerable flows.

Stealthiness was high in this vector, observed in 30 (60%) of the 50 vulnerable account registration flows. On 27 (54%) of the vulnerable forms (e.g., kuldnebors.ee, stena.ee, Elisa Raamat app), the existence check of the email address was performed before other validation steps, such as password complexity checks or terms of service acceptance. Amongst those, 8 services (barbora.ee, bet365.ee, euronics.ee, aeromotors.ee, forum.ee, jupiter.err.ee, paavlikaltsukas.ee, zalando.ee) facilitated easier discovery of account existence through dedicated pre-submission API checks for identifier availability.

On 8 (16%) vulnerable forms (aboutyou.ee, aripaev.ee, bet365.ee, hind.ee, jupiter.err.ee, kae.edu.ee, kava.ee, paavlikaltsukas.ee), stealthy discovery of account existence was enabled by the lack of an email notification or confirmation step after registration.

4.2.4 Email Change Forms

The functionality to change the email address was available on 34 (62%) of the 55 tested services. The email change form was found to be a universally weak vector, as it revealed account existence on 33 (97%) of the 34 services where the form was available. Only Maxima Estonia app was found to be secure against leaking account existence on the email change form.

Stealthiness was observed in 21 (64%) of the 33 vulnerable email change flows from two primary implementation flaws. On 15 (45%) of the vulnerable forms (e.g., delfi.ee, kaup24.ee, amoremi.ee), the existence check on the new email address was performed before other validation steps, such as current password confirmation. On a partially overlapping set of 13 (39%) vulnerable forms (e.g., aboutyou.ee, opiq.ee, stena.ee), stealthy discovery of account existence was enabled by the lack of an email notification or confirmation step after a successful change. Curiously, 29 (88%) of the 33 vulnerable email change flows had no CAPTCHA protection.

4.3 Side-Channel Vulnerability Findings

Leaking account existence was also possible through indirect information disclosure, known as side-channels. The findings in this subsection are based solely on the analysis of HAR files from

the websites. Due to technical limitations, a similar analysis was not performed on the mobile apps.

Timing Attacks. While the analysis was based on a single test for each scenario and thus is not statistically conclusive, the HAR analysis indicated that timing differences could potentially be used as a side-channel for discovering account existence. Although most tested forms had quick responses, a notable number exhibited significant response time differences between tests with existing and non-existing accounts. Of 167 website flows measured, 29 (17%) had an observed time difference of 500ms or more (see Figure 16). In some cases, these one-off differences were extreme; for example, a test on apolloklubi.ee revealed a response time difference of approximately 10 seconds. The presence of these substantial, albeit single-run, timing discrepancies across 19 services suggests that a more dedicated statistical analysis could likely confirm an exploitable side-channel on at least some of them (see Appendix J.1).

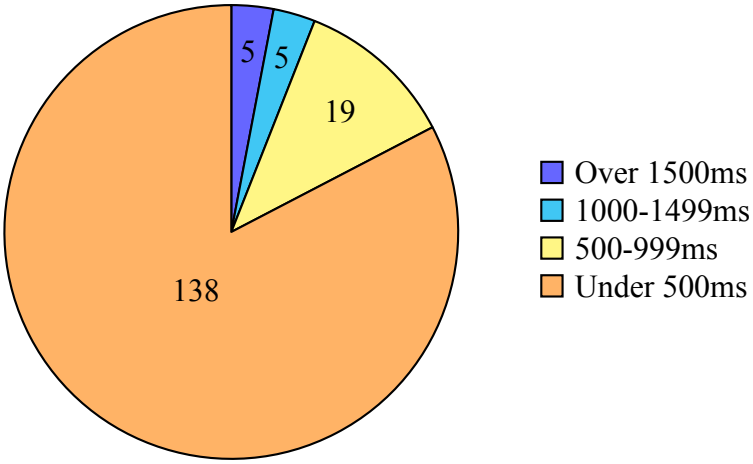

Figure 16. Distribution of timing differences across website forms tested

Response Content Differences. While all visual differences indicate differences in server responses, the opposite is not true. Subtle variations in response content were the only way account existence could be discovered on k-rauta.ee sign-in form, with minor HTML structural changes (see Figure 17), soov.ee password reset form, with differing page titles (see Figure 18), and city24.ee sign-in form, with its API exposing more than shown to user (see Figure 19).

4.4 Platform Patterns

Piano.io. Three major Estonian news portals (delfi.ee, ohtuleht.ee and postimees.ee) were observed to use the *Piano.io* platform for user account and

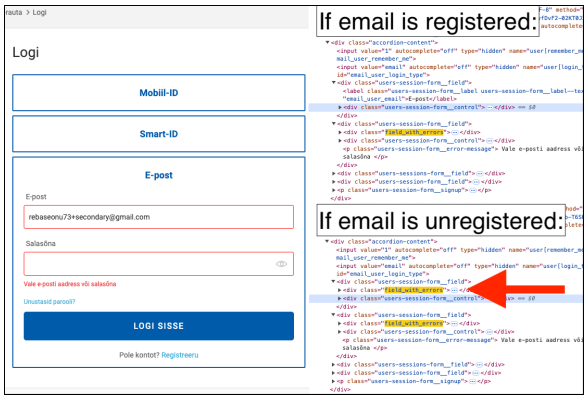

Figure 17. Response content difference on sign-in form of k-rauta.ee

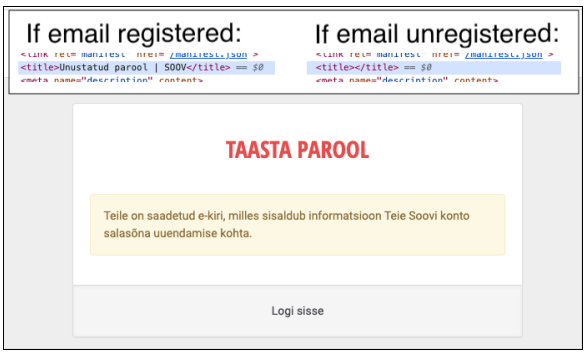

Figure 18. Response content difference on reset form of Soov.ee

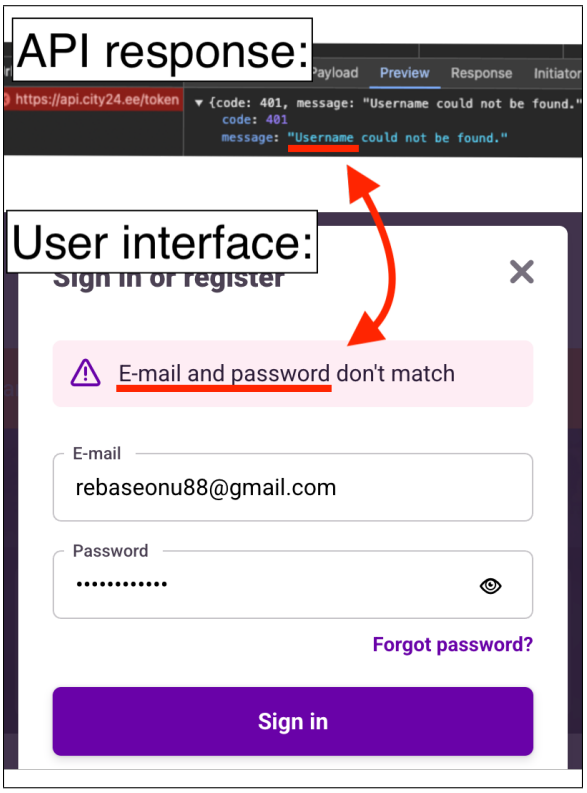

Figure 19. Response content difference on sign-in form of city24.ee

subscription management. This shared backend likely accounts for their nearly identical vulnerability profiles (see Appendix J.1). Furthermore, these portals all featured a subscription sharing functionality that introduced an additional information disclosure vector. This feature revealed not only whether a given email address was registered but also the full name associated with the account and, in some cases, whether the user had an active subscription (see Figures 20, 21 and 22). This finding highlights a supply-chain risk, where a vulnerability in a single third-party platform leads to a cascading effect across multiple downstream services.

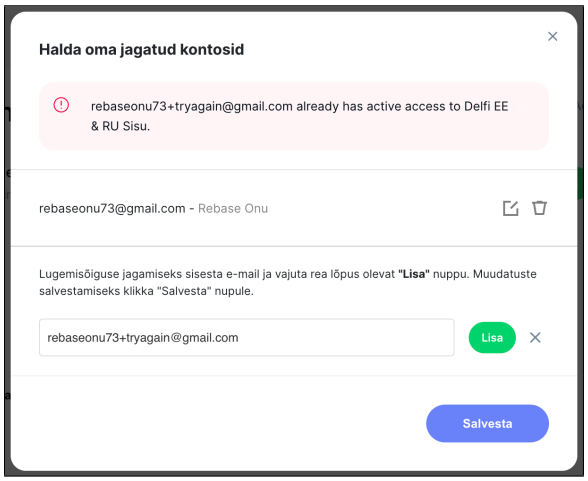

Figure 20. Subscription sharing form on delfi.ee, showing an error for existing subscription and full name for an existing account

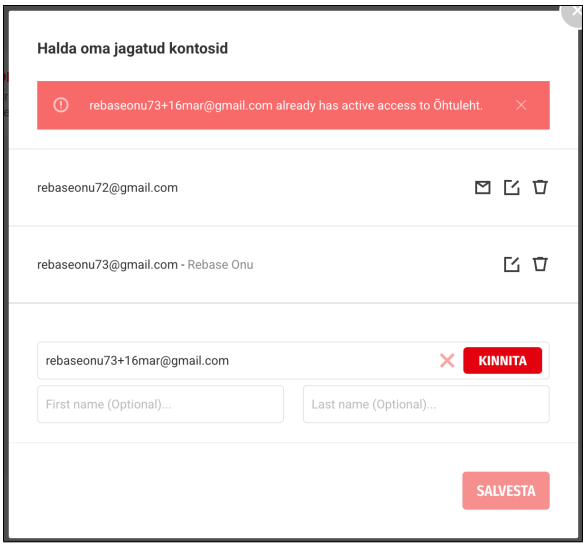

Figure 21. Subscription sharing form on ohtuleht.ee, showing an error for existing subscription, just the email for a non-existent account and full name for an existing account

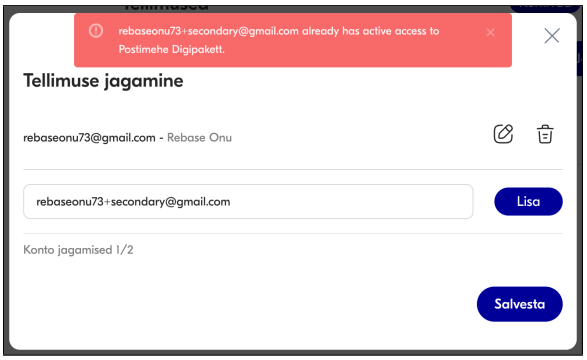

Figure 22. Subscription sharing form on postimees.ee, showing an error for existing subscription and full name of an existing account

Stuudium. A similar pattern of shared vulnerabilities was found across several school portals in Tartu (jpg.ope.ee, petersoni.ope.ee, tamme.ope.ee and tartukristlik.ope.ee), all of which use the *Stuudium* educational platform. While only jpg.ope.ee was formally included in the tested set of services, the others are shown for reference as they exhibited identical vulnerabilities due to the shared platform. On these sites, leaking account existence was possible through the sign-in form, which displayed a distinct error message if an account did not exist (see Figures 23, 24, 25 and 26). Notably, the vulnerability was particularly severe as the existence of an account could be discovered using not only an email address but also a user's full name or even their personal identification code. The presence of

such a vulnerability on an educational platform represents a significant supply-chain risk and is especially concerning from a data protection perspective due to it being targeted at children (see Section 3.2).

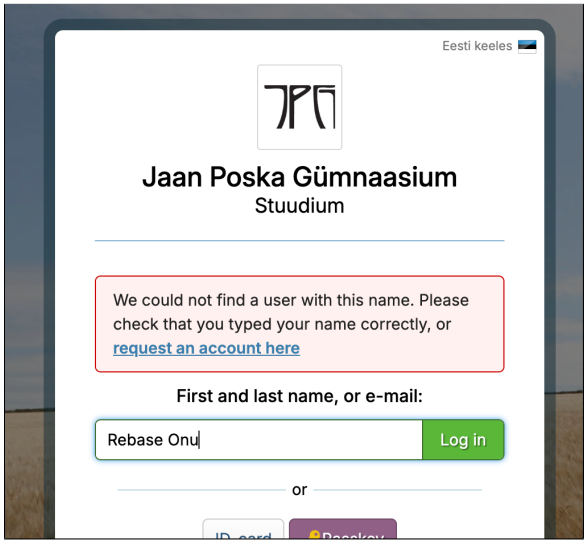

Figure 23. Sign-in form of jpg.ope.ee showing an error message about account non-existence

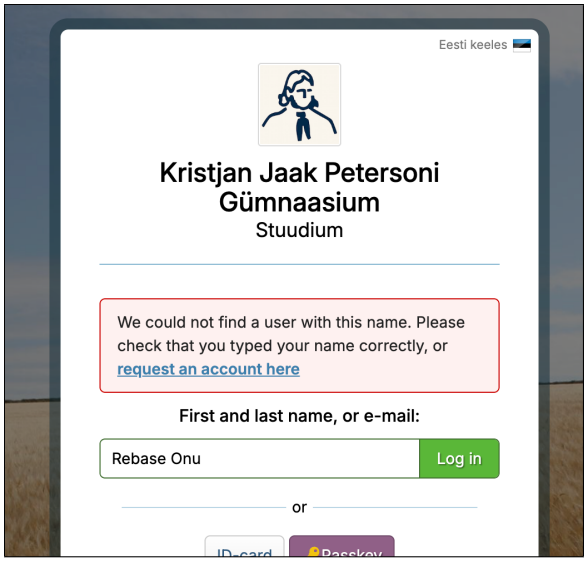

Figure 24. Sign-in form of petersoni.ope.ee showing an error message about account non-existence

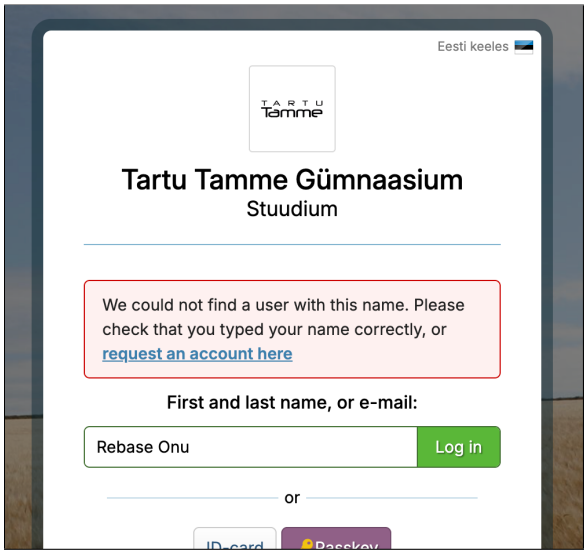

Figure 25. Sign-in form of tamme.ope.ee showing an error message about account non-existence

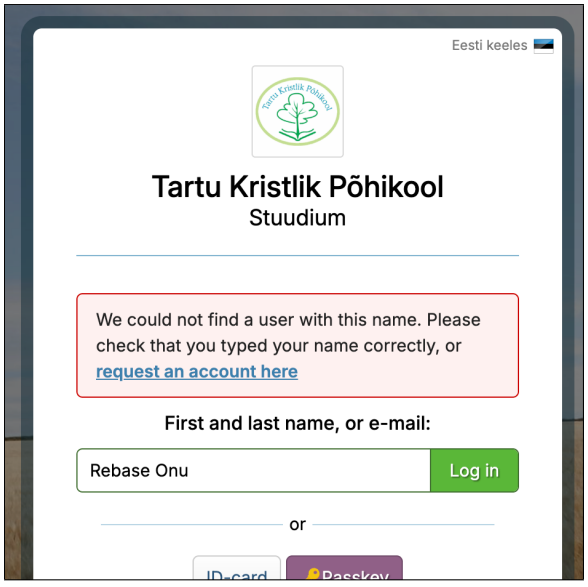

Figure 26. Sign-in form of tartukristlik.ope.ee showing an error message about account non-existence

4.5 Anti-Bot Measures

Defences against automated attacks were often weak or entirely absent across the tested services. CAPTCHAs were frequently missing from vulnerable forms, and where they were present, their implementation was sometimes flawed. For example, the CAPTCHA check on okidoki.ee

was observed not to block account enumeration because the check was performed after the server had already revealed the account's existence.

Rate limiting was similarly ineffective in many cases, failing to prevent rapid, scripted requests. Even the presence of a web application firewall like *Cloudflare*, used by 22 (44%) of the 50 tested websites, does not guarantee protection, as its effectiveness depends on proper configuration.

Advanced defences were rare, with isolated examples of TLS fingerprinting on `osta.ee` and anti-devtools techniques on `bet365.ee`, both of which were found to be bypassable.

5. Vulnerability Disclosure

The responsible disclosure process is detailed in this section, from the initial notification of service providers to subsequent reassessment and escalation. A multi-phased strategy was used to report vulnerabilities, encourage fixing, and systematically document the outcomes.

A log of all correspondence is available in Appendix K. A full archive of the correspondence with service providers is also available for reference at https://gregoreesmaa.github.io/account-enumeration/correspondence/.

All correspondence labelled the vulnerability as *Account Enumeration*. In retrospect, this specific phrasing may have biased service providers to focus primarily on the threat of mass-enumeration, even though the risk of targeted attacks was also explained. For clarity and precision in this thesis, the vulnerability is instead referred to by the more comprehensive term *Account Existence Leaks* to better encompass its full scope (see Section 2.3). The collected data, however, reflects the original, potentially misconstrued terminology.

5.1 Preliminary AKI Interaction

Prior to initiating contact with service providers, the Estonian Data Protection Inspectorate (AKI) was consulted to ensure the research methodology and disclosure plan were sound and to obtain a legal position on the matter. After an initial email exchange, a meeting was held with representatives from AKI, including their legal and technology departments.

AKI officials indicated that leaking account existence is indeed a data protection issue that data controllers would have to address. A key clarification was made regarding the term "*personal data breach*": while the potential for an account existence leak is a regulatory violation, it only becomes a formal "*personal data breach*" requiring notification if there is evidence of actual exploitation.

It was indicated by AKI that supervision proceedings against non-compliant service providers could be initiated based on a formal complaint or on their own initiative, with the response depending on the specifics of the case.

Crucially, it was also confirmed by AKI's representatives that simply updating a privacy policy is not a sufficient fix for a technical vulnerability, especially as such changes may not apply retroactively to users who have not accepted them. The legal standing of the research and the validity of the planned escalation path were affirmed by the consultation.

5.2 Phase 1: Initial Vulnerability Reporting

In the first phase, each vulnerable service provider was informed of the vulnerabilities that leak account existence identified in their systems (see Section 4).

5.2.1 Preparation of Disclosure Materials

To ensure clarity and consistency, a standardised initial vulnerability report was a PDF file rendered from a LaTeX template (see Appendix E for an example). Each report was personalised for the target service and included a comprehensive overview of the vulnerability, its implications, and recommended fixes:

- A clear introduction to the vulnerability of leaking account existence.
- A statement that the service was tested and found to be vulnerable, including the date of the test.
- An explanation of why the vulnerability constitutes a potential data breach under the General Data Protection Regulation (GDPR).
- A 30-day deadline for fixing, after which the service would be reassessed.
- Subsections addressing each identified vulnerable form (e.g., sign-in form, password reset), containing:
  - A screenshot illustrating the vulnerability.
  - An explanation of how the information disclosure manifests: as an explicit error message, a visual difference, or a subtle difference in the server response.
  - A conditional warning about missing anti-bot measures if no detectable CAPTCHA was present.
  - A warning about side-channels, such as timing differences, that could be used to discover account existence.
  - A conditional warning about the possibility of stealthy discovery of account existence, whether due to poor validation order or the absence of a confirmation email.
  - Guidance for fixing based on OWASP best practices [9].
- A summary of how security contacts were identified for the disclosure.

The report was written in English in an effort to motivate responses in English, so that they could be provided as an appendix to this thesis without being translated. The report was attached to a concise cover email that introduced the research and requested a confirmation of receipt (see Appendix D).

5.2.2 Contact and Notification

To lend credibility to the academic nature of the research, the initial vulnerability reports were sent from author's University of Tartu email address `gregor.eesmaa@ut.ee`. Microsoft Graph API was used to automate sending the emails, to ensure consistency and to avoid human error.

The process for identifying contacts was systematic, prioritising official security contacts where available. The hierarchy for contact discovery was:

1. The `security.txt` file [27]
2. The privacy policy accepted during registration
3. General contact or help pages
4. Internet domain registration data (from the internet.ee WHOIS directory)
5. The relevant business registry

If a confirmation of receipt was not received within seven days, the next contact in the hierarchy was sought, and the report was recompiled to include information about the previous contact attempt. A 30-day period since the first report was provided for service providers to address the reported issues, after which each service was systematically reassessed to determine if the vulnerabilities had been fixed (see Appendix J.2). These reassessments were conducted between May and June 2025.

The infrastructure for responsible reporting was found to be underdeveloped. The use of a `security.txt` file was minimal, found for only 6 (11%) of the 55 tested services. Even when present, issues such as expired contact information (`moodle.edu.ee`, `rimi.ee`, `Elisa Raamat`) or non-functional email addresses (`Lidl Plus app`) hindered standardised vulnerability reporting.

Additionally, privacy policies of services were stored for later comparison.

5.2.3 Responses to Initial Vulnerability Reports

Of all 55 of the service providers that were contacted, 21 (38%) provided a substantive response. Other service providers sent non-substantive manual responses, automated responses, or provided no response at all (see Figure 27).

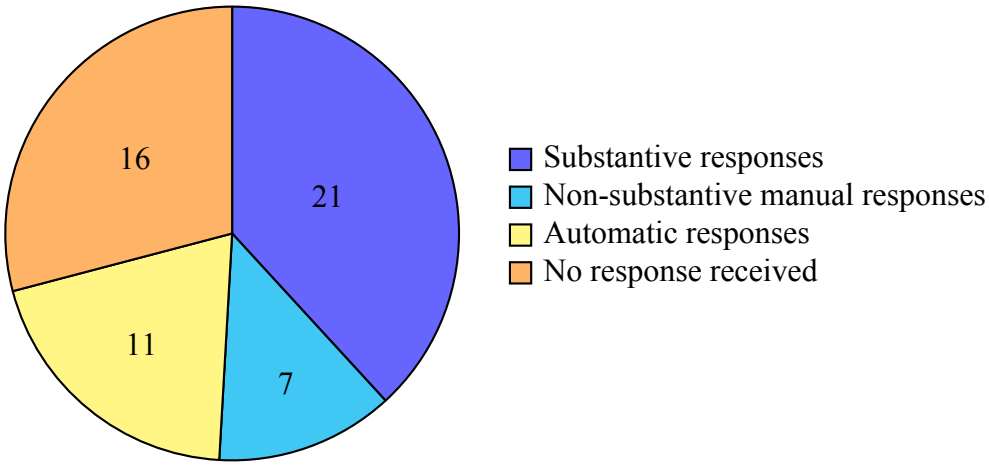

Figure 27. Types of responses received following the initial vulnerability report

Curiously, even though the report was written in English, 14 (50%) of the 28 manual responders wrote (at least partially) in Estonian (see Figure 28). This may indicate that the author's name and email address were recognised as Estonian, or that service providers preferred to respond in their native language. The latter potentially signals that English was a barrier to communication, and implies the lack of an Estonian translation might have skewed the outcomes.

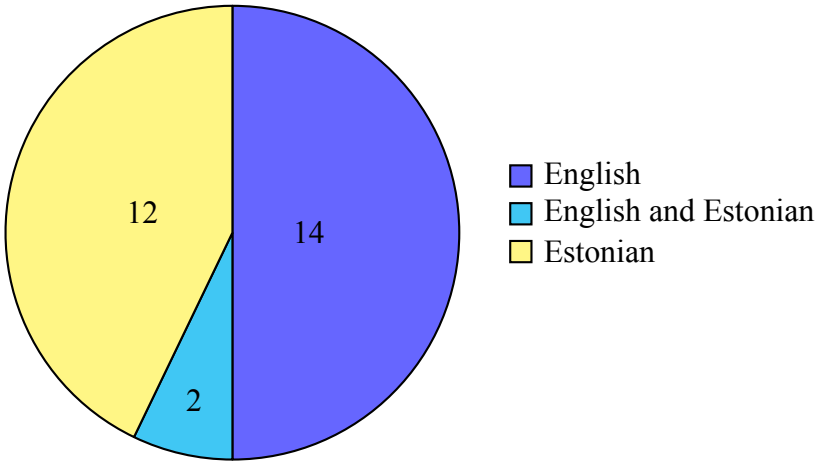

Figure 28. Languages of manual responses to the initial vulnerability report

Of the 55 services tested, 5 (9%) fixed the vulnerability after the initial vulnerability report (eliis.eu, inforegister.ee, kuldnebors.ee, opiq.ee,

S soov.ee). An additional 21 (38%) implemented partial fixes that did not completely solve the underlying problem.

These insufficient fixes usually stemmed from a misunderstanding of the vulnerability's root cause. One common mistake involved making only cosmetic changes; providers would alter the user interface but leave behind distinguishable API responses that could still be exploited (see Figures 29–32).

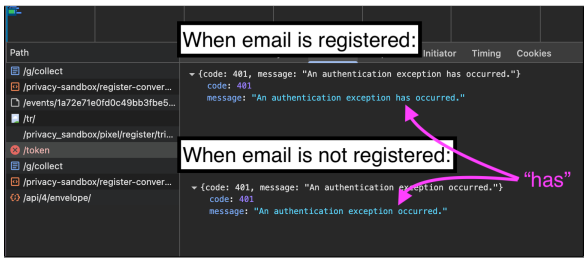

Figure 29. API response differences on the login form of city24.ee after an attempted fix

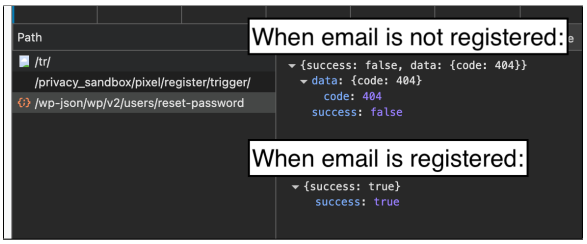

Figure 30. API response differences on the reset form of geenius.ee after an attempted fix

Figure 31. API response differences on the reset form of olybet.ee after an attempted fix

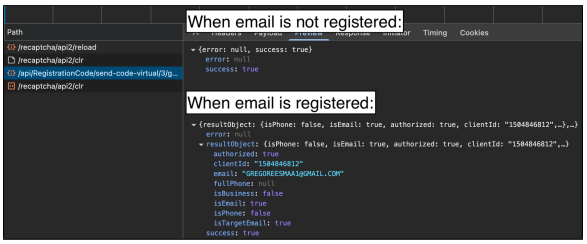

Figure 32. API response differences on the register form of rimi.ee after an attempted fix

Another frequent mistake was an asymmetric implementation, where the recommended generic message was applied only to failure cases. This meant success cases still returned a unique response, failing to stop the information leak (see Figures 33 and 34).

A third category of failed fixes involved using a generic but still distinguishable errors, like "*something went wrong*". This approach didn't mitigate the vulnerable vector because it still created an observable difference from a success case (see Figures 35 and 36).

The fact that 26 (47%) of the tested services took action, indicates that responsible disclosure can be effective.

5.3 Phase 2: Reassessment Follow-Up

For 10 services that had provided a substantive response to the initial report but had not fully eliminated the information disclosure, a reassessment report was sent. This report was similar to

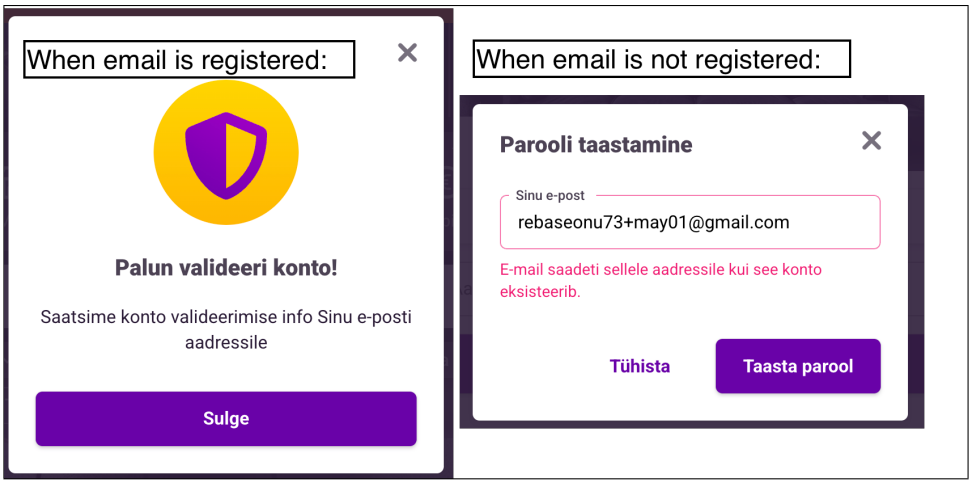

Figure 33. Messaging difference on the reset form of city24.ee after an attempted fix (left: "*We sent account validation info to your email address*", right: "*Email sent to this address if this account exists.*")

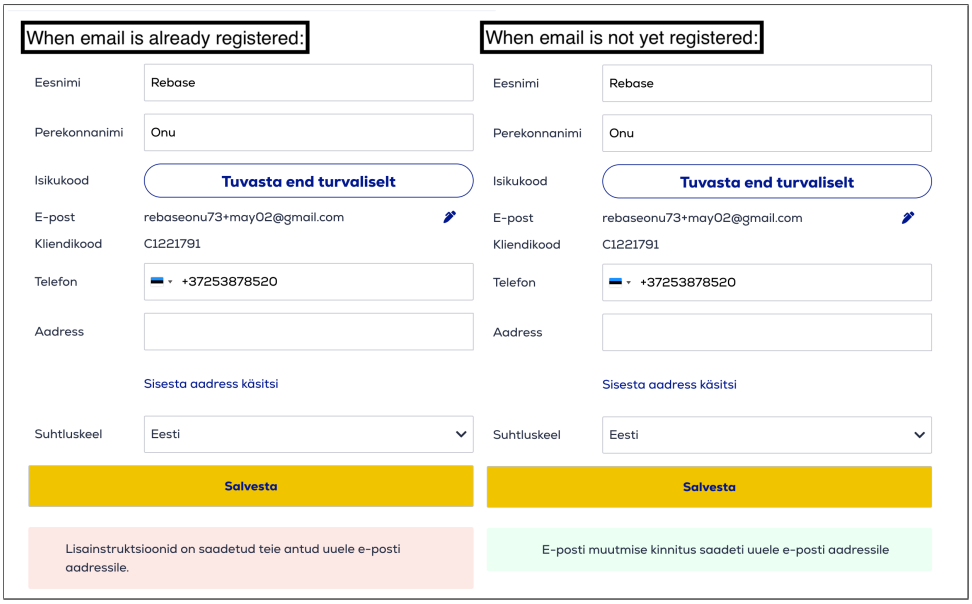

Figure 34. Messaging difference on the email change form of euronics.ee after an attempted fix (left: "*Additional instructions are sent to your new email address*", right: "*Email change confirmation sent to new email address*")

the initial one but emphasised the more subtle ways the forms remained vulnerable, which were overlooked in the initial attempts at fixing (see Appendix F and Appendix G).

After 30 days, a second reassessment was conducted to determine if the vulnerabilities had been addressed (see Appendix J.3). This round of reassessment was conducted in June 2025.

Of the 10 recipients, 3 (30%) fixed the vulnerability completely (amoremi.ee, city24.ee, apolloklubi.ee). Partial fixes were implemented by 3 (30%) additional

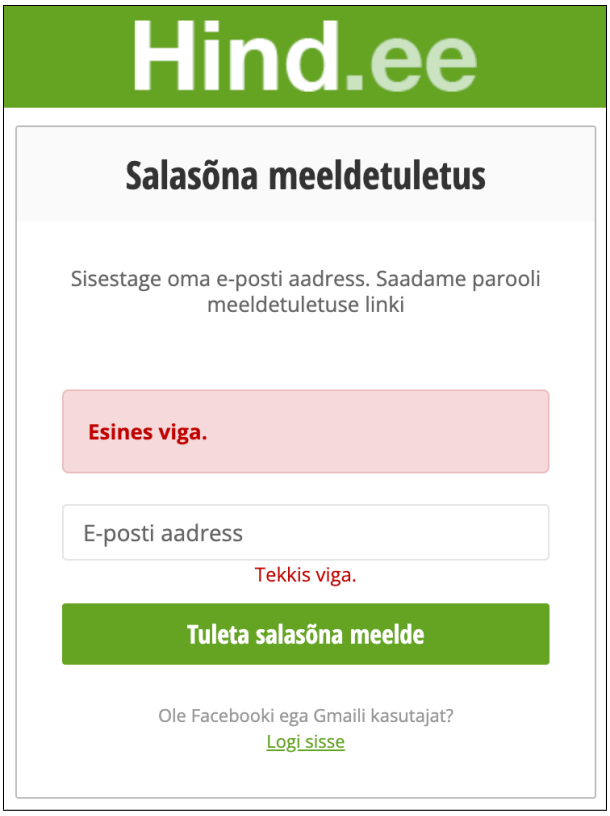

Figure 35. Messaging difference on reset form of `hind.ee` after an attempted fix ("*An error occurred*")

services. This indicates that while some services are willing to engage, the fixing process can be slow and incomplete.

Notably, `upload.ee` did not receive a standard reassessment report due to its rapid response. Instead, further feedback was provided in a free-form email, but the vulnerability was not yet fixed at this stage.

5.4 Phase 3: Escalation as a Data Subject

For service providers that did not respond or failed to fix the vulnerabilities, a formal escalation process was initiated by sending a GDPR request, with the author acting in the capacity of a data subject.

This request was composed in both Estonian and English to ensure it was understood regardless of the provider's operational language (see Appendix H and Appendix I). The request was built as a LaTeX template so it could contain the service name and state the vulnerable forms. The request invoked specific rights:

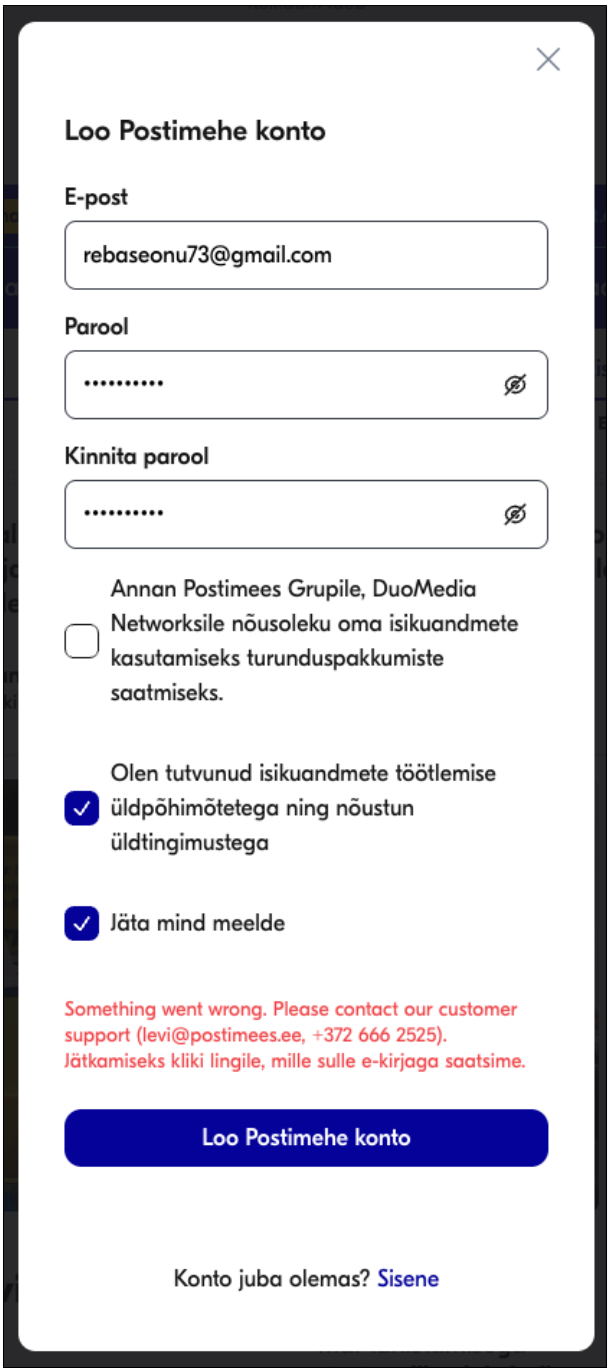

Figure 36. Messaging difference on register form of `postimees.ee` after an attempted fix

- Article 15 (Right of Access): To demand the legal basis for processing personal data in a manner that enabled account existence to be leaked.
- Article 18 (Right to Restriction of Processing): To demand a halt to the unlawful processing that revealed account existence.
- Article 21 (Right to Object): To formally object to the data processing, particularly where it might be justified under "*legitimate interests*".

This step shifted the communication from voluntarily responding to a vulnerability report to a legal obligation to respond.

The requests for each service were digitally signed one-by-one systematically and carefully, in an attempt to reduce the risk of human error. The latest reassessment report was included for reference, as it provided detailed overview and mitigation recommendations.

To avoid ambiguity, these requests were sent from author's personal email address, and accounts were created with this address where none previously existed to ensure clear standing as a data subject. Gmail API was used to automate sending the requests, ensuring consistency and reducing the risk of human error.

5.4.1 Responses to GDPR Requests

Of the 47 service providers that received a GDPR request, 34 (72%) provided a substantive response. This response rate is about double the rate for the initial vulnerability report, which is expected given the more probing nature of the GDPR request as compared to the initial vulnerability report.

A significant drop in the use of English was observed in these responses; only 14 (36%) of the 39 manual responders wrote in English (see Figure 37). This is likely because the GDPR request was sent in both Estonian and English, with the Estonian version appearing first, and GDPR requires responding in a clear and plain language, which providers may interpret as using the data subject's native language.

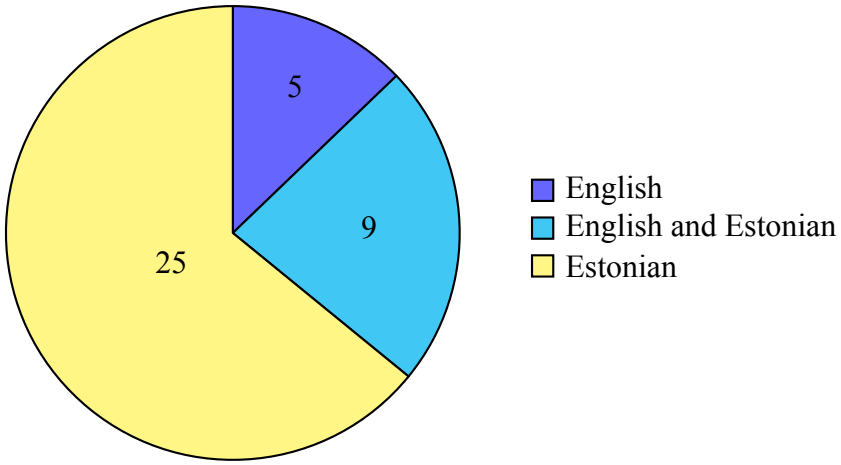

Figure 37. Languages of manual responses to the GDPR request

Most responses were received in last days before the statutory one-month deadline. Only Maxima Estonia app and cvkeskus.ee formally requested an extension, citing complexity of the request[3].

A concerning 13 (28%) of the providers failed to provide a substantive response to the GDPR requests. Of those, 5 (38%) services manually confirmed receipt, but missed the statutory one-month deadline to provide a substantive response, and as of submission of this thesis have not provided it (elron.ee, go3.tv, moodle.edu.ee, peaasi.ee, bet365.ee). The other 8 (62%) provided no response at all (see Figure 38).

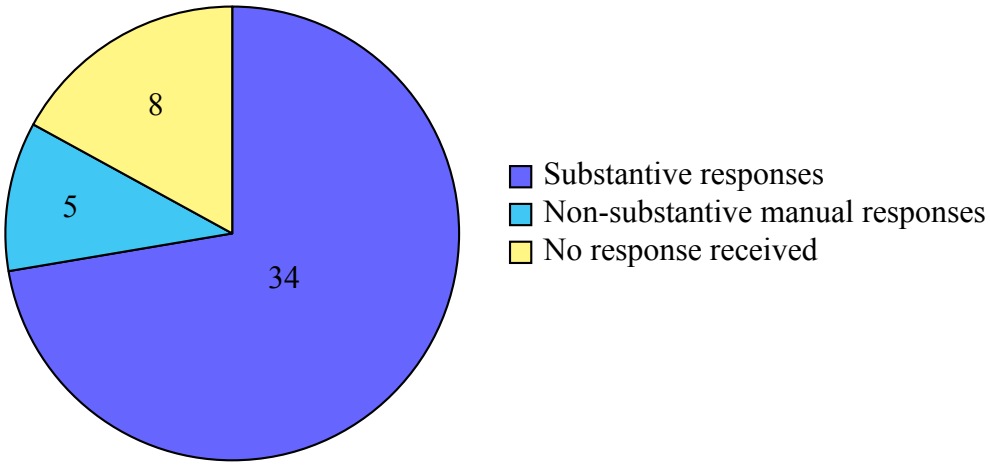

Figure 38. Types of responses received from service providers following the GDPR request

The legal bases cited for the data processing were varied (see Figure 39). Of the 47 providers that received the GDPR request, 12 (26%) relied on legitimate interest necessity (GDPR Article 6(1)(f)), while 8 (17%) cited contractual necessity (GDPR Article 6(1)(b)). Only olybet.ee and Maxima Estonia app explicitly acknowledged having no legal basis for the processing before confirming the fix.

Of the 47 services that received a GDPR request, 7 (15%) subsequently fixed the vulnerability (harid.ee, upload.ee, ikea.ee, nami-nami.ee, k-rauta.ee, olybet.ee, Maxima Estonia app; see Figure 40). A partial fix was attempted by 5 (11%) services (okidoki.ee, jpg.ope.ee, fv.ee, euronics.ee, ResQ Club app).

[3] cvkeskus.ee asked for a 14-day extension, but missed that and responded on day 23

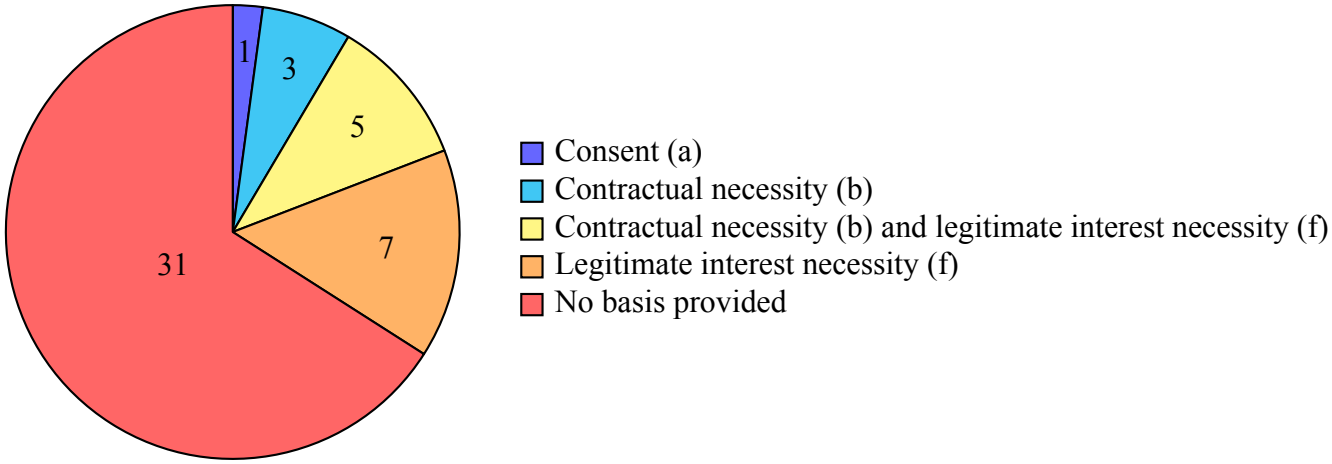

Figure 39. Distribution of legal bases cited by service providers for processing personal data in a manner that enabled account existence to be leaked

Additionally, 13 (28%) of the services promised to fix the vulnerability, with some providing specific timelines (e.g., jysk.ee, okidoki.ee). However, 14 (30%) refused to fully fix the vulnerability (hind.ee, aboutyou.ee, kv.ee, aripaev.ee, delfi.ee, aeromotors.ee, postimees.ee, flirtic.ee, zalando.ee, jpg.ope.ee, ohtuleht.ee, forum.ee, rimi.ee, LIDL Plus). The final 13 (28%) provided no details on remediation plans or lack thereof (barbora.ee, bet365.ee, elron.ee, go3.tv, iha.ee, jupiter.err.ee, kava.ee, moodle.edu.ee, notino.ee, paavlikaltsukas.ee, peaasi.ee, solnet.ee, stena.ee).

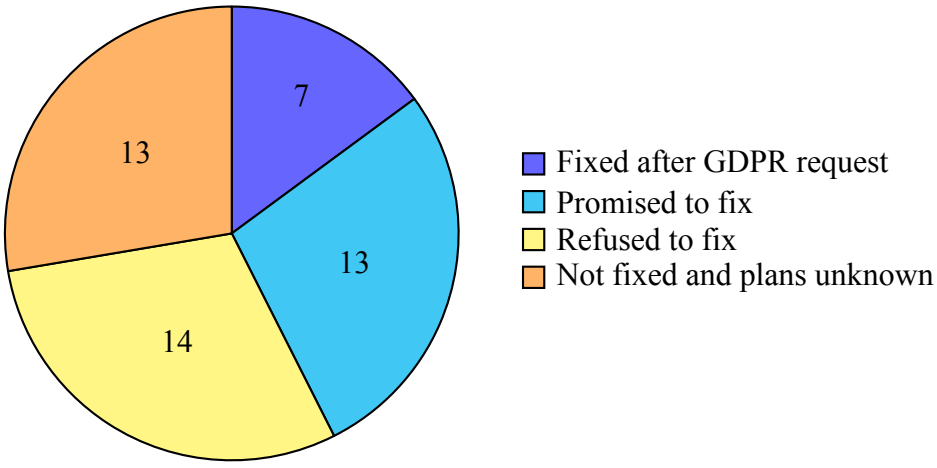

Figure 40. Distribution of actions in response to the GDPR request

5.4.2 Justifications and Arguments Against Fixing

Four recurring justifications were most often cited for not fixing the vulnerability: an appeal to common practice, appeal to ensuring security, user experience (UX) concerns, and necessity for performing a contract. Additionally, several providers tried to downplay the risks or deflect accountability altogether.

Curiously, many services that promised to fix the issue, also provided justifications. It is possible they did not want to be held liable for the vulnerability having been present in the first place. However, no legal justifications were found to be convincing, as none proved freely given consent or necessity for leaky mechanisms (see Section 2.4)).

Appeal to Common Practice. Claims that the practice is widespread or that specific larger services have similar vulnerabilities were made by 14 (41%) of the 34 services that provided substantive responses (e.g., upload.ee, flirtic.ee, jysk.ee, Lidl Plus app). It was argued that since industry leaders like Google, Microsoft or Facebook have similar information disclosures, the practice constitutes an acceptable standard. For example, it was explicitly stated by upload.ee, "*I would assume that Facebook and Microsoft are quite up to date about how exactly GDPR applies*". This argument attempts to normalise the vulnerability by citing its prevalence, rather than addressing its inherent risks and legal implications under GDPR.

Appeal to Ensuring Security. Appeals to ensuring security were made by 10 (29%) of the 34 services that provided substantive responses. Mostly, such justifications were vague, citing generic security reasons (e.g., "*preventing abuse*" by jysk.ee, or "*ensuring security*" by kaup24.ee). Some providers provided a more specific justification of preventing duplicate account creation (e.g., delfi.ee, ekool.eu, Elisa Raamat app). Often, legitimate interest necessity was cited (GDPR Article 6(1)(f)) as the legal basis in conjunction with security reasons (e.g., jysk.ee, rimi.ee, kaup24.ee).

The argument that providing specific feedback is a necessary security measure is likely due to a misinterpretation or over-generalisation of the reported vulnerability, because the proposed fix would not have allowed creation of duplicate accounts nor would have reduced security any way. In any case, this claim is undermined by the existence of services that had already fixed the issue, suggesting this vulnerable form of processing is not strictly necessary.

User experience (UX) concerns. A frequent justification, raised by 9 (26%) of the 34 services that provided substantive responses, was that indistinguishable error messages would create a frustrating user experience. This was particularly emphasised on password reset forms. For example, the secure alternative was dismissed by upload.ee as bordering on "*nonsense*" that would cause a user to "*wait endlessly for email that never arrives*".

This UX argument was often supplemented by citing legitimate interest necessity (GDPR Article 6(1)(f)) to reduce customer support costs. This justification, however, appears to overlook the critical test of necessity [28], as user-friendly and secure alternatives are possible and were proposed.

Curiously, several service providers (rimi.ee, cvkeskus.ee, aeromotors.ee) implied they had done a privacy balancing test, which is the step right after the necessity test. Such likely slip-up implies that the proposed fixes were not well understood by these service providers.

Necessity for Performing a Contract. There were 8 (24%) services that justified the vulnerable processing as necessary for performance of a contract, under GDPR Article 6(1)(b). Justifications centered on the argument that the processing was necessary for functions such as account management (ikea.ee, kaup24.ee, okidoki.ee, postimees.ee), access provisioning (Circle K app, forum.ee, k-rauta.ee), or helping users correct typing errors (aboutyou.ee).

However, the concept of "*necessity*" under GDPR is a strict standard. The successful implementation of secure forms by other services is practical proof that exposing account status is not, in fact, necessary to fulfill a contract.

Deflecting Accountability. A common tactic was to avoid accountability through deflection. The risk was downplayed, with some providers arguing it was a mere "*potential weakness*" and not a data breach (rimi.ee). Some dismissed the vulnerability without explanation (zalando.ee), while others suggested taking the matter directly to the DPA (AKI) (flirtic.ee, kv.ee).

This denial was often coupled with questionable legal interpretations, such as the likely incorrect claim that an email address is not personal data unless it contains a name (piletilevi.ee, olybet.ee, Maxima Estonia app). The argument possibly considers the narrower, North-American concept of Personally Identifiable Information (PII). EU's GDPR defines

personal data more broadly as "*any information relating to an identified or identifiable natural person*" [29].

A novel justification was the concern that the recommended fix could create a new vector for spam (kaup24.ee, okidoki.ee). This justification was confusing, as both services also provided password reset functionality, and thus are already susceptible to spamming their customers. Such a risk could be addressed with anti-bot measures like rate limiting, in which case personal data disclosure would be unnecessary.

Finally, responsibility was deflected onto third-party data processors by several providers (delfi.ee, ohtuleht.ee, postimees.ee and jpg.ope.ee), or onto an existing ISO/IEC 27001 certification (postimees.ee, kaup24.ee), demonstrating a misunderstanding of their non-delegable duties as likely data controllers under GDPR.

5.4.3 Other Notable Patterns

Anti-bot Measures. It was pointed out by several providers (jpg.ope.ee, city24.ee, euronics.ee) that the assessment of their anti-bot measures had been incorrect, though no further details were provided. The warning about these measures, intended solely to highlight an increased risk factor, nonetheless received significant attention from service providers.

It was also claimed by forum.ee that its use of Cloudflare made automated attacks impossible, a claim that was disproven when a simple script successfully mass-enumerated accounts. This highlights that the mere presence of an anti-bot service, used by 22 (44%) of the tested websites, does not guarantee protection if not configured correctly.

The use of CAPTCHA was argued against by some other providers (kaup24.ee, kv.ee) due to UX concerns or perceived ineffectiveness, demonstrating a de-prioritisation of this defensive layer.

Improper Handling of Data Subject Requests. In one case, the author's account was deleted entirely by fv.ee in response to the GDPR request, an action that was neither requested nor consented to. This drastic measure demonstrated a fundamental misunderstanding of data subject rights, conflating a request to restrict processing (Article 18) with a request for erasure (Article 17), particularly as erasure was explicitly opposed in the request. This action appears to violate the principle of data integrity and represents a failure to correctly handle a nuanced data subject request.

In a particularly unprofessional response, two providers (forum.ee, jpg.ope.ee) conducted an online search for the author's name to argue that the data was already public, an action that was a likely GDPR violation and appeared to be an intimidation attempt.

The suggestion to delete an account was also made by other providers (aboutyou.ee, flirtic.ee), indicating a broader lack of preparedness for this type of vulnerability report. In a contrasting response, it was likely incorrectly and also irrelevantly, noted by olybet.ee that an account could not be deleted due to legal obligations under the Gambling Act.

For context, the Estonian Gambling Act §53(3)[30] requires does require storing some user data "*for at least five years*". However, it is likely not strictly necessary to hold that data online, as the only requirement is it be stored "*in a manner enabling to produce such information while responding to an inquiry from the Tax and Customs Board, police or Financial Intelligence Unit*".

User-Specific "Fixes". Two providers implemented user-specific fixes that did not address the systemic vulnerability[4].

Circle K app proposed a fix that would result in the author "*no longer receiving operational communications from Circle K, including service updates, promotional materials and account-related notifications*", while the author could "*use the mobile application and benefit from Circle K's services normally*". The solution actually entailed replacing the account email with optedout_anon_xyz123@example.com, which also prevented author from logging in using their personal email.

The response from the provider of the Stuudium platform, used by jpg.ope.ee, was particularly problematic. The provider complied with the GDPR request for restriction of processing by blocking only the author from signing in to the service. This act of malicious compliance ignored the systemic vulnerability affecting all users and served to punish the good-faith disclosure.

5.4.4 Notable Individual Responses

An Atypical Fix. The approach of kae.edu.ee for addressing the issue was unique. Acknowledging the vulnerability, the provider stated that since the software was not under active

[4]These user-specific fixes are considered a refusal in the statistics, even if they do not appear as such in Appendix J.4.

development, the email-based registration form was completely disabled. While this fixed the vulnerability, it represents an extreme measure that sacrifices functionality for security.

Leaky Debug Code. Through correspondence with upload.ee, a separate information disclosure vulnerability was incidentally discovered. After a fix was implemented, a review of the page's HTML source revealed a debug row containing SQL query statistics, which inadvertently created a new side-channel. This was subsequently removed by the provider.

A Request for Consultation. A uniquely constructive response was received from jysk.ee, which, after promising to fix the vulnerability, extended an informal offer for further consultation on implementing a privacy-compliant registration process, citing the detail in the submitted report. The offer was politely declined, as the report already contained sufficient guidance.

Claiming an Abuse of Rights. The response from aboutyou.ee was notably confused. After misinterpreting the GDPR request as a request for account deletion, it was incorrectly claimed that the author was acting on behalf of the University of Tartu and that this constituted an "*abuse of rights*" under GDPR, a fundamental misreading of the request.

Under GDPR, the data controller must address data subject requests, unless the request is "*manifestly unfounded or excessive*", neither of which was raised as a concern. This interaction highlights a likely lack of understanding of data subject rights.

The Consent Claim. A response from aripaev.ee, handled by an external law firm, was the only service to cite the data subject's consent (GDPR Article 6(1)(a)) as the legal basis. However, this did not consider that the data subject whose email is being used might not be the one submitting the form, nor that consent can be withdrawn at any time under GDPR Article 7(3).

A Privacy Policy Change. Curiously, on the same day as the response from olybet.ee, they had updated their privacy policy.

The only perhaps relevant change is generalisation of data being processed. An earlier version of the privacy policy listed some purposes and bases of processing with a category "*veebilehe külastuse andmed*" (website visit data), which was changed to "*veebilehe andmed*" (website data). This could be interpreted as an attempt to cover other similar issues in the future, but it is unclear whether this change was intentional.

5.5 Outcomes

The disclosure process yielded mixed results, highlighting significant variations in security maturity among Estonian service providers. Furthermore, the disclosure process itself encountered logistical challenges, such as bounced emails and recipients unable to open the digitally signed `.asice` containers, which required follow-up communication with unsigned attachments and delayed the fixing timelines.

After 30 days from sending the GDPR requests, a final reassessment was conducted on the services that stated they had fixed the issue or that promised fixes in the future (see Appendix J.4). This round of reassessment was conducted in August 2025.

Overall, 40 (73%) of the 55 services provided a substantive response to at least one of the outreach attempts (either the vulnerability report, reassessment report or GDPR request). The vulnerability was completely fixed by 15 (27%) of the 55 services. Conversely, 15 (27%) of services provided no substantive response at all.

The quality of fixes varied: while many implemented robust solutions, some partial fixes merely added a CAPTCHA without addressing the underlying information disclosure, or made purely cosmetic changes, such as altering an error message, which did not constitute a meaningful security improvement. At the conclusion of the research period, a significant number of services remained unfixed, including 14 (25%) that explicitly refused to completely fix the issue and 13 (24%) that provided no specific details on remediation plans or lack thereof. Curiously, some services (e.g., `postimees.ee`, `rimi.ee`, `kv.ee`, `hind.ee`, `aeromotors.ee`) responded to initial disclosure positively or attempted fixing the issue, but later, in response to GDPR requests, refused to implement a complete fix and downplayed the risks.

The overall fix and response rates among the 55 tested services were mediocre (see Figure 41):

- **5 (9%) completely fixed the issue after initial disclosure**: `eliis.eu`, `inforegister.ee`, `kuldnebors.ee`, `opiq.ee`, `soov.ee`.
- **3 (5%) completely fixed the issue after reassessment report**: `amoremi.ee`, `apolloklubi.ee`, `city24.ee`.

- **7 (13%) completely fixed the issue after GDPR request**: harid.ee, ikea.ee, k-rauta.ee, Maxima Estonia app, nami-nami.ee, olybet.ee, upload.ee.
- **13 (24%) promised to fix the issue**: Circle K app, cvkeskus.ee, ekool.eu, Elisa Raamat app, euronics.ee, fv.ee, geenius.ee, jysk.ee, kae.edu.ee, kaup24.ee, okidoki.ee, piletilevi.ee, ResQ Club app.
- **14 (25%) refused to fix the issue**: aboutyou.ee, aeromotors.ee, äripaev.ee, delfi.ee, flirtic.ee, forum.ee, hind.ee, jpg.ope.ee, kv.ee, LIDL Plus app, ohtuleht.ee, postimees.ee, rimi.ee, zalando.ee
- **13 (24%) provided no details on remediation plans or lack thereof**: barbora.ee, bet365.ee, elron.ee, go3.tv, iha.ee, jupiter.err.ee, kava.ee, moodle.edu.ee, notino.ee, paavlikaltsukas.ee, peaasi.ee, solnet.ee, stena.ee.

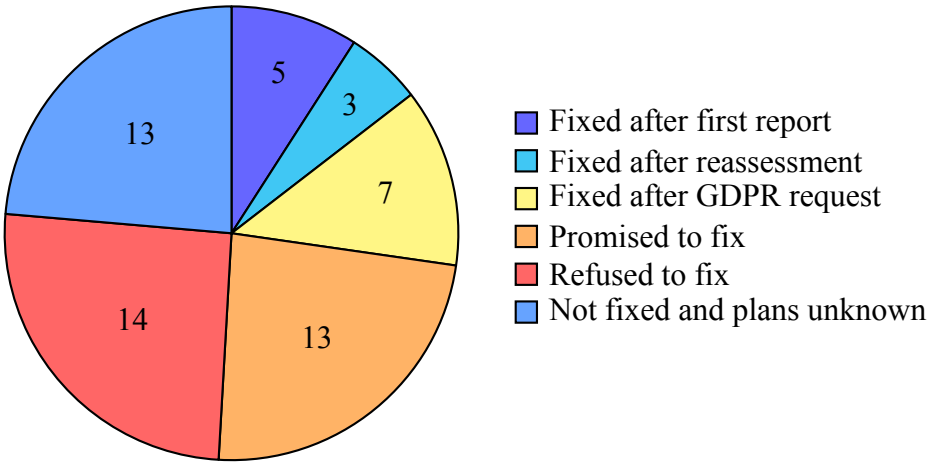

Figure 41. Distribution of outcomes for the addressed vulnerabilities

The observed overall fix rate was 15 (27%), or when including promises to fix, a potential rate of 28 (51%). The observed substantive response rate was 40 (73%) of the 55 services.

Both the fix rate and response rate are surprisingly high in contrast to the findings of the Swiss study by Maceiras et al. [3], where only 1 (2%) of 59 services fixed the issue and 5 (8%) responded in over a year. The greater success of this disclosure process can likely be attributed to the customised reports and the multi-stage follow-up strategy, where the formal GDPR request

was particularly effective. This thesis also targeted more local and small services, which may have been more responsive to direct outreach than larger, international platforms that were covered by the Swiss study.

6. Discussion and Recommendations

In this section, the study's findings are synthesised to interpret their significance in a broader context. This discussion analyses the systemic issues the findings reveal about data protection culture, technical practices, and regulatory adherence. The analysis culminates in actionable recommendations for service providers, DPAs, and users.

6.1 A Global Blind Spot

The findings of this study together with earlier research [3, 4], confirm that account existence leaks are a global blind spot, where services in even technologically mature nations struggle with ensuring data protection by design and by default. The fact that every tested service (including government-provided moodle.edu.ee and harid.ee) was vulnerable is not only a technical finding – it shows a deep, systemic disconnect between high-level ideals and the on-the-ground implementation of security-by-design principles.

The persistence of this vulnerability can be partly attributed to a development culture where SMEs lack dedicated security resources [23] and developers under pressure favour functional solutions over security documentation [24].

Notably, the response rate (75%) and the remediation rate (22%) observed in this study were significantly higher than in the work of Maceiras et al. [3] (8.5% and 1.7% respectively). This outcome may suggest a regional difference in provider responsiveness or security maturity, making the Estonian context a valuable case study.

6.2 The 80/20 Security Culture

A clear imbalance in security implementation was observed across different functionalities. Sign-in forms were found to be largely secure, with 44 (86%) of 51 tested sign-in forms being non-vulnerable. In contrast, account registration and email change forms were almost universally flawed: 50 (100%) of 50 tested registration forms and 33 (97%) of 34 tested email change forms were vulnerable (see Section 4.2).

This pattern suggests that security efforts are disproportionately focused on the most obvious attack surface – the sign-in page – while secondary account management flows are often neglected, despite being just as critical for data protection. Such disproportionate attention might be attributable to a common balancing act, where efforts are allocated to achieve the

greatest perceived impact with the least amount of work, akin to a Pareto-like principle (80% consequences come from 20% causes).

6.2.1 Risk of Shared Platforms

The vulnerabilities discovered in shared platforms such as *Piano.io* and *Stuudium* (see Section 4.4) move the problem from an individual service flaw to a systemic risk for the entire digital ecosystem. A single vulnerability in a shared service simultaneously exposes the user bases of major news portals or the personal data of students across numerous schools. The responses of some service providers demonstrated a diffused sense of responsibility, implying the third-party platform is solely responsible for security. This position likely challenges the fundamental principle of non-delegable duties under GDPR, which holds the data controller ultimately accountable for the processing done on their behalf. This pattern of centralised failure poses a significant threat, showing how interconnected digital ecosystems can propagate and amplify vulnerabilities on a massive scale.

6.2.2 Effectiveness of Countermeasures

Countermeasures against leaking account existence were revealed to be often weak or entirely absent. Crucial secondary defences like CAPTCHA and rate limiting were frequently missing or poorly implemented. The case of `okidoki.ee`, which performed its email existence check before the CAPTCHA validation, is a prime example of a reactive security posture. This approach renders the CAPTCHA ineffective against discovering account existence, indicating that defences are often bolted on as an afterthought rather than being integrated into the core logic of the authentication flow. Similarly, scripted tests showed that rate limiting on many services was not robust enough to prevent rapid, automated probing.

A number of excluded services mitigated leaking account existence via email by using usernames as the primary identifier. However, this approach simply shifts the problem to leaking account existence via usernames, which is inherently prone to information disclosure during registration due to the need to check for uniqueness (see Section 2.1.5). In most cases, usernames could be chosen by the users, where nothing prevents reuse on different websites. Such reuse increases risks, as knowledge of usernames associated with a user on one service could enable successfully discovering their accounts elsewhere. Reuse is less risky when the username is generic enough not to be effectively identifying the user.

For example, on `swedbank.ee` the username is assigned arbitrarily by the service provider. In that case, risks related to leaking account existence are not applicable, but the user has to

keep track of another *semi-random* piece of information, potentially increasing customer support costs due to forgotten usernames.

The most robust defence observed was the use of national eID schemes (ID-card, Mobile-ID, Smart-ID), which removes the vulnerable email identifier from the authentication process entirely. Similarly, third-party Single Sign-On (SSO) through widespread providers (Google, Facebook) has a similar effect by reducing both the attack surface across websites and the extent of information gained in a potential leak, as only the existence of the SSO account can be determined, not an association with a specific website.

6.3 Barriers to Remediation

Only a few services provided particularly unprofessional responses (e.g., `aboutyou.ee`, `jpg.ope.ee`, `fv.ee`, `flirtic.ee`). While this indicates general maturity in addressing security and privacy issues, there were still significant barriers to remediation.

Illusion of Necessity. A common justification for inaction was that the observable response discrepancy was necessary for either user experience or contract performance, often citing legitimate interest (GDPR Article 6(1)(f)) as the legal basis. This argument, however, likely relies on a flawed interpretation of GDPR's strict necessity tests. The fact that numerous services successfully implemented fixes without degrading their service is direct proof that the data-disclosing process was not necessary. The ability to provide a secure, user-friendly alternative, as demonstrated by the services that remediated the issue, invalidates the legal claim that the insecure method was essential.

Claims to Common Practice. Several providers deflected responsibility by pointing to the practices of industry giants like *Google* or *Facebook*. This appeal to common practice is a logical fallacy that likely holds no legal standing under GDPR. Each data controller must assess their risks and compliance independently; the actions of other companies do not set a legal precedent. This line of reasoning suggests a tendency towards reactive compliance, where organisations model their practices on others rather than proactively adhering to their own legal obligations. The claim from `kaup24.ee` that the problem is unsolvable further illustrates this culture of normalising, rather than remediating, known weaknesses.

6.4 Recommendations

Based on the findings, the following recommendations are proposed to address vulnerabilities that leak account existence.

6.4.1 For Service Providers

Adopt Secure Design Patterns. The secure patterns proposed in Section 2.1 could be adopted. Measures could be taken to prevent regression, especially considering the natural turnover of software developers.

Address Legacy Systems. For legacy systems where a full refactor is infeasible, providers should pre-plan for emergent risks and eventual sunsetting. Regulations like GDPR may force changes, while new security vulnerabilities can require systemic updates that are impractical for unmaintained codebases. When a service is no longer viable, providers must decide whether to accept potential legal consequences or invest in continued support. Where maintenance is overwhelming, a pre-planned strategy for sunsetting the platform is crucial. This can involve migrating valuable content to alternative platforms (e.g., videos to YouTube, text to the Web Archive) and forwarding users to modern, functional equivalents, ensuring a responsible transition rather than abandonment.

Contribute to Global Security Standards. This research identified two gaps in prominent security guidelines. First, the OWASP ASVS and Authentication Cheat Sheet omit the email change form as a vector for leaking account existence [9, 10]. Second, the ASVS primarily frames the issue as a risk of mass-harvesting identifiers, overlooking the distinct privacy risk of a targeted query to confirm a single individual's association with a service. A pull request could be proposed to these projects to incorporate these findings. Such a contribution would ensure the work has an impact beyond Estonia and helps improve global security standards.

6.4.2 For Data Protection Authorities (DPAs)

Publish Authoritative Technical Guidance. The widespread provider confusion and likely flawed legal justifications encountered during this research highlight a critical need for regulatory clarity. It is strongly recommended that the Estonian Data Protection Inspectorate (AKI) and other EU DPAs publish explicit, legally-backed technical guidance on preventing account existence leaks.

This guidance should state unequivocally that it is a data protection issue, outline unacceptable practices, and provide examples of compliant implementations. This would eliminate ambiguity, debunk common fallacies, and provide a clear standard against which data controllers can be held accountable.

6.4.3 For Users

Use Email Masking and Aliasing. While the primary responsibility lies with service providers, users can mitigate their exposure. Creating unique email addresses for different services using dedicated masking services (*SimpleLogin*, *Addy.io*, *Apple's Hide My Email*) makes it much harder for an attacker to guess a user's identifier. Alternatively, some email providers such as Gmail allow users to use an alias in form of 'myemail+alias@example.com', where the part right after + could include randomness to result in a unique and unguessable email address. Using unique emails is a strong defence because even if one email is known or exposed in a breach, others cannot be derived and remain secret.

6.5 Future Work

This study sets the basis for future research in several directions.

Escalation To DPA. The immediate next step for the author is to submit formal complaints to the Estonian Data Protection Authority (AKI) for all 27 (49%) of the tested service providers that failed to remediate the vulnerabilities or commit to doing so. This action shall aim to trigger official supervisory proceedings to compel the as-yet unfixed services to fix the vulnerabilities. It shall also aim to focus AKI's attention to the scale of the privacy concern, in hopes of prompting creation of guidelines that mention risks of leaking account existence and suggest mitigation strategies.

Long-Term Follow-Up. A follow-up study could be conducted to re-assess the 55 Estonian services in a few years. Such a study would measure the lasting impact of the disclosure and track whether security postures improve or regress over time.

Broaden the Scope. Another follow-up study could be conducted to apply the same methodology to phone numbers as identifiers, a vector explicitly excluded from this thesis's scope. Additionally, the research methodology – especially the multi-phase disclosure strategy using formal GDPR requests – could be replicated in other EU countries to determine whether the findings – the vulnerability prevalence and the response patterns – are consistent across different regions and cultures.

Develop Specialised Automation Tools. Following up on the discarded idea of using modern automation tools (see Section 3.7), a study should attempt to create a LLM-based agent. This agent would be tasked specifically to detect account existence leaks, addressing the concern of high cost and enabling large-scale, continuous auditing.

7. Conclusion

This thesis investigated the prevalence of account existence leaks across popular Estonian online services, revealing a critical gap between the nation's reputation as a digital leader and the systemic neglect of data protection by design. The findings confirm that a decades-old, well-documented security flaw persists on a scale that challenges the foundation of trust in Estonia's digital ecosystem.

The systematic analysis of 55 services answered the research questions posed. It confirmed that leaking account existence is a universal issue, with every tested service vulnerable through at least one vector – primarily registration and email change forms **(RQ1)**. Furthermore, a majority of services, 31 (56%), lacked basic anti-automation countermeasures, facilitating the stealthy account existence checks and indicating that mitigating security measures were absent or ineffective **(RQ2)**.

A multi-stage responsible disclosure process, combining voluntary reporting with formal GDPR requests, proved effective for compelling remediation, resulting in 15 (27%) of services implementing fixes and 13 (24%) promising to do so. However, this engagement also revealed systemic challenges: widespread misunderstanding of GDPR obligations, a tendency to deflect accountability, and resistance to implementing fixes based on flawed justifications related to user experience or common practice **(RQ3)**.

The primary contribution of this work is the first comprehensive, large-scale analysis of account existence leaks in the Estonian context. This study provides empirical evidence of a systemic security failure and validates a GDPR-based disclosure strategy that achieved a significantly higher rate of correction than observed in similar international research. The research culminates in actionable recommendations for service providers, DPAs, and users.

Ultimately, account existence leaks are not merely a technical oversight but a symptom of a deeper cultural and regulatory failure to uphold data protection by design. This failure manifests as a persistent gap between policy and practice, leaving users exposed to significant and avoidable privacy risks. Closing this gap requires a fundamental shift from reactive compliance to a proactive culture of security-by-design.

References

[1] OWASP Foundation. Testing for Account Enumeration and Guessable User Account (OTG-IDENT-004). https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/03-Identity_Management_Testing/04-Testing_for_Account_Enumeration_and_Guessable_User_Account. Accessed: 2025-04-19.

[2] MITRE CWE. CWE-204: Observable Response Discrepancy. https://cwe.mitre.org/data/definitions/204.html. Accessed: 2025-04-19.

[3] Maceiras M., Salehzadeh Niksirat K., Bernard G., Garbinato B., Cherubini M., Humbert M., and Huguenin K. Know their Customers: An Empirical Study of Online Account Enumeration Attacks. *ACM Trans. Web* 18.3 (June 2024). DOI: 10.1145/3664201. https://doi.org/10.1145/3664201.

[4] Hasegawa A. A., Watanabe T., Shioji E., and Akiyama M. I know what you did last login: inconsistent messages tell existence of a target's account to insiders. *Proceedings of the 35th Annual Computer Security Applications Conference*. ACSAC '19. San Juan, Puerto Rico, USA: Association for Computing Machinery, 2019, pp. 732–746. DOI: 10.1145/3359789.3359832. https://doi.org/10.1145/3359789.3359832.

[5] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC. Accessed on 2025-07-24. Apr. 27, 2016. https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[6] European Union. Trade secrets. https://europa.eu/youreurope/business/running-business/intellectual-property/trade-secrets/index_en.htm. Accessed: 2025-04-19.

[7] MITRE ATT&CK. Account Discovery (T1087). https://attack.mitre.org/techniques/T1087/. Accessed: 2025-04-19.

[8] University of Tartu. Guidelines for using AI applications in teaching and studies. accessed on August 9, 2025. https://ut.ee/en/content/guidelines-using-ai-applications-teaching-and-studies.

[9] OWASP Foundation. Authentication Cheat Sheet. https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html. Accessed: 2025-04-19.

[10] Open Worldwide Application Security Project. Application Security Verification Standard (ASVS) 5.0.0. Accessed: 2025-06-24. May 2025. https://owasp.org/www-project-application-security-verification-standard/.

[11] MITRE Corporation. CVE-1999-0656. https://www.cve.org/CVERecord?id=CVE-1999-0656. Accessed: 2025-06-29. Feb. 2000.

[12] MITRE Corporation. CVE-2001-1483. https://www.cve.org/CVERecord?id=CVE-2001-1483. Accessed: 2025-06-29. June 2005.

[13] MITRE Corporation. CVE-2001-1013. https://www.cve.org/CVERecord?id=CVE-2001-1013. Accessed: 2025-06-29. Feb. 2002.

[14] MITRE Corporation. CVE-2001-1528. https://www.cve.org/CVERecord?id=CVE-2001-1528. Accessed: 2025-06-29. July 2005.

[15] MITRE Corporation. CVE-2004-2150. https://www.cve.org/CVERecord?id=CVE-2004-2150. Accessed: 2025-06-29. July 2005.

[16] Christey S. PLOVER - Preliminary List Of Vulnerability Examples for Researchers (version 0.24). Mar. 2006. https://cwe.mitre.org/documents/sources/PLOVER.pdf.

[17] MITRE Corporation. History. https://cwe.mitre.org/about/history.html. Accessed: 2025-06-29. Sept. 2022.

[18] Kim E., Park K., Kim H., and Song J. Design and analysis of enumeration attacks on finding friends with phone numbers: A case study with KakaoTalk. *Computers & Security* 52 (2015), pp. 267–275. DOI: https://doi.org/10.1016/j.cose.2015.04.008. https://www.sciencedirect.com/science/article/pii/S0167404815000589.

[19] Kim J., Kim K., Cho J., Kim H., and Schrittwieser S. Hello, Facebook! Here Is the Stalkers' Paradise!: Design and Analysis of Enumeration Attack Using Phone Numbers on Facebook. Dec. 2017, pp. 663–677. DOI: 10.1007/978-3-319-72359-4_41.

[20] Jang H., Ji W., Woo S., and Kim H. Design and Evaluation of Enumeration Attacks on Package Tracking Systems. Aug. 2020, pp. 543–559. DOI: 10.1007/978-3-030-55304-3_28.

[21] Kim K., Kim T., Lee S., Kim S., and Kim H. When Harry Met Tinder: Security Analysis of Dating Apps on Android. *Secure IT Systems*. Ed. by Gruschka N. Cham: Springer International Publishing, 2018, pp. 454–467. DOI: 10.1007/978-3-030-03638-6_28.

[22] Swedbank internet bank user data leaked (in Latvian). Blog post (Archived). Accessed: 2025-04-19. 2013. https://web.archive.org/web/20180129092106/https://defense.lv/2013/12/05/nopludusi-swedbank-internetbankas-lietotaju-dati/.

[23] Ambreen L., Jain M., and Loonkar S. Effective cybersecurity risk management practices for small and medium-sized enterprises: A comprehensive review. *Multidisciplinary Reviews* 6 (May 2024), 2023ss080. DOI: 10.31893/multirev.2023ss080.

[24] Gorski P. L., Möller S., Wiefling S., and Iacono L. L. “I just looked for the solution!”On Integrating Security-Relevant Information in Non-Security API Documentation to Support Secure Coding Practices. *IEEE Transactions on Software Engineering* 48.9 (2022), pp. 3467–3484. DOI: 10.1109/TSE.2021.3094171.

[25] The Alexa.com Team. We will be retiring the Alexa.com APIs on December 15, 2022. Archived announcement. Accessed: June 28, 2025. 2022. https://web.archive.org/web/20221126133457/https://support.alexa.com/hc/en-us/articles/4411466276375.

[26] Le Pochat V., Van Goethem T., Tajalizadehkhoob S., Korczyński M., and Joosen W. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. *Proceedings of the 26th Annual Network and Distributed System Security Symposium*. NDSS 2019. Feb. 2019. DOI: 10.14722/ndss.2019.23386.

[27] Security.txt Standard. https://securitytxt.org/. Accessed: 2025-04-19.

[28] European Data Protection Board. Legitimate interest: when and how to apply it. European Data Protection Board. Public-facing summary of the conditions for applying legitimate interest under Article 6(1)(f) of the GDPR. Oct. 2024. https://www.edpb.europa.eu/system/files/2024-10/edpb_summary_202401_legitimateinterest_en.pdf.

[29] Thuret-Benoist M. The difference between PII and Personal Data - blog. [Accessed 18-07-2025]. https://techgdpr.com/blog/difference-between-pii-and-personal-data/.

[30] Gambling Act. Riigi Teataja. Accessed: 2025-08-10. Official consolidated text (English translation). https://www.riigiteataja.ee/en/eli/ee/525112020004/consolide.

Appendices

A. Inclusions From Top 150 .ee Websites in Tranco List

Full list generated on October 24, 2024, available at https://tranco-list.eu/list/249P9. [26]

Rank	Website	Included	Explanation
1	linktr.ee	No	Not targeted at Estonians
2	filmyzilla.com.ee	No	Unable to load
3	lin.ee	No	Unable to load
4	google.ee	No	Provides an email address
5	iha.ee	Yes	
6	delfi.ee	Yes	
7	shope.ee	No	Unable to load
8	postimees.ee	Yes	
9	err.ee	Yes	Used jupiter.err.ee
10	auto24.ee	No	Has no email-based authentication
11	ut.ee	No	Has no account registration
12	shp.ee	No	Unable to load
13	found.ee	No	Not targeted at Estonians
14	tr.ee	No	Not targeted at Estonians
15	orb.ee	No	Unable to load
16	tallinn.ee	No	Has no email-based authentication
17	ohtuleht.ee	Yes	
18	swedbank.ee	No	Has no account registration
19	mail.ee	No	Provides an email address
20	neti.ee	No	Does not have user accounts
21	kaup24.ee	Yes	
22	okidoki.ee	Yes	
23	cookieclicker.ee	No	Not targeted at Estonians
24	kuldnebors.ee	Yes	
25	bet365.ee	Yes	
26	cvkeskus.ee	Yes	
27	osta.ee	No	Has no email-based authentication
28	kv.ee	Yes	

Rank	Website	Included	Explanation
29	euronics.ee	Yes	
30	olybet.ee	Yes	
31	jalgpall.ee	No	Does not have user accounts
32	mnt.ee	No	Has no email-based authentication
33	elisa.ee	No	Has no account registration
34	ilmateenistus.ee	No	Does not have user accounts
35	tootukassa.ee	No	Has no email-based authentication
36	soov.ee	Yes	
37	directo.ee	No	Has no email-based authentication
38	33win.ee	No	Not targeted at Estonians
39	online.ee	No	Provides an email address
40	eestiloto.ee	No	Has no email-based authentication
41	ikea.ee	Yes	
42	terviseportaal.ee	No	Has no email-based authentication
43	hot.ee	No	Provides an email address
44	upload.ee	Yes	
45	eesti.ee	No	Has no email-based authentication
46	rik.ee	No	Has no email-based authentication
47	telia.ee	No	Has no account registration
48	tv3.ee	Yes	Used play.tv3.ee
49	imgtr.ee	No	Not targeted at Estonians
50	zone.ee	No	Has no email-based authentication
51	lemm.ee	No	Not targeted at Estonians
52	tlu.ee	No	Has no account registration
53	ttu.ee	No	Has no account registration
54	k-rauta.ee	Yes	
55	rue.ee	No	Not targeted at Estonians
56	emta.ee	No	Has no email-based authentication
57	paste.ee	No	Not targeted at Estonians
58	pmo.ee	No	related to postimees.ee
59	aripaev.ee	Yes	
60	vm.ee	No	Has no email-based authentication

Rank	Website	Included	Explanation
61	estpak.ee	No	related to telia.ee
62	elion.ee	No	related to telia.ee
63	seb.ee	No	Has no account registration
64	ria.ee	No	Has no email-based authentication
65	riigiteataja.ee	No	Has no email-based authentication
66	opiq.ee	Yes	
67	lhv.ee	No	Has no account registration
68	eki.ee	No	Does not have user accounts
69	zalando.ee	Yes	
70	inforegister.ee	Yes	
71	city24.ee	Yes	
72	1a.ee	No	related to k-rauta.ee
73	e-resident.gov.ee	No	Does not have user accounts
74	elron.ee	Yes	
75	teatmik.ee	No	Does not have user accounts
76	jysk.ee	Yes	
77	piletilevi.ee	Yes	
78	webmail.ee	No	Provides an email address
79	rimi.ee	Yes	
80	tele2.ee	No	Has no account registration
81	notino.ee	Yes	
82	seti.ee	No	Unable to load registration
83	eev.ee	No	Not targeted at Estonians
84	veebimajutus.ee	No	Has no email-based authentication
85	taltech.ee	No	related to ttu.ee
86	apollokino.ee	Yes	Used apolloklubi.ee
87	barbora.ee	Yes	
88	kava.ee	Yes	
89	aeromotors.ee	Yes	
90	tartu.ee	No	Does not have user accounts
91	dyson.com.ee	No	Does not have user accounts
92	nami-nami.ee	Yes	

Rank	Website	Included	Explanation
93	paavlikaltsukas.ee	Yes	
94	linki.ee	No	Unable to load
95	blogspot.com.ee	No	Unable to load
96	eenet.ee	No	Does not have user accounts
97	stat.ee	No	Has no email-based authentication
98	cert.ee	No	Has no email-based authentication
99	solnet.ee	Yes	
100	visittallinn.ee	No	Does not have user accounts
101	f48.ee	No	Does not have user accounts
102	politsei.ee	No	Has no email-based authentication
103	omniva.ee	No	Has no email-based authentication
104	riigikogu.ee	No	Has no email-based authentication
105	hom.ee	No	Not targeted at Estonians
106	icds.ee	No	Does not have user accounts
107	fv.ee	Yes	
108	jakarta.ee	No	Not targeted at Estonians
109	sci-hub.ee	No	Does not have user accounts
110	yandex.ee	No	Provides an email address
111	emu.ee	No	Has no account registration
112	uueduudised.ee	No	Does not have user accounts
113	hind.ee	Yes	
114	moodle.edu.ee	Yes	
115	rce.ee	No	Unable to load
116	teliatv.ee	No	related to telia.ee
117	valitsus.ee	No	Has no email-based authentication
118	staycool.ee	No	Unable to load
119	planet.ee	No	related to zone.ee
120	tahvel.edu.ee	Yes	looked at HarID
121	id.ee	No	Does not have user accounts
122	estonia-company.ee	No	Not targeted at Estonians
123	artun.ee	No	Has no account registration
124	amoremi.ee	Yes	

Rank	Website	Included	Explanation
125	flirtic.ee	Yes	
126	starman.ee	No	related to elisa.ee
127	oci.ee	No	Not targeted at Estonians
128	geenius.ee	Yes	
129	elkdata.ee	No	related to veebimajutus.ee
130	tallnerk.ee	No	Unable to load
131	maaamet.ee	No	Has no email-based authentication
132	toits.ee	No	Not targeted at Estonians
133	aboutyou.ee	Yes	
134	otp.ee	No	Does not have user accounts
135	folklore.ee	No	Provides an email address
136	forum.ee	Yes	
137	dv.ee	No	related to aripaev.ee
138	1182.ee	No	Unable to load registration
139	stena.ee	Yes	
140	ruvikol.ee	No	Unable to load
141	osport.ee	No	Unable to load registration
142	vesikoer.ee	No	Has no email-based authentication
143	airport.ee	No	Does not have user accounts
144	www.ee	No	Unable to load registration
145	mfa.ee	No	Unable to load
146	energia.ee	No	Has no email-based authentication
147	bauhof.ee	No	Has no email-based authentication
148	apollo.ee	No	related to apollo.ee
149	kae.edu.ee	Yes	
150	empresaenestonia.ee	No	Not targeted at Estonians

B. Inclusions From Top 20 Android Apps in Play Store

Category "*Top for €0*", accessed on January 16th, 2025.

Table 2. Inclusions from top 20 Android apps in Play Store

Rank	App	Included	Explanation
1	rednote	No	Not targeted at Estonians
2	ChatGPT	No	Not targeted at Estonians
3	Temu: Shop Like a Billionare	No	Not targeted at Estonians
4	Adobe Acrobat Reader: Edit PDF	No	Not targeted at Estonians
5	Kaup24.ee Mobiilne e-pood	No	Uses webview for authentication-related flows; covered under kaup24.ee analysis
6	PDF Reader - PDF Converter	No	Not targeted at Estonians
7	Lidl Plus	Yes	
8	Maxima Estonia	Yes	
9	TikTok	No	Not targeted at Estonians
10	WhatsApp Messenger	No	Not targeted at Estonians
11	PDF Reader and PDF Editor	No	Not targeted at Estonians
12	Coop Eesti	No	Has no email-based authentication
13	Duolingo	No	Not targeted at Estonians
14	Circle K	Yes	
15	Smart-ID	No	Has no email-based authentication
16	Rimi	No	Uses webview for authentication-related flows; covered under rimi.ee analysis
17	Telegram	No	Not targeted at Estonians
18	Bolt Food: Delivery and Takeaway	No	Has no email-based authentication
19	Partnerkaart	No	Has no email-based authentication
20	Google Wallet	No	Not targeted at Estonians

C. Inclusions From Top 20 iOS Apps in App Store

Category "*Top Free Apps*", accessed on January 16th, 2025.

Table 3. Inclusions from top 20 iOS apps in App Store

Rank	App	Included	Explanation
1	rednote	No	Not targeted at Estonians
2	ChatGPT	No	Not targeted at Estonians
3	Threads	No	Not targeted at Estonians
4	Coop Eesti	No	Has no email-based authentication
5	Zalando - fashion and clothing	No	Uses webview for authentication-related flows; covered under zalando.ee analysis
6	Temu: Shop Like a Billionare	No	Not targeted at Estonians
7	MAXIMA Eesti	Yes	
8	Google Chrome	No	Not targeted at Estonians
9	Lidl Plus	Yes	
10	Rimi	No	Uses webview for authentication-related flows; covered under rimi.ee analysis
11	Google - More ways to search	No	Not targeted at Estonians
12	Remini - AI Photo Enhancer	No	Not targeted at Estonians
13	Partnerkaart	No	Has no email-based authentication
14	RIA DigiDoc	No	Has no email-based authentication
15	Circle K	Yes	
16	Duolingo - Language Lessons	No	Not targeted at Estonians
17	ResQ Club	Yes	
18	Microsoft Teams	No	Not targeted at Estonians
19	WhatsApp Messenger	No	Not targeted at Estonians
20	Elisa Raamat	Yes	

D. Example Initial Disclosure Email

Subject: *User email address enumeration vulnerability on example.com*

Hello,

During our research on user account enumeration flaws on Estonian online services, we found that your *<website|app>* is vulnerable to the exposure of registered users.

The vulnerability report is attached.

We would appreciate it if you could confirm receipt of this email to prevent us from reaching out to you through other channels.

Best regards,
Gregor Eesmaa
Computer Science MSc student
University of Tartu, Institute of Computer Science

E. Example Initial Disclosure Report

Account Enumeration Vulnerability:
`example.com`

Gregor Eesmaa
gregor.eesmaa@ut.ee
University of Tartu

August 18, 2025

1 Introduction

Account enumeration is a security vulnerability enabling attackers to determine if specific user accounts exist on a service. The vulnerability usually lies in the account registration functionality of a service, where an error message is returned, indicating that a user with the specified email address is already registered. However, an online service can also leak this information in other, more subtle ways, which are often overlooked by software developers.

On 2025-03-01, we tested `example.com` and found that the service is vulnerable to account enumeration. **The vulnerability allows any party to test whether a user with a specific email address is registered with the service.** Disclosing such information to third parties constitutes a data breach, as an email address and the fact of whether its holder has an account with an online service are considered personal data, and may be disclosed to third parties only if there is a legal basis for doing so [1].

We advise you to investigate the potential data breach, and notify the supervisory authority and the affected data subjects, if necessary. After **2025-04-01**, we will reassess the service and notify the Estonian Data Protection Inspectorate in case the vulnerability has not been mitigated. Detailed guidelines for mitigating this type of flaw are available in [2].

2 Vulnerabilities Found

We tested the login form, password reset form, account registration form and email change form of `example.com`.We identified security issues in all of these functionalities.

The vulnerabilities found are described in more detail in subsections below.

2.1 Login form

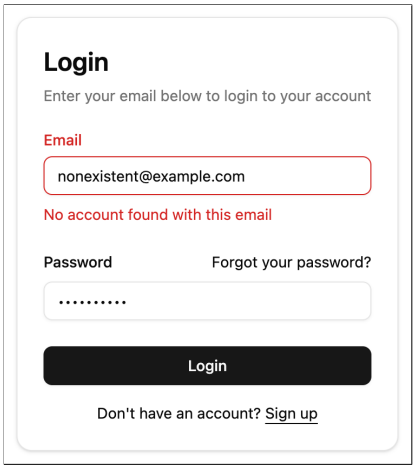

Figure 1: The vulnerability in the login form

The login form is susceptible to account enumeration attacks. This is because when a user with the provided email does not exist, the form shows a different error message compared to when the password is incorrect (see Figure 1). Additionally, the form appears to lack anti-bot measures such as CAPTCHA, enabling attackers to easily automate these attacks [3].

It is also crucial to eliminate any side-channels that an attacker could exploit to differentiate between account existence and non-existence. For example, the response should not be faster for an existing account than for an email with which an account does not exist.

To mitigate the flaw, return the same error message whether the email is registered or not. For example, always return the message "Email or password is incorrect". [2]

2.2 Password reset form

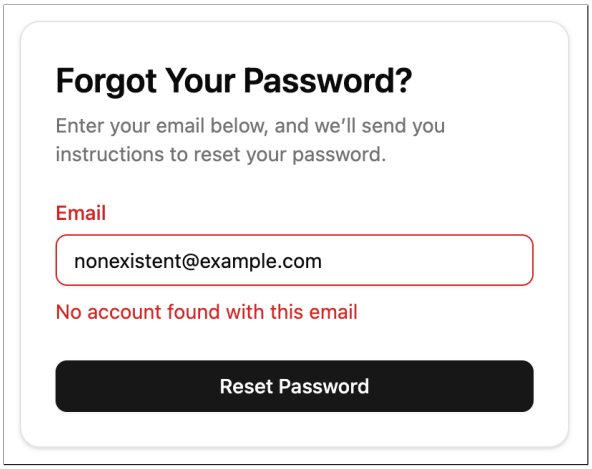

Figure 2: The vulnerability in the password reset form

The password reset form is also susceptible to account enumeration attacks. This is because when a password reset is requested for an email address that is not registered with the service, the form shows an error message (see Figure 2). Additionally, the form appears to lack anti-bot measures such as CAPTCHA, enabling attackers to easily automate these attacks [3].

It is also crucial to eliminate any side-channels that an attacker could exploit to differentiate between account existence and non-existence. For example, the response should not be faster for an existing account than for an email with which an account does not exist.

To mitigate the flaw, return the same message whether the email is registered or not. For example, the message could read as follows: "A password reset link has been sent if an account with this email exists". [2]

2.3 Account registration form

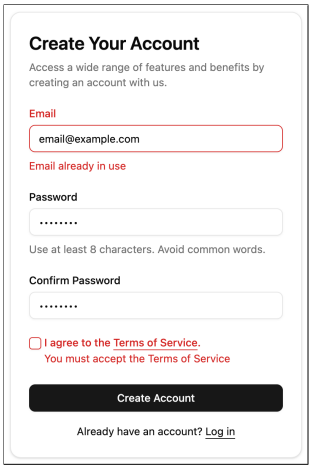

Figure 3: The vulnerability in the account registration form

The account registration form is also susceptible to account enumeration attacks. This is because when the provided email address is already taken, the form shows an error message (see Figure 3). Additionally, the form appears to lack anti-bot measures such as CAPTCHA, enabling attackers to easily automate these attacks [3].

The form normally sends a confirmation email to the email owner on successful submission. However, by introducing validation errors in the form, an attacker can determine whether an email address is already registered, without successfully submitting the form. This allows the attacker to also verify unregistered email addresses without triggering a confirmation email, thereby ensuring that the email owner remains unaware of the potential attack.

It is also crucial to eliminate any side-channels that an attacker could exploit to differentiate between account existence and non-existence. For example, the response should not be faster for an existing account than for an email with which an account does not exist.

To mitigate the flaw, return the same message whether the email is registered or not. For example, the message could read as follows: "We have sent further instructions to the provided email address". Send an email in both cases, but differentiate the content based on account existence. For example, for new registration, provide means for account activation, and for existing accounts, provide means for account recovery. [2]

2.4 Email change form

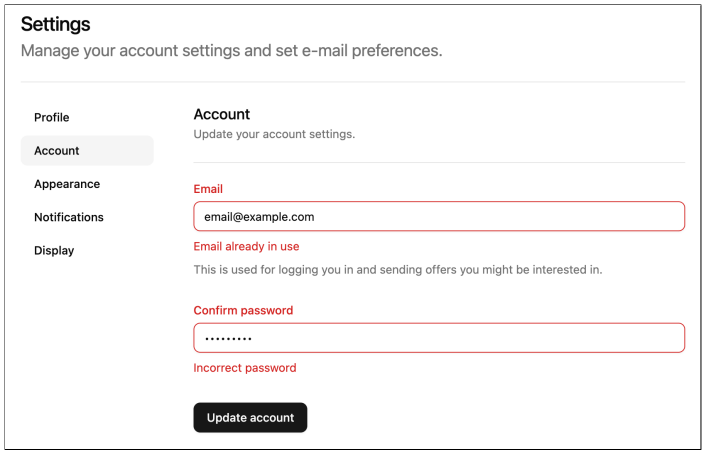

Figure 4: The vulnerability in the email change form

The email change form is also susceptible to account enumeration attacks. This is because when the provided email address is already taken, the form shows an error message (see Figure 4). Additionally, the form appears to lack anti-bot measures such as CAPTCHA, enabling attackers to easily automate these attacks [3].

Moreover, no confirmation email is sent to the provided email address after this form is submitted. Additionally, by introducing validation errors in the form, the attacker can avoid submitting the form altogether. This allows the attacker to verify unregistered email addresses, ensuring that the email owner remains unaware of the potential attack.

It is also crucial to eliminate any side-channels that an attacker could exploit to differentiate between account existence and non-existence. For example, the response should not be faster for an existing account than for an email with which an account does not exist.

To mitigate the flaw, return the same message whether the email is registered or not. For example, the message could read as follows: "We have sent further instructions to the provided new email address". Send an email in both cases, but differentiate the content based on account existence. For example, if the email is unused, provide means for confirming the new email, but if the email is used, provide means for account recovery.

3 Security Contacts

A valid `security.txt` [4] file was not found on `example.com`.We recommend implementing a `security.txt` file to ensure any future security issues can be reported to the appropriate contact person. In its absence, we have taken the following actions:

- No contact emails were found in a privacy policy of `example.com`.
- The email address `support@example.com` was found in the contact or help page of `example.com` and this report was sent to this email address on 2025-03-02, with no confirmation of receipt received to date.
- The email address `domains@example.com` was found in the Estonian Internet Foundation WHOIS database and this report was sent to this email address on 2025-03-09, with no confirmation of receipt received to date.
- The email address `john@example.com` was found in the relevant business registry and this report was sent to this email address.

About This vulnerability report is part of an ongoing study on user enumeration vulnerabilities in Estonian online services. The study is conducted by the University of Tartu master's student Gregor Eesmaa (supervised by Arnis Paršovs - arnis.parsovs@ut.ee). The findings of this study will be published in a master's thesis scheduled for defence in August 2025.

References

[1] European Union. *General Data Protection Regulation (GDPR): Regulation (EU) 2016/679*. Official Journal of the European Union, L 119/1. 2016. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679.

[2] OWASP. *Authentication Cheat Sheet - Authentication and Error Messages*. Accessed: 2025-01-26. URL: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html#authentication-and-error-messages.

[3] OWASP. *Authentication Cheat Sheet - Protect Against Automated Attacks*. Accessed: 2025-01-26. URL: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html#protect-against-automated-attacks.

[4] EdOverflow and Yakov Shafranovich. *security.txt – A proposed standard which allows websites to define security policies*. Accessed: 2025-01-26. URL: https://securitytxt.org/.

F. Example Reassessment Report Email

Subject: *Re: User email address enumeration vulnerability on example.com*

Hello,

We have conducted a follow-up reassessment of the service, and we appreciate your commitment to improving the security of your service. However, we also found that the account enumeration vulnerability is **still not fully mitigated**.

The attached updated report provides an overview of the remaining concerns and is intended to guide further corrective actions.

Best regards,
Gregor Eesmaa
Computer Science MSc student
University of Tartu, Institute of Computer Science

G. Example Reassessment Report

Account Enumeration Vulnerability:
`example.com`

Gregor Eesmaa
gregor.eesmaa@ut.ee
University of Tartu

August 18, 2025

1 Introduction

Account enumeration is a security vulnerability enabling attackers to determine if specific user accounts exist on a service. The vulnerability usually lies in the account registration functionality of a service, where an error message is returned, indicating that a user with the specified account identifier is already registered. However, an online service can also leak this information in other, more subtle ways, which are often overlooked by software developers. For example, even without a direct message, small visual differences in responses, or slight variations in how the server behaves (like the exact data returned) for existing versus non-existing accounts, can still reveal if an account is registered.

On 2025-04-01, we reassessed `example.com` and found that despite significant changes, **the service is still vulnerable to account enumeration**. Since the initial findings have been already addressed, this report will now highlight the more subtle residual hints of account existence that were identified.

If the account identifier of an online service is personal data (e.g. email address, personal code etc), then the fact, whether it is associated to an account, is also considered personal data. Any disclosure of personal data to third parties without a legal basis constitutes a data breach [1].

We advise you to investigate the potential data breach, and notify the supervisory authority and the affected data subjects, if necessary. After **2025-05-01**, we will reassess the service and notify the Estonian Data Protection Inspectorate in case the vulnerability has not been mitigated. Detailed guidelines for mitigating this type of flaw are available in [2].

2 Vulnerabilities Found

We tested the login form, password reset form, account registration form and email change form of `example.com`.We identified security issues in all of these functionalities.

The vulnerabilities found are described in more detail in subsections below.

2.1 Login form

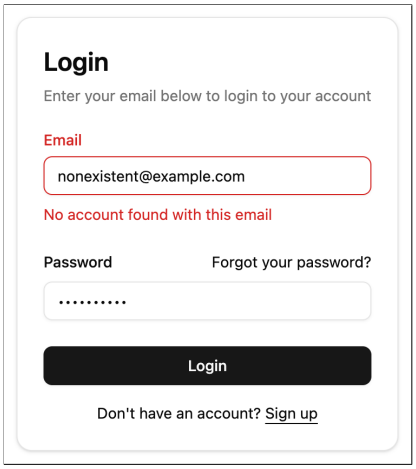

Figure 1: The vulnerability in the login form

The login form is susceptible to account enumeration attacks. This is because when a user with the provided email does not exist, there are subtle differences in the response received from the server compared to when the password is incorrect (see Figure 1).

It is also crucial to eliminate any side-channels that an attacker could exploit to differentiate between account existence and non-existence. For example, the response should not be faster for an existing account than for an email with which an account does not exist.

To mitigate the flaw, the response must be uniform for both registered and unregistered email addresses. This uniformity must apply to the message displayed to the user as well as the underlying HTTP response details (like status codes, headers, and body content).

For example, the indistinguishable user-facing message could be: “Email or password is incorrect”. [2]

2.2 Password reset form

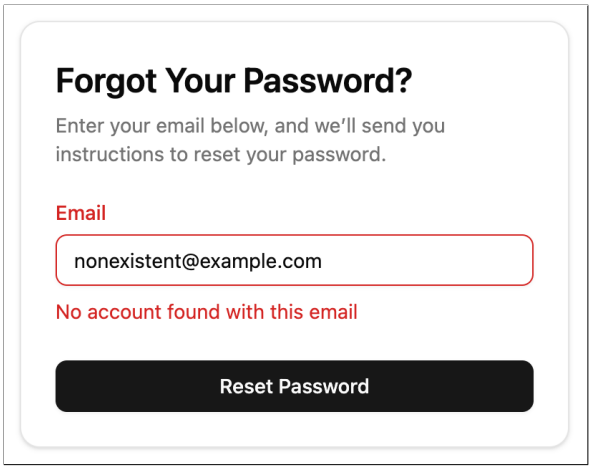

Figure 2: The vulnerability in the password reset form

The password reset form is also susceptible to account enumeration attacks. This is because when a password reset is requested for an email address that is not registered with the service, the resulting view appears visually different, compared to when the email is registered (see Figure 2).

It is also crucial to eliminate any side-channels that an attacker could exploit to differentiate between account existence and non-existence. For example, the response should not be faster for an existing account than for an email with which an account does not exist.

To mitigate the flaw, the response must be uniform for both registered and unregistered email addresses. This uniformity must apply to the message displayed to the user as well as the underlying HTTP response details (like status codes, headers, and body content).

For example, the indistinguishable user-facing message could be: "A password reset link has been sent if an account with this email exists". [2]

2.3 Account registration form

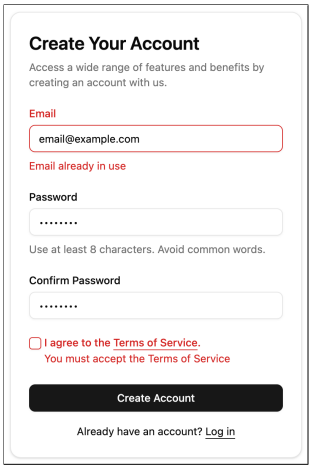

Figure 3: The vulnerability in the account registration form

The account registration form is also susceptible to account enumeration attacks. This is because when the provided email address is already taken, the form shows an error message (see Figure 3).

The form normally sends a confirmation email to the email owner on successful submission. However, by introducing validation errors in the form, an attacker can determine whether an email address is already registered, without successfully submitting the form. This allows the attacker to also verify unregistered email addresses without triggering a confirmation email, thereby ensuring that the email owner remains unaware of the potential attack.

It is also crucial to eliminate any side-channels that an attacker could exploit to differentiate between account existence and non-existence. For example, the response should not be faster for an existing account than for an email with which an account does not exist.

To mitigate the flaw, the response must be uniform for both registered and unregistered email addresses. This uniformity must apply to the message displayed to the user as well as the underlying HTTP response details (like status codes, headers, and body content).

For example, the indistinguishable user-facing message could be: “We have sent further instructions to the provided email address”. Send an email in both cases, but differentiate the content based on account existence. For example, for new registration, provide means for account activation, and for existing accounts, provide means for account recovery. [2]

2.4 Email change form

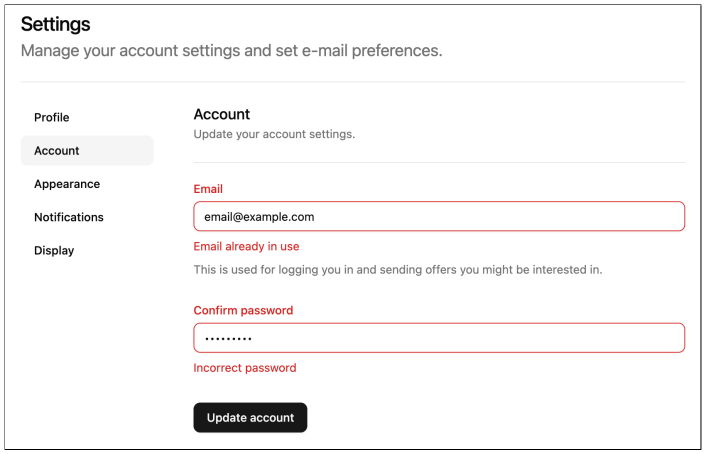

Figure 4: The vulnerability in the email change form

The email change form is also susceptible to account enumeration attacks. This is because when the provided email address is already taken, the form shows an error message (see Figure 4).

Moreover, no confirmation email is sent to the provided email address after this form is submitted. Additionally, by introducing validation errors in the form, the attacker can avoid submitting the form altogether. This allows the attacker to verify unregistered email addresses, ensuring that the email owner remains unaware of the potential attack.

It is also crucial to eliminate any side-channels that an attacker could exploit to differentiate between account existence and non-existence. For example, the response should not be faster for an existing account than for an email with which an account does not exist.

To mitigate the flaw, the response must be uniform for both registered and unregistered email addresses. This uniformity must apply to the message displayed to the user as well as the underlying HTTP response details (like status codes, headers, and body content).

For example, the indistinguishable user-facing message could be: “We have sent further instructions to the provided new email address”. Send an email in both cases, but differentiate the content based on account existence. For example, if the email is unused, provide means for confirming the new email, but if the email is used, provide means for account recovery.

About This vulnerability report is part of an ongoing study on user enumeration vulnerabilities in Estonian online services. The study is conducted by the University of Tartu master's student Gregor Eesmaa (supervised by Arnis Paršovs - arnis.parsovs@ut.ee). The findings of this study will be published in a master's thesis scheduled for defence in August 2025.

References

[1] European Union. *General Data Protection Regulation (GDPR): Regulation (EU) 2016/679*. Official Journal of the European Union, L 119/1. 2016. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679.

[2] OWASP. *Authentication Cheat Sheet - Authentication and Error Messages*. Accessed: 2025-01-26. URL: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html#authentication-and-error-messages.

H. Example GDPR Request Email

Subject: *IKÜM taotlus seoses isikuandmete töötlemisega teenuses example.com / GDPR Request Regarding Processing of Personal Data on example.com*

```
Lugupeetud example.com vastutav andmetöötleja

Pöördun, et edastada ametlikult mure isikuandmete kaitse
üldmääruse (IKÜM) alusel seoses minu isikuandmete töötlemisega
teenuses example.com, mida pean ebaseaduslikuks ning mis võib
viia nende avaldamiseni selleks volitamata isikutele ilma selge
õigusliku aluseta.

Manuses on ametlik taotlus, milles teostan oma õigusi IKÜM
artiklite 15, 18 ja 21 alusel. Palun selle läbivaatust ja
vastavate meetmete võtmist ajaraamis, mis on sätestatud IKÜM
artiklis 12(3) (üks kuu taotluse saamisest).

Palun teavitage taotluse kättesaamisest.

Lugupidamisega
Gregor Eesmaa

--

Dear Data Controller of example.com,

I am contacting you to formally raise concerns under the General
Data Protection Regulation (GDPR) regarding what I believe to
be the unlawful processing of my personal data by example.com,
potentially leading to its disclosure to unauthorised parties
without a clear legal basis.

Please find attached a formal application exercising my rights
under Articles 15, 18, and 21 of the GDPR. I request that
you review and act upon it accordingly, within the timeframe
established by Article 12(3) of the GDPR (one month from
receipt).

Please confirm receipt of this request.

Kind regards,
Gregor Eesmaa
```

I. Example GDPR Request

18. august 2025

Lugupeetud `example.com` vastutav andmetöötleja

Kirjutan teile andmesubjektina EL isikuandmete kaitse üldmääruse ("IKÜM") raames, seoses minu isikuandmete töötlemisega teenuses `example.com`.

Kuupäeval 2025-03-01, teavitati teid kasutajakontode loendamise turvanõrkusest (*account enumeration vulnerability*), mis **lubab selleks volitamata isikutel kinnitada, kas konkreetse meiliaadressiga kasutaja on teie teenuses registreeritud**. Tänase seisuga näib, et see turvanõrkus on lahenduseta sisse logimise vormil, parooli lähtestamise vormil, kasutaja registreerimise vormil ja meiliaadressi vahetamise vormil (vt lisatud aruannet).

Minu isikuandmeid – täpsemalt fakti, et mul on kasutaja teie veebiteenuses – töödeldakse eelnevalt mainitud vormides viisil, mis võimaldab selle avaldamist selleks volitamata isikutele. Ma ei ole teadlik ühestki IKÜM artiklis 6 kirjeldatud seaduslikust alusest, mis lubaks sellist töötlemist, ega ole andnud selget nõusolekut sellisel kujul isikuandmete töötlemiseks ja võimalikuks avaldamiseks.

Seetõttu esitan järgnevad ametlikud taotlused:

1. IKÜM artikli 15 kohaselt taotlen, et esitaksite eesmärgi, viidates IKÜM artikli 6 järgi kehtivale õiguslikule alusele, mille alusel töödeldakse minu (eelnevalt kirjeldatud) andmeid viisil, mis muudab need haavatavaks kasutajakontode loendamisele selleks volitamata isikutele (s.t., avaldades, kas minu meiliaadress on teenuses registreeritud).
2. IKÜM artikli 18 kohaselt nõuan piirangut oma isikuandmete töötlemisele viisil, mis on haavatav kasutajakontode loendamisele (st, avaldades, kas minu meiliaadress on teenuses registreeritud). Väidan, et selline isikuandmete töötlemine on ebaseaduslik. Samuti olen vastu oma isikuandmete kustutamisele.
3. IKÜM artikli 21 kohaselt esitan vastuväite minu isikuandmete igasugusele töötlemisele eesmärgiga avaldada kasutajakonto olemasolu selleks volitamata isikutele, eelkõige kui sellist töötlemist võidakse põhjendada töötleja õigustatud huvina. Selles kontekstis kaaluvad sellise huvi üles minu huvid, põhiõigused- ja vabadused isikuandmete kaitsele ja privaatsusele.

Palun vastake neile taotlustele e-posti teel ühe kuu jooksul, nagu on sätestatud IKÜM artiklis 12(3), ja täpsustage võetud meetmeid nende täitmiseks. Juhul, kui te ei suuda esitada rahuldavat vastust, mis käsitleb neid muresid ja parandab igasuguse ebaseadusliku töötlemise nõutud ajavahemiku jooksul, esitan vastavalt IKÜM artiklile 77 kaebuse asjakohasele järelvalveasutusele (Andmekaitse Inspektsioon).

Lugupidamisega

Gregor Eesmaa
isikukood: [personal code]
meiliaadress: [personal email address]
/ allkirjastatud digitaalselt /

18th August 2025

Dear Data Controller of `example.com`,

I am writing to you as a data subject under the EU General Data Protection Regulation ("GDPR") regarding the processing of my personal data by `example.com`.

On 2025-03-01, you were notified of account enumeration security vulnerability that **allows unauthorised parties to verify whether a user with a specific email address is registered with your service**. As of today, this vulnerability appears to remain unresolved on the login form, password reset form, account registration form and email change form (see attached report).

My personal data – specifically, the fact that I hold an account with your online service – is being processed in the previously mentioned forms in a manner that allows its disclosure to any unauthorised party. I am not aware of any legal basis under GDPR Article 6 that would allow such processing, nor have I provided explicit consent for this specific form of processing and potential disclosure.

Accordingly, I make the following formal requests:

1. Pursuant to Article 15 of GDPR, I request you provide the purpose, citing a valid legal basis under GDPR Article 6, for processing my personal data (as described above) in a manner that is vulnerable to account enumeration by unauthorised parties (i.e., revealing whether my email address is registered within the service).
2. Pursuant to Article 18 of GDPR, I request the restriction of my personal data being processed in a manner vulnerable to account enumeration (i.e., revealing whether my email address is registered within the service). I contest the lawfulness of this specific processing. I also oppose erasure of my personal data.
3. Pursuant to Article 21 of GDPR, I object to any processing of my personal data for the purpose of revealing account existence to any unauthorised party, particularly where such processing might be claimed under legitimate interests. My interests, fundamental rights and freedoms to data protection and privacy override such interests in this context.

Please respond to these requests over email within the statutory time limit of one month, as per Article 12(3) of GDPR, and confirm the measures taken to comply with it. In case of failure to provide a satisfactory response, addressing these concerns and rectifying any unlawful processing within the required timeframe, I will file a formal complaint with the competent supervisory authority (Andmekaitse Inspektsioon) in accordance with Article 77 of GDPR.

Kind regards,

Gregor Eesmaa
personal code: [personal code]
email address: [personal email address]
/ signed digitally /

J. Analysis Results

General **Re.** Reassessment details Service was reassessed Service has improvements

— Untested or irrelevant

Form-related columns

S Sign-in vulnerability (3 cols) **P** Password reset vulnerability (4 cols)

R Registration vulnerability (5 cols) **E** Email change vulnerability (5 cols)

Anti-bot measures Notification measures

δt Time difference of successful and unsuccessful query (ms)

Vulnerability indicators

Vulnerable: error message Vulnerable: visual differences

Vulnerable: response differences Not vulnerable

Form anti-bot measures

No CAPTCHA Has CAPTCHA

Notification measures

Unsafe validation order Safe validation order Safe, but irrelevant No email

Email received Email received, but irrelevant

J.1 Results of First Analysis

Table 4. Initial analysis results of the services tested

Service	S	δt	P	δt	R	δt	E	δt
aboutyou.ee	✓ –	64	⚠ ✗ ☰	88	⚠ ✓ ☰ 🔕	494	⚠ ✗ ☰ 🔕	88
aeromotors.ee	✓ –	32	⚠ ✗ ☰	21	⚠ ✗ ! 🔔	13	⚠ ✗ ! 🔕	31
amoremi.ee	✓ –	10	◐ ✗ ☰	156	⚠ ✗ ! 🔔	14	⚠ ✗ ! 🔔	154
apolloklubi.ee	✓ –	69	✓ – –	0	⚠ ✗ ☰ 🔔	9767	⚠ ✗ ☰ 🔔	9744
aripaev.ee	✓ –	85	✓ – –	10	⚠ ✓ ☰ 🔕	342	– – – –	–
barbora.ee	✓ –	365	✓ – –	5	⚠ ✗ ! 🔔	1309	– – – –	–
bet365.ee	✓ –	40	⚠ ✗ !	5	⚠ ✗ ! 🔕	23	⚠ ✗ ☰ 🔕	33
Circle K app	◐ –	–	⚠ – ☰	–	– – – –	–	⚠ – ☰ 🔕	–
city24.ee	⇄ ✗	194	⚠ ✗ ☰	50	⚠ ✗ ☰ 🔔	338	⚠ ✗ ☰ 🔔	81
cvkeskus.ee	⚠ ✗	1823	⚠ ✗ ☰	1078	⚠ ✗ ☰ 🔔	239	⚠ ✗ ! 🔔	106
delfi.ee	✓ –	85	✓ – –	4	⚠ ✓ ☰ 🔔	584	⚠ ✗ ! 🔕	591
ekool.eu	✓ –	46	⚠ ✗ ☰	13	⚠ ✗ ☰ 🔔	12	⚠ ✗ ☰ 🔔	24
eliis.eu	✓ –	77	⚠ ✗ ☰	294	⚠ ✗ ☰ 🔔	391	⚠ ✗ ☰ 🔔	162
Elisa Raamat app	✓ –	–	✓ – – –		⚠ – ! 🔔 –		– – – –	–
elron.ee	✓ –	77	⚠ ✗ ☰	626	⚠ ✗ ☰ 🔔	536	– – – –	–
euronics.ee	✓ –	1	⚠ ✓ ☰	212	⚠ ✗ ! 🔔	0	⚠ ✗ ☰ 🔔	129
flirtic.ee	✓ –	2	⚠ ✗ ☰	120	⚠ ✓ ! 🔔	553	◐ ✗ ! 🔔	226
forum.ee	✓ –	15	⚠ ✓ ☰	876	⚠ ✗ ! 🔔	67	⚠ ✗ ☰ 🔔	680
fv.ee	✓ –	92	⚠ ✗ ☰	37	⚠ ✗ ☰ 🔔	533	– – – –	–
geenius.ee	⚠ ✗	537	⚠ ✗ ☰	23	⚠ ✗ ☰ 🔔	159	– – – –	–
go3.tv	✓ –	998	⚠ ✓ ☰	273	⚠ ✓ ! 🔔	882	⚠ ✗ ☰ 🔕	486
harid.ee	✓ –	158	✓ – –	38	– – – – –		⚠ ✓ ! 🔔	2
hind.ee	✓ –	60	⚠ ✗ ☰	886	⚠ ✗ ! 🔕	64	– – – –	–
iha.ee	✓ –	7	✓ – –	1	⚠ ✓ ! 🔔	1	– – – –	–
ikea.ee	✓ –	274	⚠ ✓ ☰	249	⚠ ✗ ☰ 🔔	181	⚠ ✗ ☰ 🔔	252
inforegister.ee	✓ –	3	⚠ ✗ ☰	1264	⚠ ✗ ☰ 🔔	2087	– – – –	–
jpg.ope.ee	⚠ ✗	108	– – – –		– – – – –		– – – –	–
jupiter.err.ee	✓ –	76	⚠ ✗ ☰	316	⚠ ✗ ! 🔕	74	⚠ ✗ ! 🔕	35
jysk.ee	✓ –	60	✓ – –	329	⚠ ✓ ☰ 🔔	463	– – – –	–
k-rauta.ee	⇄ ✗	112	✓ – –	146	⚠ ✗ ! 🔔	592	– – – –	–
kae.edu.ee	✓ –	80	– – – –		⚠ ✗ ☰ 🔕	445	– – – –	–
kaup24.ee	✓ –	22	✓ – –	50	⚠ ✗ ! 🔔	2	⚠ ✗ ! 🔔	26
kava.ee	– – –		– – – –		⚠ ✗ ! 🔕	52	– – – –	–
kuldnebors.ee	✓ –	26	✓ – –	176	⚠ ✗ ! 🔔	235	– – – –	–
kv.ee	✓ –	918	✓ – –	25	⚠ ✗ ! 🔔	1017	– – – –	–
LIDL Plus app	⚠ –	–	⚠ – !	–	⚠ – ! 🔔	–	⚠ – ! 🔔	–
Maxima Estonia app	✓ –	–	⚠ – !	–	◐ – ! 🔔	–	✓ – – –	–
moodle.edu.ee	✓ –	2	✓ – –	39	⚠ ✓ ! 🔔	134	⚠ ✗ ! 🔔	8
nami-nami.ee	✓ –	5	⚠ ✗ ☰	39	⚠ ✗ ! 🔔	76	– – – –	–
notino.ee	✓ –	101	✓ – –	68	⚠ ✓ ☰ 🔔	888	⚠ ✗ ☰ 🔔	146
ohtuleht.ee	✓ –	90	✓ – –	19	⚠ ✓ ☰ 🔔	251	⚠ ✗ ! 🔕	26
okidoki.ee	✓ –	127	⚠ ✗ ☰	76	⚠ ✓ ! 🔔	348	⚠ ✗ ☰ 🔔	92
olybet.ee	✓ –	282	◐ ✓ ☰	63	⚠ ✓ ! 🔔	1034	⚠ ✗ ! 🔔	929
opiq.ee	✓ –	26	⚠ ✓ ☰	619	⚠ ✓ ☰ 🔔	589	⚠ ✗ ☰ 🔕	198
paavlikaltsukas.ee	✓ –	2	⚠ ✗ ☰	16	⚠ ✗ ! 🔕	2	⚠ ✗ ! 🔕	4
peaasi.ee	✓ –	81	⚠ ✗ ☰	74	– – – – –		– – – –	–
piletilevi.ee	✓ –	81	✓ – –	8	⚠ ✗ ☰ 🔔	142	⚠ ✗ ☰ 🔔	8
postimees.ee	✓ –	84	✓ – –	6	⚠ ✓ ☰ 🔔	322	⚠ ✗ ! 🔕	36
ResQ Club app	✓ –	–	⚠ – ☰	–	◐ – ☰ 🔔	–	⚠ – ☰ 🔔	–
rimi.ee	✓ –	148	– – – –		⚠ ✓ ☰ 🔔	5348	⚠ ✗ ☰ 🔔	350
solnet.ee	✓ –	4	⚠ ✗ ☰	20	⚠ ✓ ☰ 🔔	62	– – – –	–
soov.ee	✓ –	0	⇄ ✗ ☰	598	⚠ ✓ ! 🔔	440	– – – –	–
stena.ee	– – –		⚠ ✗ ☰	158	⚠ ✗ ! 🔔	192	⚠ ✗ ! 🔕	7
upload.ee	– – –		⚠ ✗ ☰	233	⚠ ✓ ! 🔔	359	– – – –	–
zalando.ee	◐ ✗	–	– – – –		– – – – –		⚠ ✗ ☰ 🔕	131

J.2 Results After Reassessment

Table 5. Second analysis results of the services tested

Service	Re.	S	δt	P	δt	R	δt	E	δt
aboutyou.ee			166		783		628		303
aeromotors.ee			24		10		5		20
amoremi.ee			5		245		15		166
apolloklubi.ee			12		0		9702		866
aripaev.ee			8		2		841		
barbora.ee			314		7		904		
bet365.ee			100		13		78		182
Circle K app									
city24.ee			12		333		259		34
cvkeskus.ee			1548		614		579		200
delfi.ee			79		41		589		40
ekool.eu			1		44		2		20
eliis.eu			67		203		60		184
Elisa Raamat app									
elron.ee			51		439		993		
euronics.ee			6		266		81		121
flirtic.ee			2		198		107		752
forum.ee			82		481		28		970
fv.ee			89		148		781		
geenius.ee			180		91		321		
go3.tv			61		403		2876		522
harid.ee			40		24				49
hind.ee			69		941		85		
iha.ee					47		9		
ikea.ee			275		434		516		447
inforegister.ee			0		752		139		
jpg.ope.ee			138						
jupiter.err.ee			59		1021		24		44
jysk.ee			83		679		364		
k-rauta.ee			70		2140		69		
kae.edu.ee			36				367		
kaup24.ee			6		34		86		69
kava.ee							53		
kuldnebors.ee			48		217		144		
kv.ee			791		57		995		
LIDL Plus app									
Maxima Estonia app									
moodle.edu.ee			100		48		140		6
nami-nami.ee			267		16		47		
notino.ee			101		29		895		259
ohtuleht.ee			56		39		322		32
okidoki.ee			44		107		124		83
olybet.ee			366		771		687		763
opiq.ee			28		513		199		1
paavlikaltsukas.ee			0		9		1608		1
peaasi.ee			81		74				
piletilevi.ee			62		5		148		11
postimees.ee			199		12		567		45
ResQ Club app									
rimi.ee			10		456		4113		368
solnet.ee			3		67		52		
soov.ee			37		47		17		
stena.ee					124		1393		92
upload.ee					349		646		
zalando.ee									31

J.3 Results After Partial Second Reassessment

Table 6. State of the services tested after a partial third analysis

Service	Re.	S	δt	P	δt	R	δt	E	δt
aboutyou.ee			166		783		628		303
aeromotors.ee			24		10		5		20
amoremi.ee			67		21		7		126
apolloklubi.ee			161		1		1		1534
aripaev.ee			8		2		841		
barbora.ee			314		7		904		
bet365.ee			100		13		78		182
Circle K app									
city24.ee			82		109		127		17
cvkeskus.ee			1548		614		579		200
delfi.ee			79		41		589		40
ekool.eu			1		44		2		20
eliis.eu			67		203		60		184
Elisa Raamat app									
elron.ee			51		439		993		
euronics.ee			3		151		6		
flirtic.ee			2		198		107		752
forum.ee			82		481		28		970
fv.ee			3		113		407		
geenius.ee			305		22		561		
go3.tv			61		403		2876		522
harid.ee			40		24				49
hind.ee					921		83		
iha.ee					47		9		
ikea.ee			275		434		516		447
inforegister.ee			0		752		139		
jpg.ope.ee			138						
jupiter.err.ee			59		1021		24		44
jysk.ee			83		679		364		
k-rauta.ee			70		2140		69		
kae.edu.ee			36				367		
kaup24.ee			6		34		86		69
kava.ee							53		
kuldnebors.ee			48		217		144		
kv.ee			965		3		864		
LIDL Plus app									
Maxima Estonia app									
moodle.edu.ee			100		48		140		6
nami-nami.ee			267		16		47		
notino.ee			101		29		895		259
ohtuleht.ee			56		39		322		32
okidoki.ee			44		107		124		83
olybet.ee			366		771		687		763
opiq.ee			28		513		199		1
paavlikaltsukas.ee			0		9		1608		1
peaasi.ee			81		74				
piletilevi.ee			62		5		148		11
postimees.ee			128		6		447		73
ResQ Club app									
rimi.ee			4		372		4327		353
solnet.ee			3		67		52		
soov.ee			37		47		17		
stena.ee					124		1393		92
upload.ee					349		646		
zalando.ee									31

J.4 Results After Partial Final Assessment

Table 7. State of the services tested after a partial final analysis

Service	Re.	S	δt	P	δt	R	δt	E	δt
aboutyou.ee			166		783		628		303
aeromotors.ee			24		10		5		20
amoremi.ee			67		21		7		126
apolloklubi.ee			161		1		1		1534
aripaev.ee			8		2		841		
barbora.ee			314		7		904		
bet365.ee			100		13		78		182
Circle K app									
city24.ee			82		109		127		17
cvkeskus.ee			2954		1456		562		149
delfi.ee			79		41		589		40
ekool.eu			8		7		1		33
eliis.eu			67		203		60		184
Elisa Raamat app									
elron.ee			51		439		993		
euronics.ee			4		88		1		6
flirtic.ee			2		198		107		752
forum.ee			59		677		22		903
fv.ee			18		103		833		
geenius.ee			51		35		477		
go3.tv			61		403		2876		522
harid.ee			338		113				22
hind.ee			46		890		54		
iha.ee					47		9		
ikea.ee			275		68		191		138
inforegister.ee			0		752		139		
jpg.ope.ee			88						
jupiter.err.ee			59		1021		24		44
jysk.ee			163		369		434		665
k-rauta.ee			122		4		283		
kae.edu.ee			36				367		
kaup24.ee			2		35		6		19
kava.ee							53		
kuldnebors.ee			48		217		144		
kv.ee			965		3		864		
LIDL Plus app									
Maxima Estonia app									
moodle.edu.ee			100		48		140		6
nami-nami.ee			1		226		8		
notino.ee			101		29		895		259
ohtuleht.ee			56		39		322		32
okidoki.ee			128		89		263		94
olybet.ee			722		642		1029		763
opiq.ee			28		513		199		1
paavlikaltsukas.ee			0		9		1608		1
peaasi.ee			81		74				
piletilevi.ee			57		32		134		0
postimees.ee			128		6		447		73
ResQ Club app									
rimi.ee			4		372		4327		353
solnet.ee			3		67		52		
soov.ee			37		47		17		
stena.ee					124		1393		92
upload.ee					618		803		
zalando.ee									31

K. Correspondence Log

A full archive of the correspondence with service providers is available for reference at https://gregoreesmaa.github.io/account-enumeration/correspondence/. This log excludes follow-ups that provided the unpacked contents of digitally signed containers.

Correspondence Source

Author's University Email Author's Personal Email

Communication Direction ("Dir.")

Outgoing Message Incoming Message

Language ("Lang.")

English Estonian

Legal Basis Cited Under GDPR Article 6(1) ("Basis")

(a) Consent (b) Contractual Necessity (f) Legitimate Interest

Justifications ("Justif.")

Common Practice User Experience Security Concerns Account Management Deflecting Accountability

Actions ("Act.")

Claims to have fixed Fix in progress Planned to fix Refuses complete fix Requests extension

Table 8. Correspondence with the service providers

Date	Dir.	Lang.	Summary
aboutyou.ee			
2025-03-16			Initial vulnerability report
2025-03-18			Confirmation of receipt; Explaining bug bounty program
2025-03-22			Follow-up
2025-03-24			Other; Explaining bug bounty program
2025-05-31			GDPR requests
2025-05-31			Automated reply
2025-05-31			Other; Asks to confirm account deletion
2025-05-31			Other; Generic compliance response

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
2025-05-31	→		Follow-up
2025-06-13	←		Other; Provides encrypted response
2025-06-13	←		Other; Provides one-time password to response
2025-06-21	→		Follow-up; Asking to send password again
2025-07-15	→		Follow-up; Asking to send password again
2025-07-15	←		Automated reply
2025-07-15	→		Follow-up; Asking to send response again
2025-07-16	←		Basis: (b) (f) ; Justif.: ; Act.:
2025-07-23	→		Follow-up
2025-08-11	←		Basis: (f) ; Justif.: ; Act.:
aeromotors.ee			
2025-03-16	→		Initial vulnerability report
2025-03-16	←		Automated reply
2025-03-23	→		Initial vulnerability report
2025-03-23	←		Automated reply
2025-05-31	→		GDPR requests
2025-06-02	←		Act.:
2025-06-16	←		Basis: (f) ; Justif.: ; Act.:
2025-06-21	→		Follow-up
amoremi.ee			
2025-03-16	→		Initial vulnerability report
2025-03-17	←		Act.:
2025-03-23	→		Initial vulnerability report
2025-05-18	→		Reassessment report
apolloklubi.ee			
2025-03-16	→		Initial vulnerability report
2025-03-18	←		Confirmation of receipt
2025-04-14	←		Act.:

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
2025-05-18	→		Reassessment report
Äaripaev.ee			
2025-03-16	→		Initial vulnerability report
2025-03-16	←		Automated reply
2025-03-23	→		Initial vulnerability report
2025-05-31	→		GDPR requests
2025-05-31	←		Automated reply
2025-05-31	←		Automated reply
2025-06-30	←		Basis: (a); Justif.: ; Act.:
2025-07-17	→		Follow-up
2025-08-06	←		Basis: (a); Justif.: ; Act.:
Bbarbora.ee			
2025-03-16	→		Initial vulnerability report
2025-03-23	→		Initial vulnerability report
2025-03-23	←		Automated reply
2025-03-23	←		Confirmation of receipt; Promised to contact, if interested
2025-05-31	→		GDPR requests
bet365.ee			
2025-03-16	→		Initial vulnerability report
2025-03-23	→		Initial vulnerability report
2025-05-31	→		GDPR requests
2025-05-31	←		Other; Unable to open ASICe container
2025-05-31	→		GDPR requests
Circle K			
2025-04-06	→		Initial vulnerability report
2025-04-06	←		Automated reply
2025-04-07	←		Act.:
2025-06-21	←		Automated reply

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
2025-06-21	→	EN ET	GDPR requests
2025-06-25	←	ET	Confirmation of receipt
2025-07-03	←	ET	Confirmation of receipt; Not sure why this was sent
2025-07-18	←	EN ET	Basis: (b); Justif.: ; Act.:
2025-07-20	→	EN ET	Follow-up; Confirmation of receipt; Accept disabling email notifications
2025-07-23	→	ET	Follow-up
2025-07-24	←	EN ET	Act.:
2025-07-30	←	EN ET	Act.:
city24.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-23	→	EN	Initial vulnerability report
2025-03-28	←	ET	Act.:
2025-05-18	→	EN	Reassessment report
cvkeskus.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-23	→	EN	Initial vulnerability report
2025-05-31	→	EN ET	GDPR requests
2025-06-30	←	ET	Act.:
2025-07-23	←	ET	Basis: (f); Justif.: ; Act.:
delfi.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-16	←		Automated reply
2025-03-19	←	ET	Confirmation of receipt
2025-05-31	→	EN ET	GDPR requests
2025-05-31	←		Automated reply
2025-06-27	←	ET	Basis: (f); Justif.: ; Act.: ; Combined response by delfi.ee and ohtuleht.ee
ekool.eu			

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
2025-04-27	→	EN	Initial vulnerability report
2025-04-27	←		Automated reply
2025-05-05	←	EN	Confirmation of receipt
2025-05-19	←	EN	Justif.: ; Act.:
2025-05-19	→	EN	Follow-up; Asking for informal reasoning
2025-06-22	→	EN ET	GDPR requests
2025-06-22	←		Automated reply
2025-06-25	←	ET	Confirmation of receipt
2025-07-11	←	EN	Basis: (f) ; Justif.: ; Act.:
eliis.eu			
2025-04-27	→	EN	Initial vulnerability report
2025-04-28	←	ET	Act.:
2025-06-01	→	EN	Fix confirmation
Elisa Raamat			
2025-04-06	→	EN	Initial vulnerability report
2025-06-21	→	EN ET	GDPR requests
2025-06-21	←		Automated reply
2025-06-21	→	EN ET	GDPR requests
2025-06-25	←	ET	Confirmation of receipt
2025-07-03	←	ET	Justif.: ; Act.:
elron.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-23	→	EN	Initial vulnerability report
2025-05-31	→	EN ET	GDPR requests
2025-06-02	←	ET	Confirmation of receipt
euronics.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-23	→	EN	Initial vulnerability report
2025-04-21	←	EN	Act.:

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
2025-05-18	→	EN	Reassessment report
2025-06-24	→	EN ET	GDPR requests
2025-07-11	←	ET	Justif.: ☻ ↱; Act.: ⟳
flirtic.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-23	→	EN	Initial vulnerability report
2025-05-31	→	EN ET	GDPR requests
2025-06-03	←	ET	Justif.: 🌐 ↱; Act.: ⊗
2025-06-21	→	EN ET	Follow-up
2025-06-27	←	ET	Justif.: 🌐 👤🔒 ↱; Act.: ⊗
forum.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-22	←	ET	Justif.: 🌐; Act.: ⊗
2025-03-22	→	EN	Follow-up
2025-05-31	→	EN ET	GDPR requests
2025-06-29	←	ET	Basis: (b) (f); Justif.: 🛡 👤🔒; Act.: ⊗
fv.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-16	←		Automated reply
2025-03-16	←	EN	Confirmation of receipt
2025-06-24	→	EN ET	GDPR requests
2025-06-24	←		Automated reply
2025-07-22	←	EN ET	Act.: 📅; Deleted account
2025-07-23	→	EN	Follow-up
2025-07-24	←	EN	Other; Apologies for deleting account
geenius.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-18	←	ET	Act.: 📅
2025-05-18	→	EN	Reassessment report

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
2025-06-24	→	EN ET	GDPR requests
2025-06-26	←	ET	Act.:
go3.tv			
2025-03-16	→	EN	Initial vulnerability report
2025-03-23	→	EN	Initial vulnerability report
2025-05-31	→	EN ET	GDPR requests
2025-06-05	←	ET	Confirmation of receipt
harid.ee			
2025-03-23	→	EN	Initial vulnerability report
2025-05-31	→	EN ET	GDPR requests
2025-06-16	←	ET	Confirmation of receipt
2025-06-27	←	ET	Act.:
hind.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-20	←	EN	Act.:
2025-05-18	→	EN	Reassessment report
2025-06-04	←	EN	Act.:
2025-06-24	→	EN ET	GDPR requests
2025-07-17	←	EN	Justif.: ; Act.:
iha.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-06-24	→	EN ET	GDPR requests
ikea.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-16	←		Automated reply
2025-03-23	→	EN	Initial vulnerability report
2025-05-31	→	EN ET	GDPR requests
2025-06-03	←	EN	Confirmation of receipt

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
2025-06-30	→	EN	Basis: (b); Justif.: ; Act.:
inforegister.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-16	←		Automated reply
2025-03-16	←		Automated reply
2025-05-31	→	EN	Fix confirmation
jpg.ope.ee			
2025-04-27	→	EN	Initial vulnerability report
2025-04-29	←	ET	Justif.: ; Act.:
2025-05-01	→	EN ET	Follow-up
2025-06-01	→	EN ET	GDPR requests
2025-06-02	←	ET	Act.:
2025-06-17	←	ET	Justif.: ; Act.:
2025-06-21	→	EN ET	Follow-up
jupiter.err.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-23	→	EN	Initial vulnerability report
2025-03-23	←		Automated reply
2025-05-31	→	EN ET	GDPR requests
jysk.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-23	→	EN	Initial vulnerability report
2025-03-23	←		Automated reply
2025-04-01	←	EN	Confirmation of receipt
2025-05-31	→	EN ET	GDPR requests
2025-06-12	←	EN ET	Confirmation of receipt; Asks for informal consultation
2025-06-21	→	EN ET	Follow-up

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
2025-07-01	←		Basis: (f) ; Justif.: ; Act.:
k-rauta.ee			
2025-03-16	→		Initial vulnerability report
2025-03-16	←		Automated reply
2025-05-31	→		GDPR requests
2025-06-25	←		Basis: (b) (f) ; Justif.: ; Act.:
kae.edu.ee			
2025-03-16	→		Initial vulnerability report
2025-03-23	→		Initial vulnerability report
2025-05-31	→		GDPR requests
2025-06-01	←		Act.:
2025-06-01	←		Act.:
2025-06-04	→		Follow-up; Confirmation of proposed solution
kaup24.ee			
2025-03-16	→		Initial vulnerability report
2025-03-16	←		Automated reply
2025-03-23	→		Initial vulnerability report
2025-03-23	←		Automated reply
2025-05-31	→		GDPR requests
2025-06-04	←		Confirmation of receipt
2025-06-30	←		Basis: (b) (f) ; Justif.: ; Act.:
kava.ee			
2025-03-16	→		Initial vulnerability report
2025-03-23	→		Initial vulnerability report
2025-05-31	→		GDPR requests
kuldnebors.ee			
2025-03-16	→		Initial vulnerability report
2025-03-23	→		Initial vulnerability report

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
2025-05-18	→	EN	Fix confirmation
kv.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-23	→	EN	Initial vulnerability report
2025-05-18	→	EN	Reassessment report
2025-05-27	←	EN	Justif.: ☻ 🛡; Act.: 📅
2025-06-24	→	EN ET	GDPR requests
2025-07-16	←	ET	Justif.: ↱; Act.: ⊗
LIDL Plus			
2025-04-06	→	EN	Initial vulnerability report
2025-04-06	←	EN	Other; Non-Delivery Report: invalid forwarding address
2025-04-06	←	EN	Other; Non-Delivery Report: invalid forwarding address
2025-04-06	←	EN	Other; Non-Delivery Report: invalid forwarding address
2025-04-27	→	EN	Initial vulnerability report
2025-04-28	←	ET	Confirmation of receipt
2025-05-07	←	ET	Justif.: 🌍 ☻ 👤🔒; Act.: ⊗
2025-06-21	→	EN ET	GDPR requests
2025-06-21	→	EN ET	GDPR requests
2025-06-26	←	ET	Confirmation of receipt
2025-07-08	←	EN ET	Justif.: ↱; Act.: ⊗
Maxima Estonia			
2025-04-06	→	EN	Initial vulnerability report
2025-04-27	→	EN	Initial vulnerability report
2025-04-27	←		Automated reply
2025-06-21	→	EN ET	GDPR requests
2025-07-02	←	ET	Confirmation of receipt

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
2025-07-21	←		Act.:
2025-08-01	←		Justif.: ; Act.:
moodle.edu.ee			
2025-03-16	→		Initial vulnerability report
2025-03-16	←		Automated reply
2025-03-18	←		Confirmation of receipt
2025-05-31	→		GDPR requests
2025-05-31	←		Automated reply
2025-06-02	←		Confirmation of receipt
nami-nami.ee			
2025-03-16	→		Initial vulnerability report
2025-03-23	→		Initial vulnerability report
2025-05-31	→		GDPR requests
2025-06-02	←		Act.:
2025-06-20	←		Act.:
2025-06-21	→		Follow-up
notino.ee			
2025-03-16	→		Initial vulnerability report
2025-03-16	←		Automated reply
2025-05-31	→		GDPR requests
ohtuleht.ee			
2025-03-16	→		Initial vulnerability report
2025-03-23	→		Initial vulnerability report
2025-03-24	←		Justif.: ; Act.:
2025-03-24	→		Follow-up
2025-05-31	→		GDPR requests
2025-06-27	←		Basis: (f) ; Justif.: ; Act.: ; Combined response by delfi.ee and ohtuleht.ee

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
okidoki.ee			
2025-03-16	→		Initial vulnerability report
2025-03-16	←		Automated reply
2025-03-18	←		Confirmation of receipt
2025-05-31	→		GDPR requests
2025-06-02	←		Confirmation of receipt
2025-06-26	←		Basis: (b); Justif.: ; Act.:
olybet.ee			
2025-03-16	→		Initial vulnerability report
2025-03-17	←		Confirmation of receipt
2025-03-23	→		Initial vulnerability report
2025-06-08	→		Reassessment report
2025-06-24	→		GDPR requests
2025-07-01	←		Confirmation of receipt
2025-07-23	←		Other; Update to Privacy Policy
2025-07-23	←		Justif.: ; Act.:
opiq.ee			
2025-03-16	→		Initial vulnerability report
2025-03-17	←		Confirmation of receipt
2025-05-18	→		Fix confirmation
paavlikaltsukas.ee			
2025-03-16	→		Initial vulnerability report
2025-03-23	→		Initial vulnerability report
2025-05-31	→		GDPR requests
peaasi.ee			
2025-04-27	→		Initial vulnerability report
2025-04-27	←		Automated reply
2025-06-08	→		GDPR requests

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
2025-06-08			Automated reply
2025-06-10			Confirmation of receipt
piletilevi.ee			
2025-03-16			Initial vulnerability report
2025-03-23			Initial vulnerability report
2025-05-31			GDPR requests
2025-06-30			Justif.: ; Act.:
postimees.ee			
2025-03-16			Initial vulnerability report
2025-03-17			Confirmation of receipt
2025-04-04			Confirmation of receipt; Combined response by soov.ee and postimees.ee
2025-05-18			Reassessment report
2025-06-24			GDPR requests
2025-06-24			GDPR requests
2025-07-24			Basis: (b) (f) ; Justif.: ; Act.:
ResQ Club			
2025-04-06			Initial vulnerability report
2025-04-27			Initial vulnerability report
2025-06-09			Other; Update to Privacy Policy
2025-06-21			GDPR requests
2025-06-21			GDPR requests
2025-06-24			Act.:
rimi.ee			
2025-03-16			Initial vulnerability report
2025-03-16			Automated reply; Non-Delivery Report: invalid email address
2025-03-23			Initial vulnerability report
2025-03-23			Automated reply

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
2025-04-04			Confirmation of receipt
2025-06-08			Reassessment report
2025-06-24			GDPR requests
2025-06-24			Automated reply
2025-07-23			Basis: (f) ; Justif.: ; Act.:
solnet.ee			
2025-03-16			Initial vulnerability report
2025-05-31			GDPR requests
2025-05-31			GDPR requests
soov.ee			
2025-03-16			Initial vulnerability report
2025-03-17			Confirmation of receipt
2025-03-23			Initial vulnerability report
2025-04-04			Confirmation of receipt; Combined response by soov.ee and postimees.ee
2025-05-18			Fix confirmation
stena.ee			
2025-03-16			Initial vulnerability report
2025-05-31			GDPR requests
2025-05-31			GDPR requests
upload.ee			
2025-03-16			Initial vulnerability report
2025-03-16			Justif.: ; Act.:
2025-03-16			Follow-up
2025-03-16			Justif.: ; Act.:
2025-05-31			GDPR requests
2025-05-31			Justif.: ; Act.:
2025-05-31			Follow-up
2025-06-01			Justif.: ; Act.:

Continued on next page

Table 8 – continued from previous page

Date	Dir.	Lang.	Summary
2025-06-04	→	EN/ET	Follow-up
2025-06-05	←	ET	Act.: ✓
2025-06-08	→	EN/ET	Follow-up
2025-06-09	←	ET	Act.: ✓
zalando.ee			
2025-03-16	→	EN	Initial vulnerability report
2025-03-16	←		Automated reply
2025-05-31	→	EN/ET	GDPR requests
2025-06-02	←	EN/ET	Confirmation of receipt
2025-06-30	←	EN	Justif.: ↱; Act.: ✗
2025-07-17	→	EN	Follow-up

License

Non-exclusive licence to reproduce the thesis and make the thesis public

I, Gregor Eesmaa,

1. grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the digital archives of the University of Tartu until the expiry of the term of copyright, my thesis **Account Existence Leaks in Estonian Online Services**, supervised by **Arnis Paršovs (PhD)**;
2. grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright;
3. am aware of the fact that the author retains the rights specified in points 1 and 2;
4. confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Gregor Eesmaa
12/08/2025