Prepared for public presentation

Red Team Assessment

Prepared By

Student project Red Team
group "NoctuaSec"

for Bioinformatics club (PJATK)

On 12th August 2025

The names of tools and components in this report are arbitrary and are used solely for testing purposes.
They do not reveal internal work processes or unique technologies of the team.

Table of Contents

Executive Summary

Objective

The goal of the operation was to simulate a real-world scenario with a targeted cyberattack against the student organization "Bioinformatics club" to check its attack resistance to the most common techniques, including both popular technical vulnerabilities and human factors. The test included a web application of the club, the level of internal awareness among club members, and the readiness to withstand attacks with the use of social engineering and phishing attacks.

Methods Used

The Red Team applied a combination of Open-Source Intelligence (OSINT), social engineering, phishing campaign, and deployment of C2 infrastructure for connection with a backdoor.
In parallel, the campaign of website penetration was launched.

Findings

During the engagement period, a significant amount of data on the key figures of the organization was collected.
In particular, the member who was identified as the most susceptible to targeted attacks, as well as the head of the club.
The identified vulnerabilities could result in the leak of confidential data, unauthorized access to internal resources, and potentially significant reputational damage. The likelihood of similar incidents occurring in the future, initiated by other technical students within the university, is high.

Recommendation

The recommendation includes conducting training sessions on countering phishing and social engineering, as well as cleaning up or minimizing personal data available in public sources.

Scope

This assessment was limited to:

- The public web application of the “Bioinformatics Club”
- Publicly available information about its members (OSINT)
- Testing internal security awareness through controlled social engineering and phishing activities

No physical intrusions, denial-of-service attacks, or destructive actions were performed. All activities were conducted within the predefined engagement rules and with prior authorization.

Methodology

This engagement is conducted following the Penetration Testing Execution Standard (PTES), supplemented by the OWASP Web Security Testing Guide (WSTG) for web application assessments. The methodology emulates the tactics, techniques, and procedures of a real-world adversary, while operating strictly within the agreed legal and ethical boundaries defined in the Rules of Engagement.

1. **Pre-engagement Interactions** – Define scope, objectives, rules of engagement, and authorization. Establish secure communication channels and agree on operational constraints.

2. **Intelligence Gathering (OSINT)** – Collect publicly available information from open sources, registries, technical footprints, and social platforms to identify attack surfaces and potential vectors.

3. **Threat Modeling** – Analyze gathered intelligence to prioritize targets, model potential attack chains, and select techniques with the highest operational impact and lowest detection risk.

4. **Exploitation** – Execute planned attacks to gain access to systems, networks, or applications, using authorized techniques such as social engineering, phishing, web exploitation, or network attacks.

5. **Post-Exploitation** – Assess the extent of compromise, maintain controlled persistence, and demonstrate potential business impact without causing operational disruption.

6. **Reporting** – Document all findings, mapping them to the MITRE ATT&CK® framework, and provide actionable remediation recommendations.

Operational Security (OPSEC) is maintained throughout the engagement by using compartmentalized and encrypted data storage, multi-factor authentication, anonymized infrastructure, and a strict need-to-know principle, as well as the least privilege principle. No actions are taken beyond the authorized scope, and all testing activities are designed to avoid permanent damage or data loss.

All findings are documented and mapped to the MITRE ATT&CK® framework to standardize classification and provide clear context for defensive improvements. The mapping covered all executed and planned tactics, techniques, and procedures (TTPs), including the following examples:

- **Initial Access**
    - T1566.001 – Spearphishing Attachment: A targeted phishing email was crafted to deliver a password-protected ZIP archive containing the initial payload.
- **Execution**
    - T1204.002 – User Execution: Malicious File: The attack relies on the user executing a masqueraded .LNK file contained within the ZIP archive.
    - T1059.001 – Command and Scripting Interpreter: PowerShell: The .LNK file executes a PowerShell stager to download and run the main backdoor.
- **Persistence**
    - T1547.001 – Boot or Logon Autostart Execution: Registry Run Keys: The backdoor (IntelAudioService.cs) creates a new entry in the HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run registry key to ensure it runs automatically at logon.

- **Defense Evasion**
    - T1497 – Virtualization/Sandbox Evasion: The backdoor checks the environment for signs of a sandbox (e.g., processor count, disk size) and enters an infinite sleep loop if detected.

- **Command and Control(C2)**
    - T1105 – Ingress Tool Transfer: The PowerShell stager transfers the main backdoor executable from the server to the compromised system.
    - T1071.001 – Application Layer Protocol: Web Protocols: The C2 channel operates over HTTPS, sending and receiving commands via the public Telegram Bot API to blend in with normal web traffic.

Exploitation

(T-0) June 26 - July 24, 2025 | Pre-engagement Interactions:

During this preparatory phase, the "NoctuaSec" team completed the full cycle of legal and operational setup. A comprehensive package of governing documents was developed and signed:

- **Penetration Test Authorization and Rules of Engagement Agreement:** The main agreement that clearly defines the scope of work, rules of engagement, and liabilities.
- **Participant Informed Consent Form:** Individual consent forms from each participant, authorizing testing activities, including access to their data.
- **Internal Red Team Member Agreement:** An internal agreement defining the roles and data handling procedures for each operator.

In parallel, a secure operational infrastructure was established:

- **Compartmentalized and encrypted data vaults** (VeraCrypt) were created for each sub-team, with access granted on a need-to-know basis.
- **Secure communication channels** were set up (Signal for critical secrets, Telegram for general coordination).

(T-1) July 3 - July 10, 2025 | Intelligence Gathering (OSINT):

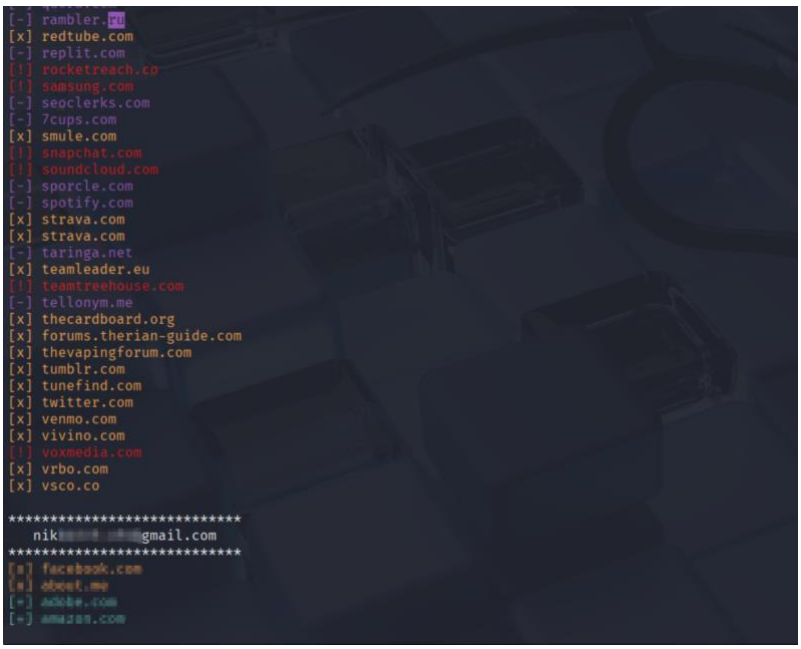
1. OSINT Objective

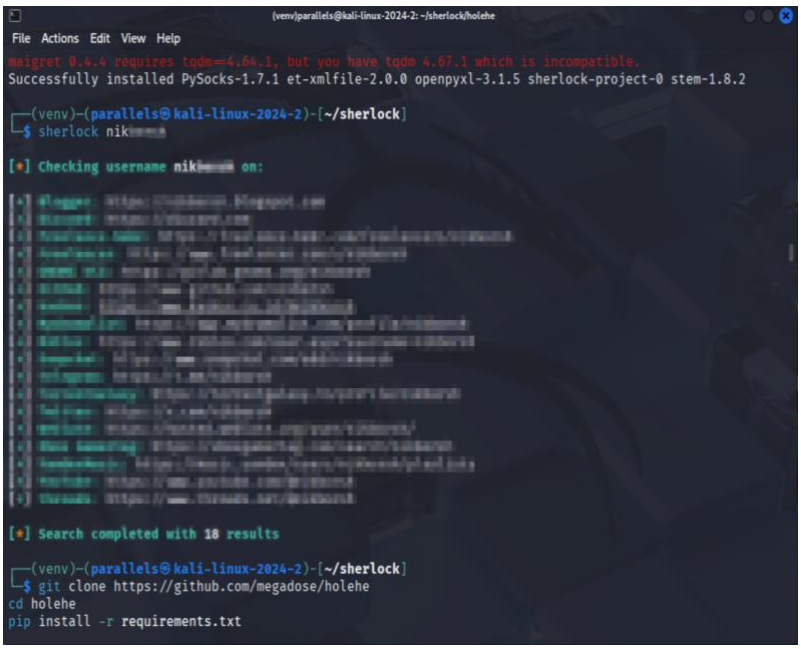
The primary objective of the OSINT phase was to gather publicly available intelligence on selected members of the "Bioinformatics Club Association". The goal was to build comprehensive profiles to identify potential vectors for the initial access phase, primarily through social engineering, and to understand the targets' technical skills, interests, and professional networks.

2. Sources and Tools Used

Intelligence was gathered from a wide range of public sources, including but not limited to:

- **Social Media Platforms:** LinkedIn, Facebook, Instagram, Twitter.
- **Professional & Academic Networks:** University websites, personal blogs.
- **Data Breach Search Engines:** Analysis of historical data breaches to identify compromised credentials and personal information.
- **General Search Engines:** Google, DuckDuckGo using advanced search operators.
- **Specialized OSINT Tools**: Sherlock for cross-platform username enumeration; Holehe for detecting existing email accounts across multiple online services, and others.





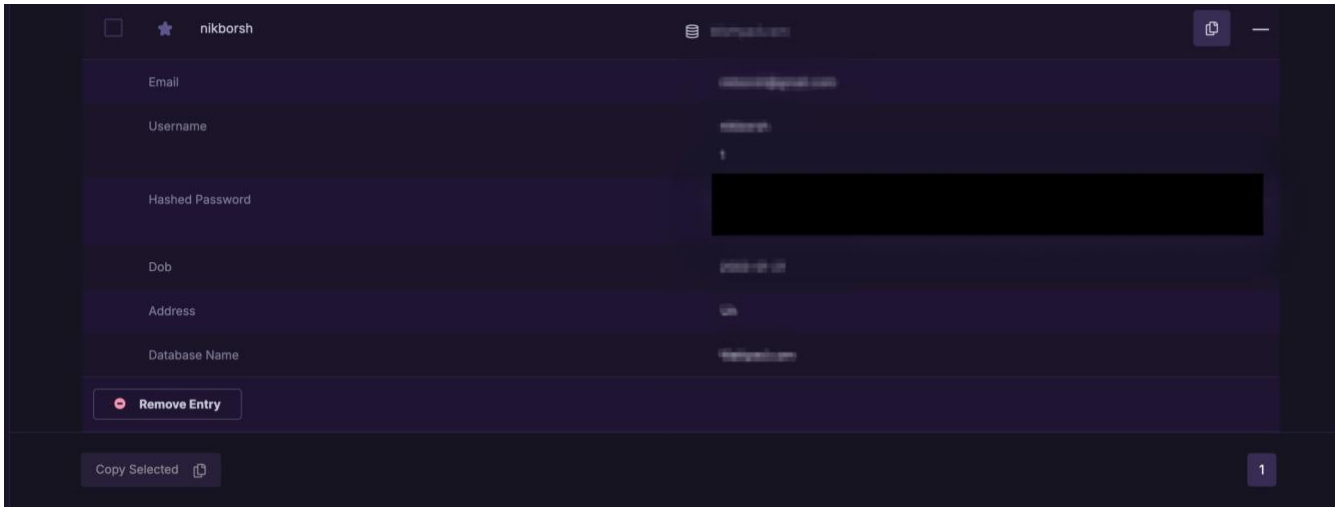
Picture 1.1 – Holehe OSINT Tool Output: Screenshot showing the Holehe OSINT tool used to check if a target email address is linked to active accounts on various online services. Results containing personal data have been redacted.

Picture 1.2 – Sherlock OSINT Tool Output: Screenshot showing the use of the Sherlock OSINT framework to identify publicly registered accounts associated with a target username. Sensitive identifiers have been redacted to protect privacy.

3. Types of Data Acquired

The investigation resulted in a significant amount of actionable intelligence, including:

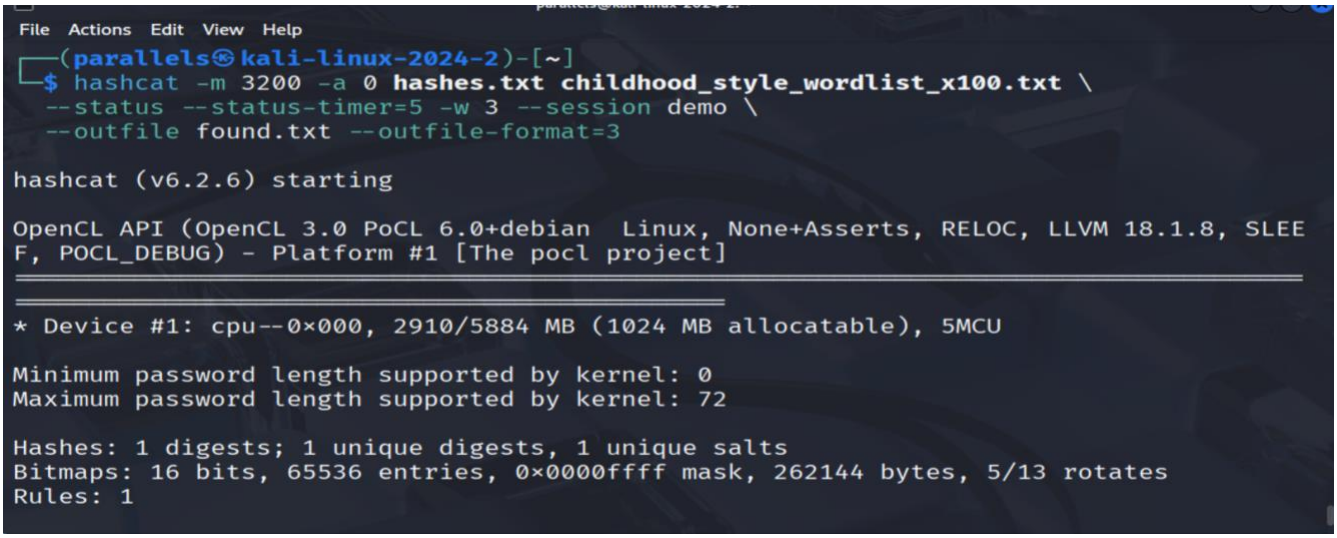
- **Contact Information and Digital Footprint:** Email addresses, usernames across various platforms, and phone numbers.
- **Professional and Academic Background:** Educational history, technical skills, current projects, and professional connections.
- **Personal Information:** Dates of birth, personal interests, hobbies, and social circles.
- **Exposure in Data Breaches:** Revealed compromised accounts, email addresses, and associated password hashes from past security incidents on third-party services.



Picture 2 – Example of hashed credential discovered during OSINT phase (full hash value redacted for security)

Password Hash Analysis Attempt

We tried to recover the password from the obtained hash using the Hashcat tool. However, due to taking up a significant amount of time by this method, and the absence of confirmation that the hash belonged to the target (due to the old date of the data leak), it was decided to stop the process. This helped us focus on more relevant parts of the assessment.



Picture 3 – Hashcat Password Recovery Attempt: Screenshot showing Hashcat (mode 3200 – bcrypt) using a custom wordlist created by AI to attempt password recovery.

4. OSINT Conclusions & Actionable Intelligence

The OSINT phase led to the following key conclusions, which formed the basis for our social engineering strategy:

- **Primary Entry Vector Identified:** One target, "Ayca Yildiran", was identified as the most viable entry point due to a wider public digital footprint and specific personal interests that could be leveraged in a phishing pretext.

- **High-Value Target Profiled:** A detailed profile of the primary target, "Mykyta Borshchov", was created, outlining his position and professional network, which would be leveraged in a later, post-exploitation phase.

- **Phishing Pretext Developed:** Based on the gathered intelligence, a highly targeted and ethically sound phishing pretext was developed, focusing on a neutral/negative trigger (a mandatory university course) to maximize realism and minimize negative emotional impact, as advised during the mentorship phase.

- **Data Breach Findings:** The discovery of compromised credentials in historical data breaches confirmed a potential (though not pursued) vector for credential stuffing attacks.

(T-2) July 10 - August 2, 2025 | Payload Development:

1. Objective

The primary objective of this phase was to develop a custom, persistent C# backdoor (Backdoor.exe) to serve as the main payload for the engagement. The design priorities were reliability, stealth, and modular functionality to support reconnaissance and potential lateral movement.

2. Payload Architecture & Functionality

The final payload is a single, self-contained executable built on the .NET framework. Its architecture is divided into three logical components:

- **Main Program (Program.cs):** Responsible for the initial execution flow, including anti-analysis checks and establishing persistence. Upon a successful start, it initiates the C2 communication loop.

- **Command & Control (TelegramManager.cs):** This module handles all communication with the operator via the Telegram Bot API over HTTPS. It is responsible for fetching new commands and exfiltrating the results.

Pictures 4.1 & 4.2 - Fragment of the C# CommandHandler class showing a few implemented commands. Each case executes specific system-level actions when triggered by the remote operator.

- **"Quiet" Command Handler (CommandHandler.cs):** This is the core logic module that parses and executes commands. To minimize behavioral detection by EDR systems, the handler executes most commands using internal .NET API functions, avoiding the creation of noisy child processes like cmd.exe. A fallback to cmd.exe is used only for non-standard commands.

3. Persistence Method

Persistence was achieved by creating an autorun key in the HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run registry path. This method was chosen for its high reliability and because it does not require administrator privileges, making it a suitable choice for a standard user-context compromise scenario.

4. Evasion Measures

Several evasion techniques were implemented to increase stealth:

- **Anti-Sandbox/Analysis:** The payload performs checks for common sandbox environments (processor count, disk size, debugger attachment) and enters a sleep-loop if detected.

- **Time-Based Evasion:** Randomized sleep intervals (sleep-delay) are used at critical execution stages (initial launch, persistence, C2 beaconing) to break the chain of actions and evade automated behavioral analysis.

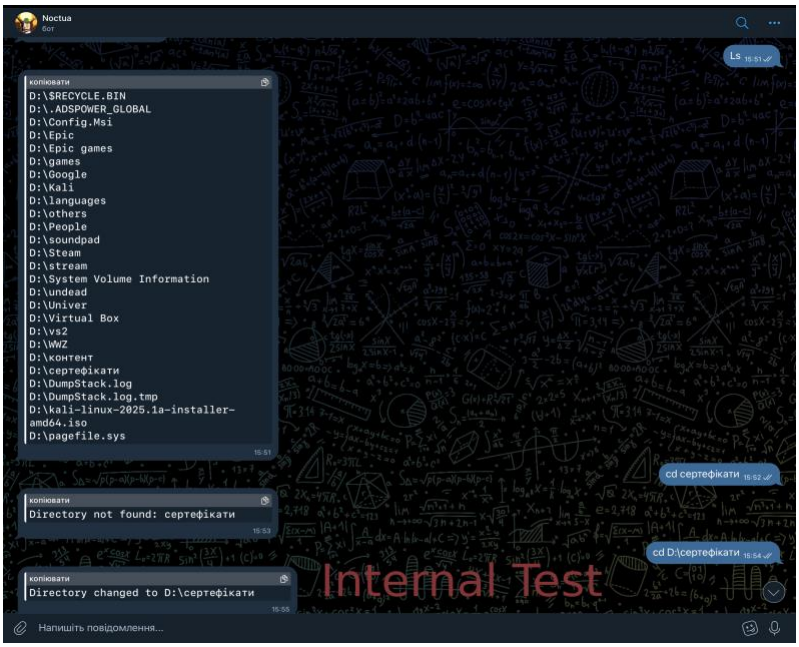
- All C2 communications are encrypted by default via HTTPS/TLS.

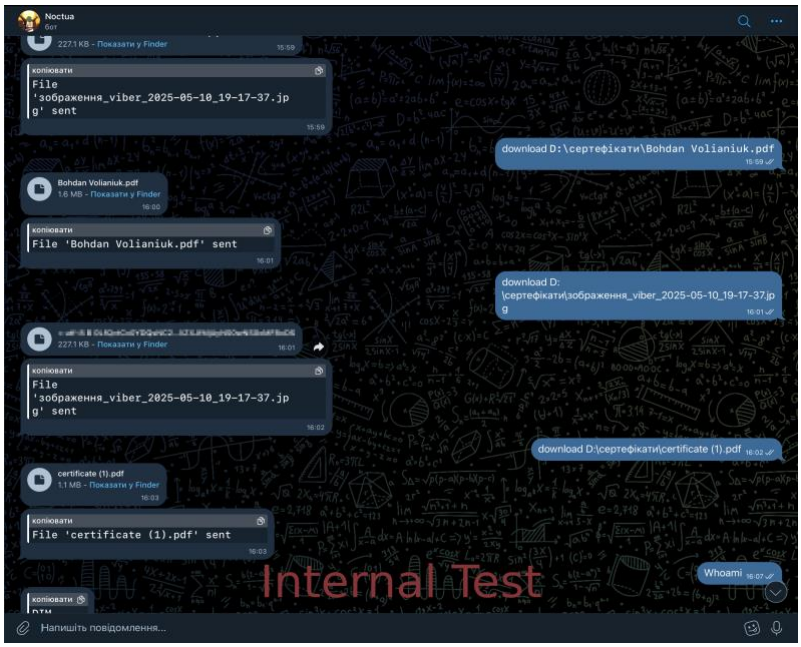
- The backdoor authenticates the operator by checking the chat_id of incoming messages, ignoring commands from unauthorized sources.

- Operational secrets (bot token, chat ID) are stored in Base64 format within the compiled executable.

5. Testing & Validation

The payload was rigorously tested in a controlled virtual environment against multiple endpoint configurations. Testing focused on ensuring operational stability, absence of crashes, and reliable compatibility with the C2 infrastructure before deployment.





Pictures 5.1 & 5.2 – Internal test of C2 Channel: Successfully used commands such as ls (Outputting all files and folders in the current directory), cd (Changing directory), and download (Downloading a file)

(T-3) July 28 - August 1, 2025 | Web Application Penetration Testing:

1. Objective

The objective of this phase was to conduct a thorough security assessment of the Bioinformatics Club's official website (bioinformaticsclub.pl) to identify and validate potential vulnerabilities. The goal was to determine if the web application could serve as a viable entry point for an attacker and to evaluate its overall security posture against common web-based threats.

A simulated attack narrative was considered: if vulnerabilities such as injection flaws, authentication weaknesses, or insecure direct object references were present, an attacker could potentially bypass the phishing vector and directly compromise accounts or backend infrastructure via the web application.

2. Methodology

The assessment was performed following a structured methodology based on the OWASP Web Security Testing Guide (WSTG). The testing combined automated scanning with in-depth manual verification to ensure comprehensive coverage.

- **Key areas of testing included:**
    - **Information Gathering (OWASP WSTG-INFO) –** The engagement began with reconnaissance to map the application's attack surface. This included network scanning with Nmap to identify open ports and services, technology stack fingerprinting using Wappalyzer, and subdomain and directory enumeration.
    - **Authentication & Session Management Testing (OWASP A07:2021** – Identification and Authentication Failures) – The login endpoint (/login) was tested for common vulnerabilities, including weak password policies and the security of session tokens.

- **Input Validation Testing (OWASP A03:2021 – Injection / A05:2021 –** Security Misconfiguration) – All user-facing input fields and API endpoints were manually and automatically tested for injection flaws, including Cross-Site Scripting (XSS) and SQL Injection (SQLi).

- **Configuration & Deployment Management Testing (OWASP A05:2021 –** Security Misconfiguration) – The server configuration was reviewed for security misconfigurations, such as insecure HTTP headers and exposed services.

Tools Used:

- **Automated Scanning:** nuclei, Nmap
- **Manual Testing & Analysis:** Burp Suite, Browser Developer Tools
- **Supplementary Tools:** sqlmap, ffuf, whois, dig

3. Findings & Technical Analysis

The web application demonstrated a solid security posture with several key defensive mechanisms in place. No critical or high-risk vulnerabilities that would allow for direct system compromise were identified.

Summary of Findings:

- **Network Reconnaissance:** An Nmap scan revealed three open ports for the main website, and an unknown service, which appears to be a local access point. The SSH service is running a modern, patched version of OpenSSH (9.6p1). Secure

- **Cross-Site Scripting (XSS) – OWASP A03:2021:** All user input fields on the main site were tested for Reflected and Stored XSS. No vectors were found to execute arbitrary JavaScript. The application appears to correctly sanitize user input. Secure

- **SQL Injection (SQLi) – OWASP A03:2021:** The /login endpoint was tested for SQL Injection using sqlmap with a sample POST request. The tool's attempts were consistently blocked, suggesting the presence of a Web Application Firewall (WAF) or a similar filtering mechanism that effectively prevented the attack. Secure

4. Overall Security Posture Assessment

While no critical vulnerabilities were discovered, the assessment provides valuable insights into the application's security. The presence of a likely WAF and strong security headers indicates a proactive approach to security from the development team.

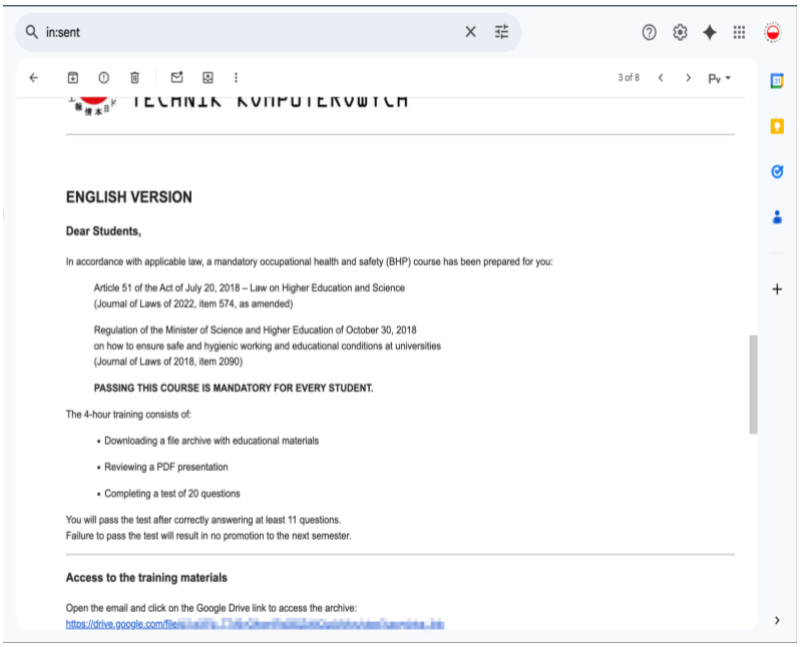
(T-4) August 2 - August 6, 2025 | Initial Access & Delivery:

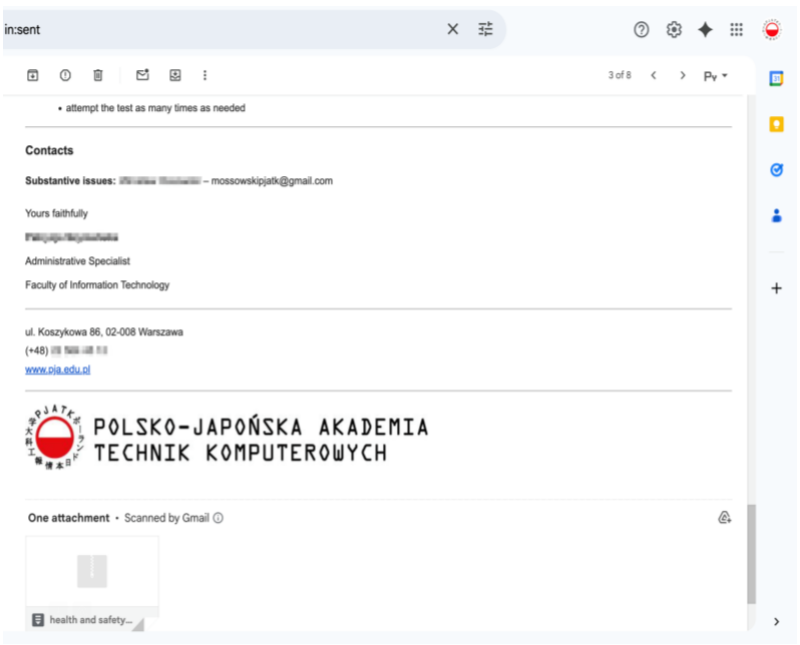
1. Objective

The objective of this phase was to execute a social engineering campaign to gain initial access to a target system within the "Bioinformatics Club" network. The goal was to successfully deliver the installer (Installer.ps1) of the backdoor (Backdoor.exe) and execute it, establish persistence, and confirm a stable Command and Control (C2) channel, thereby validating the entire attack chain developed in the previous phases.

2. Attack Vector

Based on the OSINT findings and mentor feedback, a spear-phishing email was selected as the primary attack vector. The pretext was carefully designed to be ethically sound and contextually plausible for the target audience. We chose to impersonate the university's Dean's Office, delivering a message about a mandatory health and safety course required for progression to the next semester. Such a course was implemented in the past by the university. This neutral/negative trigger was chosen over a positive lure to minimize the risk of negative emotional impact on the targets and to test their compliance with official-looking directives.





Pictures 6.1 & 6.2 - Phishing Email

3. Payload Delivery Method

To bypass email gateway security scanners and the default blocking of macros in Microsoft Office, a multi-stage delivery method was used:

- **Password-Protected ZIP Archive:** The payload was contained within a .zip archive, with the password provided in the body of the email. This prevents automated scanning of the contents.

- **Malicious LNK File:** Inside the archive, the primary payload was a shortcut file (.LNK) masquerading as a test portal.

- **PowerShell Stager Execution:** Upon execution by the user, the .LNK file launches the Installer.ps1 PowerShell stager. The stager's sole purpose was to download the main backdoor from our hosted secure server and execute it.

4. Execution & OPSEC Considerations

The entire execution chain was designed to be as stealthy as possible within the project's constraints. The PowerShell stager ran in a hidden window, downloaded the main backdoor to a masqueraded system folder (C:\ProgramData\Audio), and launched it. The backdoor itself was programmed with time-based delays and anti-sandbox checks to evade initial automated analysis before attempting to establish persistence and initiate C2 communication.

5. Outcome & Analysis

The phishing emails were sent to the selected group of consenting participants (4) at the beginning of the active testing window. However, after several days of monitoring, **no successful backdoor execution was observed.**

Analysis of Failure: This outcome is not considered a technical failure of the payload, but rather a valuable lesson in operational timing and target awareness. The primary reason for the campaign's lack of success is attributed to external circumstances:

- **Timing:** The engagement was conducted in August, which coincided with the university's summer holiday period. The targets (based on their words) were not actively checking their university-related emails or were doing so infrequently.

- **Low Urgency:** Despite the pretext of a "mandatory course", the deadline set in the email was several days away, which may not have created a sufficient sense of urgency for the targets to act immediately during their vacation time.

- **Additionally:** one participant reported that the phishing email was delivered to the spam folder. Combined with the use of a sender domain slightly different from the official university’s domain, this further reduced the likelihood of recipients engaging with the message.

This result highlights a critical aspect of real-world Red Team operations: the success of a social engineering campaign is heavily dependent on timing and the target's current context. While our delivery mechanism was technically sound, the operational window proved to be suboptimal, leading to a failure to achieve initial access.

(T-5) August 6 - August 8, 2025 | Command & Control (C2) Operations:

1. Objective

The primary objective of this phase was to establish a persistent Command & Control (C2) channel following a successful initial access. The goal was to test the target environment's ability to detect and prevent covert outbound communications from a compromised host. Success would be defined by the ability to maintain stable, remote control over the implant without triggering defensive mechanisms.

2. Methodology

Our methodology was based on adversary emulation principles, using a custom-developed toolchain:

- **C2 Framework:** A custom C2 protocol was implemented over the Telegram Bot API, utilizing standard HTTPS (T1071.001) for all communications. This approach was chosen to blend C2 traffic with legitimate web activity.

- **Payload:** The C2 logic was embedded within our C# backdoor (Backdoor.exe), which was designed with multiple evasion and persistence features.

- **Intended Persistence:** The payload was programmed to establish persistence via a Registry Run Key (T1547.001) to ensure the C2 channel would survive reboots.

3. Observations

During this phase, the C2 channel connection was not achieved. This was a direct result of the preceding Initial Access phase (Phase 4) not producing any payload execution events on target hosts. As the phishing campaign did not result in payload execution, the backdoor was never deployed, and no outbound C2 connection attempts occurred.

While a functional C2 was not established, this result provides a critical insight: **the human element proved to be the strongest layer of defense in this engagement.** The targets' operational constraints (being on vacation) effectively prevented the attack chain from progressing to the C2 phase.

4. Legal Disclaimer

The absence of a successful C2 connection during this engagement is not a guarantee of absolute network security. This finding reflects only the conditions present during the specific testing period and is a direct consequence of the initial access attempt being unsuccessful.

(T-6) August 8 - August 12, 2025 | Post-Engagement Activities & Cleanup:

Following the conclusion of the active testing window on August 8, 2025, a controlled post-engagement process was initiated to ensure the target environment was returned to its original state and that no residual artifacts from the assessment remained. These activities were conducted in accordance with the agreed-upon Rules of Engagement and professional operational security (OPSEC) practices.

Key steps included:

- **Tool & Payload Removal:** All components of our custom toolchain, including the Backdoor.exe, were removed from the server. The vanish command was used to securely delete all toolchains from participants' devices in case they were downloaded.

- **Infrastructure Decommissioning:** The C2 (Command & Control) infrastructure, which included the Telegram Bot API channel, was shut down.

- **Informing Participants:** All participants were formally notified about the end of the test, as well as about phishing emails that had been sent as part of the agreed engagement scope.

- **Data Handling & Secure Disposal:** All sensitive data collected during the engagement, including OSINT reports, screenshots, and credentials, was handled according to our internal "Sensitive Data Handling Protocol". Following the final analysis, the encrypted VeraCrypt containers holding this data were securely destroyed, and the deletion was verified by two team members.

- **Log Review & Coordination with Client:** We coordinated with the client ("Bioinformatics Club") to provide a timeline of our key activities, assisting them in correlating their internal logs with our actions to evaluate the performance of their monitoring and detection capabilities.

Outcome: The post-engagement phase successfully concluded, confirming that the operational environment was returned to its pre-test state and that no residual risk from the assessment activities persisted. All project data was securely disposed of in accordance with our legal and ethical obligations.

Risk Assessment

This section summarizes the general security risks identified during the engagement, including both observed vulnerabilities and potential attack vectors that were either mitigated or not applicable in the current scope. The purpose of this assessment is to provide the client with a realistic understanding of the security posture in the context of broader threat models.

1. OSINT & Public Exposure Risks

- **Observed Impact:** Multiple public data points were collected about the target organization and its members.
- **Potential Threat:** Adversaries could leverage OSINT for targeted phishing, credential stuffing, or reputation damage.
- **Risk Rating:** Medium.

2. Web Application Exploitation Risks

- **Observed Impact:** No critical vulnerabilities were identified during web application testing.
- **Potential Threat:** If a future code change introduces authentication or input validation flaws, an attacker could bypass phishing entirely and gain access via direct exploitation.
- **Risk Rating:** Low.

3. Network Perimeter Risks

- **Observed Impact:** Open ports (including SSH and service on port 4444) are exposed to the public internet.
- **Potential Threat:** If these services are not properly restricted, attackers may attempt brute force or exploitation of service-specific vulnerabilities.
- **Risk Rating:** Medium.

4. Phishing & Social Engineering Risks

- **Observed Impact:** The phishing campaign did not result in a successful compromise.
- **Potential Threat:** A future campaign could increase its effectiveness not necessarily by changing the sending address or domain, but by optimizing **timing** (e.g., sending during university studying time) and **urgency** in the message content (e.g., requiring immediate action to avoid a negative consequence). These psychological triggers often bypass technical safeguards and exploit human decision-making under pressure.
- **Risk Rating:** High.

5. Command & Control (C2) Channel Risks

- **Observed Impact:** No C2 channel was successfully established during testing.
- **Potential Threat:** If initial access is obtained in a future incident, covert outbound communications (e.g., HTTPS-based C2) may bypass detection and lead to confidential data leak and potential reputation loss.
- **Risk Rating:** High.

Risk Matrix

Threat Scenario	Impact (1–5)	Likelihood (1–5)	Risk Score	Risk Level	Justification
OSINT & Public Exposure Risks	2	5	10	Medium	Easily executable and may lead to more severe attacks such as phishing.
Web Application Exploitation Risks	2	3	6	Low	No vulnerabilities identified during testing and the site is not a critical component of operations.
Network Perimeter Risks	2	4	8	Medium	Open ports increase the attack surface and could be exploited if combined with other weaknesses.
Phishing & Social Engineering Risks	3	5	15	High	High likelihood due to human factor; can quickly lead to credential theft and unauthorized access.
Command & Control (C2) Channel Risks	5	3	15	High	Established C2 channels enable full system compromise; even with moderate likelihood, impact is critical.

Risk Score Legend:

- **1–7** → **Low** — Minimal impact or low likelihood; limited effect on operations.
- **8–13** → **Medium** — Noticeable impact or moderate likelihood; requires mitigation planning.
- **14–19** → **High** — Severe impact or high likelihood; requires immediate attention.
- **20–25** → **Critical** — Catastrophic impact and/or very high likelihood; urgent remediation required.

Recommendations

Based on the identified risks and their respective ratings, the following recommendations are provided to reduce the organization's attack surface and mitigate the likelihood of successful exploitation:

1. OSINT & Public Exposure Risks – Medium

- Reduce publicly available sensitive information on social media, organizational websites, and third-party platforms.
- Implement continuous OSINT monitoring for newly exposed data and potential impersonation attempts.
- Change leaked passwords
- Conduct staff awareness training on limiting public disclosure of operational details.

2. Web Application Exploitation Risks – Low

- Maintain regular application security testing (DAST/SAST) even if no current vulnerabilities are identified.
- Implement secure coding guidelines and ensure developers receive annual secure development training.
- Apply strict access control to the application backend and administration panels.

3. Network Perimeter Risks – Medium

- Close unnecessary open ports and apply a default-deny inbound firewall policy.
- Deploy network intrusion detection/prevention systems (IDS/IPS) to monitor suspicious traffic.
- Perform quarterly perimeter scans to identify and remediate new exposures.

4. Phishing & Social Engineering Risks – High

- Conduct regular phishing simulation campaigns to measure and improve staff resilience.
- Implement multi-factor authentication (MFA) for all critical systems to limit impact of credential theft.
- Provide targeted training focusing on urgent, time-sensitive phishing scenarios.

5. Command & Control (C2) Channel Risks – High

- Block known C2-related domains and IPs using threat intelligence feeds in firewalls and proxies.
- Monitor outbound traffic anomalies to detect possible exfiltration or beaconing activity.
- Implement application whitelisting to limit the execution of unauthorized software.

Next Steps

1. **Within 30 days** – Conduct targeted training for key club members on phishing and social engineering prevention. Remove or limit publicly available personal data identified during the OSINT phase.

2. **Within 60 days** – Implement technical measures to reduce the network attack surface (close or secure open ports, monitor suspicious activity). Reassess the web application for new vulnerabilities and configure basic mechanisms to detect potential C2 channels.

3. **Ongoing** – Schedule annual security assessments and maintain continuous monitoring of critical assets to ensure rapid detection and response to emerging threats.

Acknowledgements

The “NoctuaSec” team expresses our sincere and deep appreciation to Mr. Gynvael Coldwind of Dragon Sector for his invaluable mentorship to our team throughout this project.

His expert guidance on such topics as professional ethics, operational security, and Red Team strategy was crucial in shaping the quality and methodology of our test.
This project would not have been possible without his willingness to share his time and world-class experience with us.

Contacts & Sign-off

Mykhailo Andreichyn
Red Team Lead (NoctuaSec)

Email:
andrmykhailowork@gmail.com

LinkedIn:
Mykhailo Andreichyn

Bohdan Volianiuk
Red Team Member (NoctuaSec)

Email:
bogdanvolya31@gmail.com

Pawiel Hrusha
Red Team Member (NoctuaSec)

Email:
pawelgrusza19@gmail.com

Tymur Katalnikov
Red Team Member (NoctuaSec)

Email:
katalnikofff@gmail.com

CLIENT ACKNOWLEDGEMENT & ACCEPTANCE

By signing below, the Client acknowledges receipt of this report and accepts it as the final deliverable for the engagement, subject to the scope and limitations described herein.

☑ Accepted

☐ Accepted with comments/exceptions

☐ Received (acknowledgement only)

Comments / Exceptions:

_____

_____

For the Consultant:

Title: Founder, Head & Representative of NoctuaSec

Full Name: Mykhailo Andreichyn

Date: 08/15/2025

Signature: 

For the Client:

Title: President of Bioinformatics Club (PJATK)

Full Name: Mykyta Borshchov

Date: 08/15/2025

Signature: