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.

Methodology

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.

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 antianalysis 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 sleeploop 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.

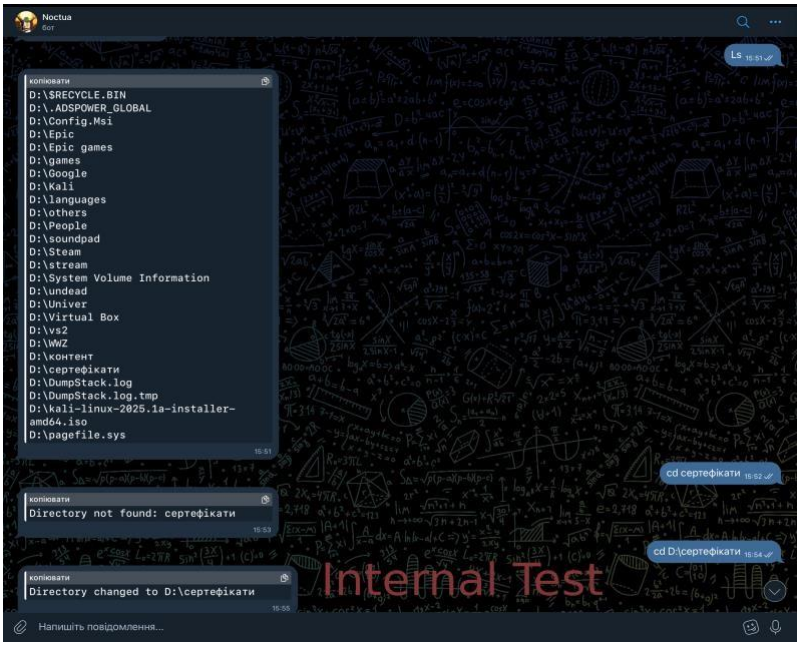
- All C2 communications are encrypted by default via HTTPS/TLS.

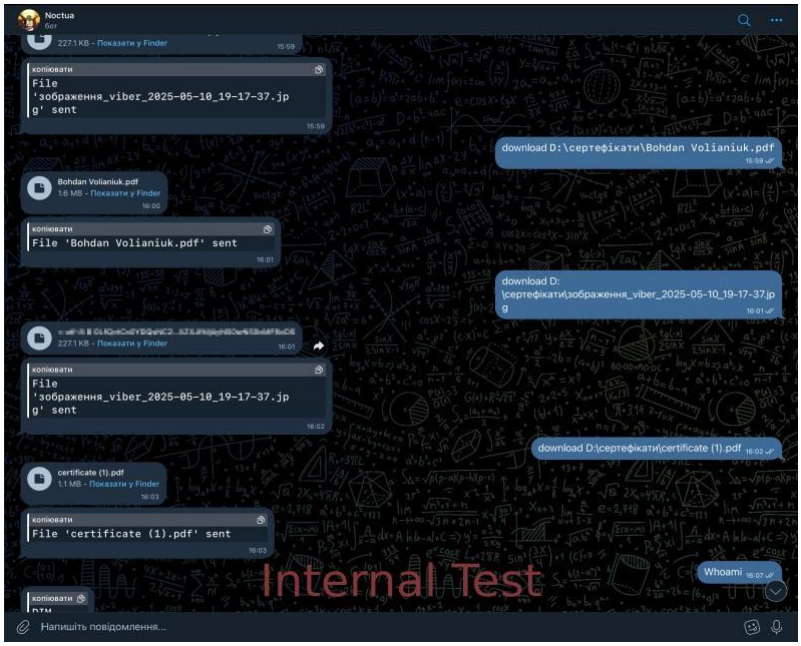
- The backdoor authenticates the operator by checking the chat_id of incoming messages, ignoring commands from unauthorized sources.

- 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)

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.

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.

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

Note: *This document is an analytical case study. The complete 34-page technical assessment report, including all raw operational logs and detailed tool outputs referenced in this study, is available as the [Full Technical Appendix (PDF) at this link].*

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: