Pakistan Journal of Life and Social Sciences
www.pjlss.edu.pk

https://doi.org/10.57239/PJLSS-2026-24.1.0037

RESEARCH ARTICLE

A Multifactorial Method for Analyzing Web Links in a Counter-Phishing System Based on Social Engineering

Anna Poluyan[1]*, Sofya Petrenkova[2], Kseniia Korovina[3]

[1,2,3]Don State Technical University, Rostov-On-Don, Russian Federation

ARTICLE INFO

Received: APR 20, 2026

Accepted: MAY 18, 2026

Keywords

Social Engineering
Phishing
URL
Links
Information Security

***Corresponding Author:**
apoluyan@donstu.ru

ABSTRACT

The relevance of this research is driven by the steady increase in the number of phishing attacks utilizing social engineering techniques to bypass traditional security measures. According to APWG data, over 1.2 million phishing incidents are recorded monthly, with more than 60% of attacks using domains registered less than 24 hours before the attack. Under such conditions, existing solutions based on blacklists prove insufficiently effective. The aim of this work is to develop a comprehensive URL analysis method capable of identifying both known and previously unseen phishing resources based on a combination of reputational, structural, and infrastructural features. The article provides an overview of modern anti-phishing methods, including lexical-structural analysis of domain names, SSL certificate validation, DNS and WHOIS data analysis, and web page content analysis. Based on this analysis, the architecture of a software module for link checking is proposed, implemented on the principle of multi-level sequential verification with early termination. The module integrates data from the Google Safe Browsing API, performs heuristic analysis of the domain for typosquatting and homograph attacks, verifies certificate validity, domain age, and the presence of DNS records. Testing of the developed module on a set of test URLs confirmed its ability to correctly classify legitimate, suspicious, and phishing resources. A key advantage of the solution is its complete autonomy: analysis is performed on the user's side without transmitting data to external networks, and web content is checked without executing JavaScript, ensuring the security of the analysis tool itself. The developed module can be used as a standalone security tool or as a component of comprehensive information security systems.

INTRODUCTION

Social engineering in the field of information security represents a targeted activity by an attacker aimed at unauthorized access to protected information, information systems (IS), critically important resources, or trusted privileges through psychological manipulation of access subjects – individuals who are legitimate users of the IS. According to data from the Russian holding "Informzaschita," more than 80% of attacks begin with a phishing email, and according to Kaspersky Lab, 30% of studied users clicked on a malicious link in an email (Informzashchita, 2025; Kaspersky, 2023). Unlike classical technical attacks based on exploiting software vulnerabilities (CVE), network equipment configuration flaws, or cryptographic weaknesses, social engineering attacks bypass technical protection mechanisms by directly impacting the human factor – the least predictable and formalizable component of an information security system.

Review of Existing Sources

Social engineering methods can be grouped according to the following features:

1. By delivery method:

- Email: mass and targeted phishing campaigns with fake HTML forms, attachments (malicious macros, executable files), or links to fake websites;
- Mobile channels: SMS, messengers, push notifications;
- Voice communication: phone calls using social pretexts ("your card is blocked," "the tax service requires confirmation");
- Physical interaction: leaving USB devices in public areas, infiltrating premises disguised as technical personnel;

2. by level of personalization:

- Mass phishing: non-targeted mass attacks with low conversion but high reach;
- Spear phishing: attacks using data from open sources (OSINT), leaks (e.g., via HaveIBeenPwned), internal databases – to impersonate a trusted sender;
- Whaling: highly targeted attacks on executives, often involving forgery of corporate style, personal addresses, and context of current tasks;

3. by impact goal:

- Theft of authentication data (login/password, OTP, PIN);
- Initiation of financial transfers;
- Delivery of malicious software (Trojans, ransomware, info-stealers);
- Obtaining intelligence information (IT infrastructure structure, employee names, internal regulations).

Social engineering threats possess the following properties:

- Low detectability by technical means: messages contain no malicious code at the time of sending, use legitimate domains or encryption, reducing the effectiveness of DLP, email filters, and NGFW;
- High degree of adaptability: attackers quickly change scenarios, email templates, and domain names, evading signature-based rules;
- Dependence on the human factor: attacks use universal psychological triggers – fear, urgency, curiosity, trust in authority – making them resistant to changes in IT infrastructure.

When formalizing a threat model, social engineering relates to (Federal Service for Technical and Export Control (FSTEC of Russia), 2021):

1. threats to confidentiality (unauthorized access to information);
2. threats to integrity (making changes on behalf of a legitimate user);
3. threats to availability (e.g., initiating account lockouts through mass password reset requests).

Here, the threat source is an external attacker, and the implementation channel is social, not technical, which places these threats outside the scope of traditional perimeter security measures.

Thus, social engineering represents a systemic, multi-vector, and difficult-to-formalize threat requiring specialized countermeasures based not only on personnel training but also on automated analysis of communications for signs of manipulative influence – which determines the relevance of developing software tools capable of identifying and neutralizing such attacks at an early stage of user interaction (Duru, 2025; Hosseinzadeh et al., 2025; Informzashchita, 2025; Nair et al., 2025).

Phishing (from English phishing, sounding like fishing) is a targeted attack based on impersonating a trusted entity to induce the victim to perform an action leading to the compromise of confidential information, credentials, financial assets, or the information system as a whole. Within the threat model, phishing is classified as an attack via a data transmission channel involving human participation, where the primary exploitation vector is not a software vulnerability but the user's

insufficient critical perception of incoming communication. According to the classification adopted in the MITRE ATT&CK Enterprise Matrix (tactic TA0059 – Social Engineering, technique T1566 – Phishing), phishing includes both the supporting technical infrastructure (fake domains, servers, web forms) and the psychological scenario of interaction with the victim (MITRE ATT&CK Framework, n.d.). Unlike passive threats (e.g., traffic interception), phishing is an active attack requiring preparatory work from the attacker (reconnaissance, creation of fake resources) and synchronization with the target's behavior. Phishing attacks are divided into the following categories depending on the degree of personalization, target audience, and delivery channel used (Afanaseva et al., 2022; Batyushkin, 2021; Gorda, Chechulin, 2024; Tarasova, 2023):

1. Mass phishing. Characterized by sending identical or slightly varying messages to a large number of addresses without prior reconnaissance.

Key features:

- Use of general wording ("Dear Customer!", "Your account is blocked!");
- Links to third-level domains (e.g., secure-bank-login.ru.fake-portal.com);
- Use of templates imitating the interfaces of major banks, email services, government portals (Gosuslugi, FNS);
- Low level of technical complexity but high scalability; a large-scale attack can be conducted with minimal technical means;

2. Targeted phishing. Involves preliminary gathering of information about the victim (via OSINT, data leaks, social networks) to create a credible message. This type of phishing attack is characterized by a high degree of personalization, targeting specific individuals or organizations. Key features include:

- Adaptation to the victim's name, position, and core processes.
- Use of domain names with significant orthographic variations of legitimate brands (e.g., gosuslugi-ru.ru, sber-bank.ru).
- Attachments in PDF or DOCX formats with traditional macros or hyperlinks to critical resources in cloud storage (Google Drive, OneDrive), bypassing traditional anti-spam filters.
- Integration into BEC (Business Email Compromise) attack chains to initiate unauthorized payments.

3. Whaling. A specialized form of targeted phishing aimed at high-ranking individuals (CEOs, financial directors, chief accountants). It is distinguished by:

- Imitation of elegant style and business correspondence.
- Consideration of the context of current tasks (e.g., "urgently process payment under contract No. XXX").
- Propagation through internal communication channels, including hiding email addresses, managing in confidential mode through prior account compromise.

4. Vishing and Smishing. Besides email vectors, attackers use alternative channels (voice calls and SMS) to diversify attacks and reduce the effectiveness of standard protective measures:

- Vishing: Voice phishing using caller ID spoofing technologies via VoIP platforms. Attackers impersonate bank call centers or government institutions, persuading victims to disclose confidential data.
- Smishing: Sending SMS messages from short numbers or numbers disguised as official ones. Messages contain either phishing links or instructions to call back a fake "hotline" number.

5. Related techniques: pharming and homograph attacks - although sometimes considered separately from phishing, they are closely related and often used in combination: Pharming: Redirecting a user to a fake website by compromising DNS servers or modifying the

local hosts file on the victim's device.
Unlike phishing, no link click is required – the attack occurs at the infrastructure level.

Factors contributing to phishing's resilience against modern security measures:
Phishing campaigns demonstrate high effectiveness and low detectability for several reasons:

- Absence of malicious code at the delivery stage. Emails and SMS contain only text and links, carrying no executable files or macros. This makes them "invisible" to antivirus scanners and behavioral analysis systems (sandboxes) focused on signatures and suspicious activity.
- Mass use of HTTPS. Most phishing resources today are equipped with TLS certificates, including free ones from Let's Encrypt. The presence of a "padlock" in the browser creates a false sense of security among users and complicates detection at the network level.
- Dynamic content generation. Phishing pages can adapt to a specific victim: they display malicious content only under certain parameters (User-Agent, IP address, geolocation). When scanned by automated bots, such pages may return an empty or legitimate response, evading inclusion in reputation lists.
- High infrastructure update speed. Attackers constantly register new domains and change email templates faster than security vendors can update blacklists. The lack of established signatures makes classical signature-based methods ineffective.

Modern approaches to countering phishing attacks based on social engineering methods include a set of organizational measures, technical means, and educational initiatives. However, all have significant limitations, making them insufficient for ensuring reliable protection, especially in resource-constrained environments of small organizations and municipal institutions (Ahmad et al., 2024; Khaled, Rafik, 2025; NomuIIa, 2024). Technical means, such as secure email gateways and traffic filtering systems, are mainly focused on identifying technical anomalies: verifying sender authenticity using SPF, DKIM, and DMARC mechanisms, analyzing attachments in isolated environments, and blocking URLs based on reputation databases. However, these approaches prove ineffective against phishing campaigns using domains registered immediately before the attack and not yet blacklisted, as well as legitimate cloud platforms like Google Sites, Microsoft Forms, or GitHub Pages, which allow attackers to host phishing forms on trusted domains with valid TLS certificates. A critical gap in automated filtering systems is the category of text messages containing no attachments or links. Such emails execute an attack solely through social engineering methods: they either directly request confidential data in the message body or contain instructions to contact the fraudster by phone. Since such messages lack classical compromise indicators applicable for machine extraction, they are completely excluded from the detection perimeter of existing technical security measures (Elevate Security, Cyentia Institute, 2023).

Phishing attacks using fake web resources continue to be one of the most common and effective vectors for user compromise. According to APWG (Anti-Phishing Working Group) data for 2025, over 1.2 million unique phishing incidents are recorded monthly, with more than 60% using domains registered less than 24 hours before the attack and protected by valid SSL certificates 88. Under such conditions, traditional approaches based solely on blacklists or signature matching prove ineffective against targeted and zero-day attacks.

PROPOSED METHODS

To ensure proactive protection, the application of a comprehensive set of methods covering reputational, structural, infrastructural, and content levels of URL analysis is required. Let's consider the methods most suitable for implementation within a software tool for comprehensive phishing protection:

1. Reputational analysis based on aggregated databases. This method involves comparing the checked URL against known repositories of phishing and malicious resources.

The key sources of reputational data for identifying phishing threats are the following services (Hadžiosmanović et al., 2014): Google Safe Browsing API, PhishTank, OpenPhish, URLhaus, Netcraft.

Integration of these sources allows achieving accuracy at the level of 90%-95% for large-scale campaigns. However, the fundamental limitation is the time delay between a phishing asset going live and its indexing in databases. Attackers exploit this "window of vulnerability," which fully undermines reliance on reputational methods alone. Nevertheless, reputational analysis remains the first line of defense, ensuring prompt blocking of verified threats.

2. Analysis of the substitution structure. This method focuses on detecting phishing forms hosted on subdomains of legitimate but compromised resources (e.g., secure.bank.com.attacker-site.net). It involves comparison with a reference list of trusted domains, including fuzzy matching metrics:

- Jaccard coefficient: $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$
- Levenshtein distance: $d(c_1, c_2)$, minimizing the number of edit operations.

The threshold for minimum cost is set at $\geq 0{,}75$, minimizing false positives while maintaining high sensitivity.

3. Heuristic analysis of the query string and URL structure.

Beyond the domain name, the entire hyperlink structure is analyzed:

- Anomalies in spelling: detection of excessive special characters (hyphens), numerical sequences, direct IP address usage instead of a domain name, and extreme URL length (over 25 characters in the domain).
- Assessment of the domain zone: Zones offering free registration (.tk, .ml, .ga, .cf) are statistically considered unreliable, as they are frequently used for short-term phishing campaigns.

Address manipulations:

- Use of URL shorteners (bit.ly and similar), hiding the final destination.
- Presence of social engineering markers in query parameters (e.g., "urgent=true", "action=verify").
- Use of special characters (e.g., "@") to create misleading constructions.

Analysis is performed by parsing the URI according to RFC 3986 and comparing components with signatures of typical attacks. For the Russian-speaking segment, identifying Cyrillic triggers in the path or parameters ("вход" (login), "платеж" (payment), "блокировка" (blocking)) is critically important.

4. SSL/TLS certificate inspection.

The presence of HTTPS is no longer a guarantee of security; however, certificate metadata serves as an important diagnostic feature:

- Certificate freshness: Issue date less than 7 days ago – a marker of a potentially fraudulent resource.
- Issuer reliability: Self-signed certificates or certificates from little-known Certificate Authorities (CAs) are considered anomalous.
- Name validity: Mismatch between the certificate and the domain name (lack of exact match or valid wildcard) indicates poor attack preparation or technical incompetence.
- Cryptographic strength: Use of outdated algorithms (SHA-1) points to a technically backward or intentionally compromised infrastructure.

Verification is performed by establishing an SSL connection bypassing the system's trusted root certificate store to obtain objective data.

5. Infrastructural analysis (WHOIS/DNS).

Domain registration and delegation data indirectly indicate the owner's intentions:

- Domain age: Correlation analysis shows that the vast majority of phishing resources (up to 75%) exist for less than a month.
- Registration transparency: Use of Whois Privacy services or absence of registrant contact information increases the risk level.
- DNS zones: Absence of standard service records (SPF, MX), typical for corporate infrastructure, may indicate forgery.

To prevent cache poisoning or spoofing, queries are performed through several authoritative public resolvers (Cloudflare, Quad9, Google Public DNS).

6. Behavioral and content analysis (in a sandbox).

The final, but most resource-intensive stage, is analyzing the page content in a controlled environment (JavaScript disabled, external requests blocked):

- Linguistic analysis: Searching the text for imperative constructions aimed at urging immediate action.
- Obfuscation: Identifying interface elements hidden by CSS properties (display:none, transparency, off-screen positioning).
- Technical code: Detecting malicious iframes, automatic redirects, and fake login forms mimicking the design of major brands.

This method is used cautiously due to high infection risks and requires strict isolation.

Developing an effective anti-phishing tool requires moving away from narrowly specialized solutions towards a comprehensive, multi-level approach covering all possible delivery vectors and manifestations of social engineering. Based on the analysis of modern threats, methodological recommendations, and practices from real incidents, a set of methods was selected that collectively provide proactive protection at the stage preceding user interaction with the threat. Specifically, link analysis for signs of phishing was chosen as the first and most critical barrier, since the vast majority of phishing attacks culminate in a URL click. The method includes reputational check, lexical-structural domain analysis, SSL certificate validation, WHOIS and DNS analysis, and (optionally) content assessment. This multifactorial approach allows identifying both known and zero-day threats without relying on a single indicator that can be easily bypassed.

The link checking module is a key component of the software tool, designed for comprehensive assessment of a URL for signs of phishing activity. The module's architecture is built on the principle of multi-stage sequential verification, where the same link is analyzed by several independent methods: reputation, domain, SSL, URL structure, WHOIS, DNS, page content, and basic HTTPS check.

The module is built on a two-component scheme:

- PhishingURLChecker – the analytical core, encapsulating all checks and independent of the data presentation method.
- LinkCheckerWidget – a graphical shell based on PyQt6, providing URL input and visualization of a detailed report.
- The core accepts a URL string and a callback function print_func, through which it returns results. This implementation ensures versatility: the module can function both as part of a GUI and in console mode.

Early Termination Principle

To minimize response time and save computational resources, a cascade algorithm with immediate exit upon detecting a critical threat is implemented. The first stage is an appeal to the Google Safe Browsing API (v4). If the response contains a matches field, the check stops, and the module instantly returns a verdict of "PHISHING" with a risk level of 5/5.

Request details:

- method: HTTPS POST with JSON body;
- client ID: cyberguardian / 1.0;
- tracked threats: MALWARE, SOCIAL_ENGINEERING, UNWANTED_SOFTWARE;
- platform: ANY_PLATFORM;
- check object: URL.

This stage ensures lightning-fast blocking of resources already listed in the global reputation database. The main limitations are the need for network access and API quotas, which is acceptable in a local operation context.

Timeouts and Fault Tolerance

Network operations are performed synchronously but with strict waiting intervals (from 5 to 10 seconds depending on the request type). This prevents interface "freezing." In case of connection failures, analysis is not interrupted – the issue is recorded in the report (e.g., "DNS check error"), increasing the system's resilience to temporary unavailability of external services.

Heuristic Stage

If the reputational check does not reveal a threat, the module extracts the domain (urlparse(url).netloc) and proceeds to heuristic analysis. At this stage, the domain is compared against a reference list of trusted resources (including google.com, paypal.com, sberbank.ru, donstu.ru) using a fuzzy string matching mechanism (SequenceMatcher). Threshold values and subsequent logic allow identifying fake domains imitating known brands. When the similarity coefficient is greater than 0.75, a warning about possible spoofing (typosquatting) is issued;

- a dictionary common_typos contains typical substitutions (g00gle, micr0soft, paypai). If a typo substring is found in the domain, a warning is displayed.

Structural signs of suspiciousness:

1. domain length greater than 25 characters;
2. large number of hyphens/underscores (more than 3);
3. domain represented by an IP address;
4. large number of subdomains (more than 2).

Next, the module establishes a TLS connection with the host (socket + ssl.create_default_context()), obtains the certificate via getpeercert(), and analyzes:

1. issuer: country, organization, common name (via parse_ssl_issuer);
2. validity period: fields notBefore and notAfter compared with the current date; expired/not yet active certificates are flagged as an issue;
3. domain match: comparison of hostname with subjectAltName and commonName. Mismatch is recorded as a warning;
4. CA class: attempt to classify the certificate authority as free (Let's Encrypt/ZeroSSL, etc.) or paid (DigiCert/GlobalSign, etc.). If classification fails, the message "requires manual check" is displayed;
5. signature algorithm: upon detection of SHA-1 (if the field is accessible), a warning about the outdated algorithm is displayed.

Then, URL structure analysis follows (RFC 3986 logic via urlparse). At this stage, the scheme, path, and query parameters are analyzed:

- use of known shorteners (bit.ly, t.co, tinyurl.com, etc.);
- presence of suspicious constructions (@ in URL, or explicit http://);

- URL length greater than 100 characters;
- analysis of query parameters: searching for suspicious keywords (including Russian/English: password, verify, платеж (payment), пароль (password)).

After this, WHOIS analysis of the domain is performed.
Using python-whois, the following are extracted:

- registrar;
- creation date and its normalization (including handling lists and attempts to remove tzinfo);
- domain age (heuristic: less than 30 days = "very young domain");
- signs of privacy/anonymity (REDACTED, Whois Privacy);
- expiration date as an additional indicator of registration "quality".

The module uses protective handling of date formats, as WHOIS responses vary among different registrars.

Next, DNS checks (MX + SPF) are performed.
Using dns.resolver, two types of queries are performed:

- MX records: absence is treated as a risk factor (throwaway domains often do not configure mail).
- TXT records with SPF: searching for records containing 'spf'. Absence of SPF is recorded as a potential opportunity for sender spoofing.

Afterwards, the page content is analyzed (HTTP GET + BeautifulSoup).
The module makes a requests.get() request with a browser-emulating User-Agent, retrieves the HTML, and analyzes it without executing JavaScript, thus excluding active execution of malicious code.

Checks include:

- presence of JavaScript redirects via strings window.location / document.location within <script> tags;
- presence of hidden elements (style="display:none");
- search for phishing phrases (Rus/Eng) in the page text;
- quantity as a possible sign of content masking/spoofing.

Finally, a basic HTTPS check is performed. If the URL starts with http://, the module records a warning: the absence of a secure connection increases the risk of data interception.

After all checks are completed, a final risk assessment system is formed. Results are collected into a structure of the type "check → list of findings." Based on this, the overall risk is calculated:

- a confirmed threat via Google Safe Browsing means an immediate verdict of "PHISHING" and a risk of 5/5;
- further, risk signs are aggregated into a total score.

Report Generation

The final risk level (1–5) is transformed from the accumulated score, after which the final verdict is formed:

- 4–5: "very likely phishing / phishing"
- 3: "likely phishing"
- 2: "suspicious"
- 1: "likely safe"

The report is displayed in a text field and includes:

- header;
- visual risk level indicator (scale 1–5);
- final verdict;
- detailed results for all stages.

This format ensures transparency: the user receives not only the outcome ("phishing/not phishing") but also the reasons influencing the assessment.

To verify the correctness and reliability of the proposed method, a black-box testing method is used, where functionality assessment is carried out solely through external input data and observed output results, without considering the internal implementation of modules. This approach allows objectively checking whether the application meets specified requirements from an end-user perspective, correctly handles both standard and anomalous scenarios, and contains no functional defects affecting the reliability of the analysis.

To test the URL checking module, let's take several links ranging from legitimate to phishing:

1. https://edu.donstu.ru – a completely legitimate DSTU site, the module should give a verdict of "MINIMAL RISK." The check result is shown in Figure 1.

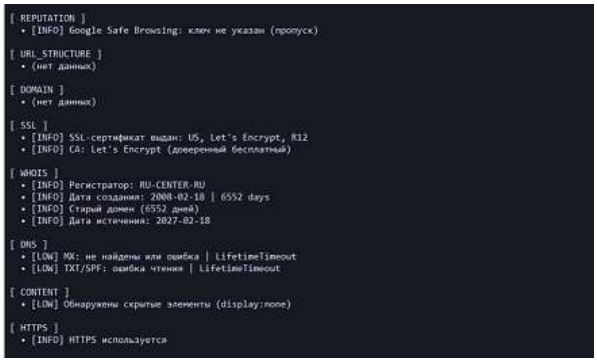

Figure 1: Result of checking a legitimate site

The module noted and added one point for the absence of MX and SPF records; however, for ordinary informational sites that do not accept email, the absence of such records is acceptable;

2. Checking the address http://unitus.mk.ua/sites/default/files/ctools/ams/cms/index/www/customer_center/customer-IDPP00C156/login.php showed that the site uses the http protocol, not https – following such links is unsafe. The check result is shown in Figure 2.

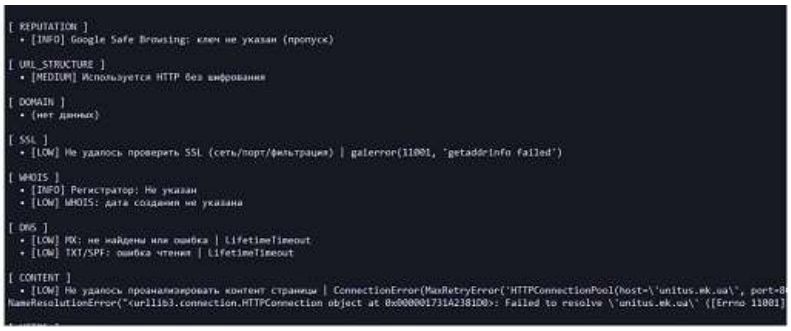

Figure 2: Result of checking an illegitimate site

The module noted an error checking the SSL certificate; it is absent because the http protocol does not provide for security certificates. The module also highlighted suspicious characters in the URL,

the absence of DNS records, and an error when checking the content. For each of these errors, the module added one point;

3. Let's check the phishing link http://hostpoint.ch.gbpevents.com.prunauneau.fr/943070958193827623/. The scan results are shown in Figure 3.

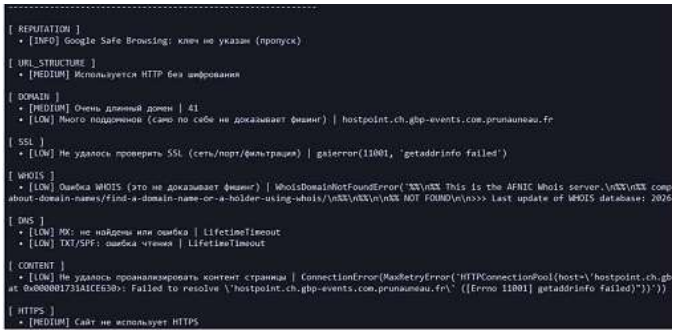

Figure 3: Results of scanning a phishing site

CONCLUSION

The developed application is designed adhering to the principles of "zero trust" and "least privilege." All components function exclusively in local mode: analysis of links, email addresses, texts, files, images, and QR codes is performed on the user's side without transmitting any data to external networks. This guarantees that confidential information – including internal domains, corporate emails, document fragments, intranet resource URLs, or image metadata – never leaves the protected perimeter and cannot be compromised during transmission or storage on third-party servers. The application does not use executable macros, external DLL libraries, embedded scripts, or ActiveX components, eliminating the possibility of malicious code injection through the analysis tool itself. All file read operations are performed in read-only mode, with no possibility of modifying the original data. Even when analyzing web content (in the link checking module), page loading is performed without executing JavaScript, without loading external resources, and without saving cookies – solely for extracting text and static HTML, preventing automatic activation of phishing redirects or exploits. Furthermore, the application does not save check history, does not write logs to disk, and does not create temporary files with sensitive content. All data processing occurs in RAM and is destroyed after the session ends. This is especially important when working with documents containing personal data and other sensitive information.

REFERENCES

Afanaseva NS, Elizarov DA, Myznikova TA. Classifying and countering phishing attacks. Ing J Don 2022,5:169-82.

Ahmad R, Terzis S, Renaud K. Getting users to click: a content analysis of phishers' tactics and techniques in mobile instant messaging phishing. Inf Comput Secur 2024,32:420-35. https://doi.org/10.1108/ICS-11-2023-0206

Batyushkin MV. "Phishing" – Computer fraud. Symbol Sci Int Sci J 2021,1:90-3.

Duru O. Smart phishing detection via URL characteristics: Machine learning and deep learning techniques. In: *2025 Innovations in Intelligent Systems and Applications Conference (ASYU),* September 10-12, 2025, Bursa, Turkey. New York: IEEE; 2025. p. 1-7. https://doi.org/10.1109/ASYU67174.2025.11208380

Elevate Security, Cyentia Institute. High Risk Users and Where to Find Them. 2023. https://ciso2ciso.com/wp-content/uploads/2023/06/High-Risk-Users-and-Where-to-Find-Them.pdf (accessed on July 2, 2023).

Federal Service for Technical and Export Control (FSTEC of Russia). Methodology for assessing information security threats. Methodological document of February 5, 2021. https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/metodicheskij-dokument-ot-5-fevralya-2021-g (accessed on December 10, 2024).

Gorda MD, Chechulin AA. Methodology for investigating phishing attacks. Informatizatsiya i svyaz' 2024,2:109-16. https://doi.org/10.34219/2078-8320-2024-15-2-109-116

Hadžiosmanović D, Sommer R, Zambon E, Hartel PH. Through the eye of the PLC: semantic security monitoring for industrial processes. In: Proceedings of the 30th Annual Computer Security Applications Conference. New York: Association for Computing Machinery; 2014. p. 126-35. https://doi.org/10.1145/2664243.2664277.

Hosseinzadeh M, Ali U, Ali S, Abbaszadi R, Gharehchopogh FS, Khoshvaght P, Porntaveetus T, Lansky J. Improving phishing email detection performance through deep learning with adaptive optimization. Sci Rep 2025,15:36724. https://doi.org/10.1038/s41598-025-20668-5

Informzashchita. Targeted phishing in Russia increased by 30%. May 5, 2025. https://www.infosec.ru/press-center/news/tselevogo-fishinga-v-rossii-stalo-na-30-bolshe/

Kaspersky. "Open cannot ignore": Employees of Russian companies tend to believe phishing emails from the security service: Press release. August 16, 2023. https://www.kaspersky.ru/about/press-releases/otkryt-nelzya-ignorirovat-sotrudniki-rossijskih-kompanij-sklonny-verit-fishingovym-pismam-ot-sluzhby-bezopasnosti?ysclid=mobbdk41su469220811 (accessed on December 10, 2024).

Khaled B, Rafik Z. A hybrid deep learning and anomaly detection framework for real-time malicious URL classification. arXiv:2512.03462 [cs.CV], 2025. https://doi.org/10.48550/arXiv.2512.03462

MITRE ATT&CK Framework. Threat Groups: Official website. n.d. https://attack.mitre.org/groups/ (accessed on December 10, 2024).

Nair R, Abbasi F, Pervez S. PhishEmailLLM: A Meta model approach to detect phishing emails by leveraging LLMs and machine learning models. In: Proceedings of the 2025 Australasian Computer Science Week. New York: Association for Computing Machinery; 2025. p. 19-29. https://doi.org/10.1145/3727166.3727169

NomuIIa AR. Advancing Phishing Protection: Employing Sophisticated Methods for Precise URL Evaluation, Master's Projects, 1374, San Jose State University, San Jose, CA, USA; 2024. https://doi.org/10.31979/etd.tt6d-f7ua

Tarasova YuA. Analysis of the phishing problem in the digital space. Int J Appl Fundam Res 2023,11:56-60.