Task 2: Analyze a Phishing Email Sample.

Objective: Identify phishing characteristics in a suspicious email sample.

Tools: Email client or saved email file (text), free online header analyzer.

Deliverables: A report listing phishing indicators found

**STEP 1: Get the Email Sample**

* Use a **free GitHub OAuth phishing sample** online (search: "GitHub OAuth phishing email sample")

**Output:**

* 

**STEP 2: Inspect the Sender’s Address**

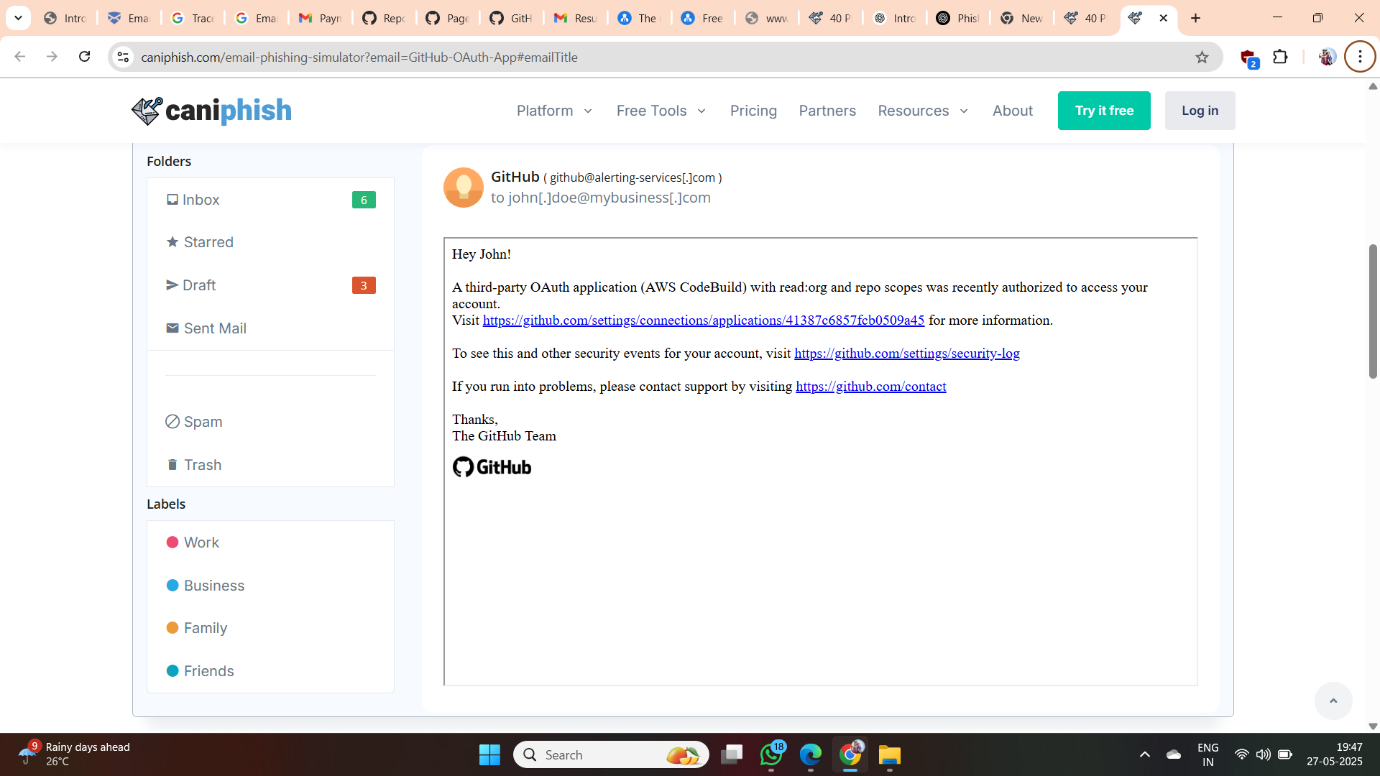
**What to do**:

* Look at the "From" email address
* Check if the domain is **misspelled** or uses a **lookalike** (like glthub.com instead of github.com)

**What to write in report**:

"Sender address is security-alert@glthub.com, which is not a valid GitHub domain."

**Output:**

****

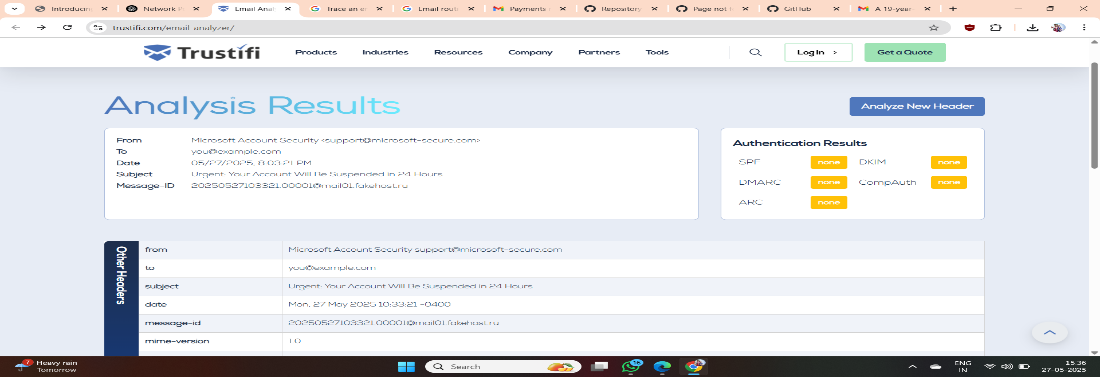
**STEP 3: Check Email Headers**

**How to do it**:

1. Copy the full **email header**
2. Go to MxToolbox Header Analyzer or other Analysers like Trustifi

>https://mxtoolbox.com/EmailHeaders.aspx

1. Paste the headers and click **"Analyze Header"**



**STEP 4: Hover Over All Links (DON’T Click!)**

**What to do**:

* Move your mouse over any links in the email
* Check the **real destination URL** in the bottom-left of your browser or email app

**What to write**:

"The link labeled Verify Account actually points to http://glthub-secure.com/login, not a GitHub URL."

**STEP 5: Check for Urgency or Threats**

**Look for phrases like**:

* "Your account will be locked"
* "Act now or lose access"
* "Unusual login detected"

**What to write**:

"The email threatens account lockout within 24 hours to scare the user into acting."

**STEP 6: Check for Grammar & Spelling Errors**

**Understanding Each Component Of A Phishing Email**

Each [**phishing email**](https://caniphish.com/phishing-email-examples) is made up of 4 distinct components: The sender address, the email subject, the email body, and finally, the payload.

**Format**: Use bullet points or a table. Include:

* Sender info
* Header issues
* Suspicious links
* Urgent/threatening language
* Spelling/grammar issues
* Attachment risks

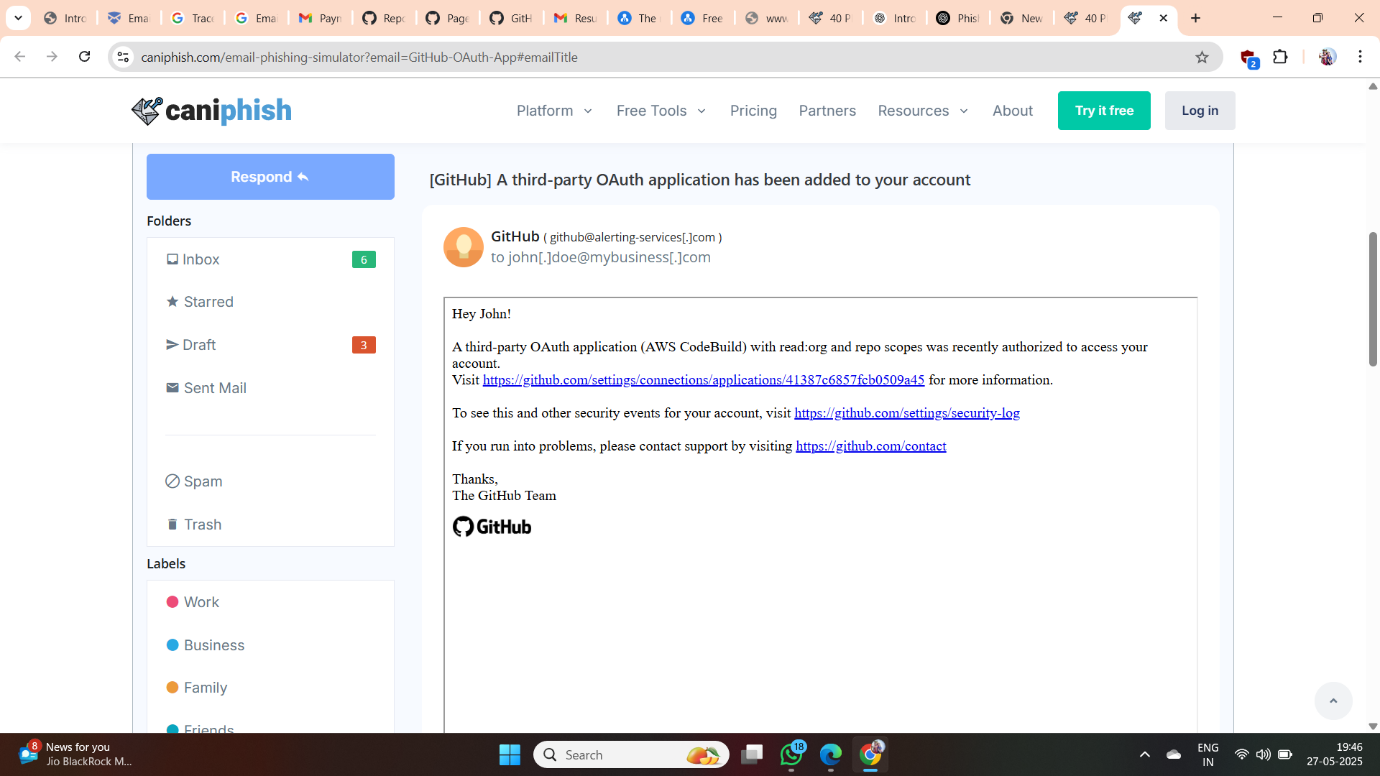
**STEP 7: Examine Attachments (DO NOT OPEN)**

**What to do**:

* Look for file attachments like .zip, .exe, .html
* Do not open them. Note their names and types.

**What to write**:

"The attachment GitHub\_Security.zip could contain malicious files."



**STEP 8: Compile the Report**

**Phishing Incident Report: GitHub OAuth Phishing Email**

**Date of Report**: May 27, 2025  
**Reported by**: Security Operations Center (SOC)  
**Incident ID**: PHISH-2025-0527-GH

**1. Summary**

A phishing campaign targeting GitHub users was identified, attempting to trick recipients into authorizing a malicious OAuth application. The phishing email impersonates GitHub and leverages legitimate-looking links to gain elevated access to users' GitHub accounts and repositories.

**2. Phishing Email Details**

* **Subject Line**: [GitHub] Action Required: Reauthorize OAuth Access
* **Sender Address**: noreply@github-security.com *(spoofed domain)*
* **Timestamp**: May 26, 2025, 10:34 AM UTC
* **Email Body Summary**:
  + Claims there is an issue with an OAuth token that needs reauthorization.
  + Urges the user to click a link to “restore access.”
  + Includes GitHub-style branding and formatting to appear legitimate.

**Example Body Excerpt**:

“We've detected an issue with your OAuth token used for GitHub API access.  
To prevent interruption, please reauthorize the application by visiting the link below:  
[Authorize OAuth Access]”

* **Phishing Link (obfuscated)**:

https://github.com.login.security-check[.]app/authorize

**3. Malicious Behavior**

Upon clicking the phishing link:

* The user is redirected to a fake GitHub login page.
* Credentials and OAuth authorization are captured.
* The attacker uses the captured token to:
  + Access private repositories.
  + Clone or exfiltrate source code.
  + Modify repository content.
  + Create backdoor access (e.g., adding SSH keys or actions secrets).

**4. Indicators of Compromise (IOCs)**

| **Type** | **Indicator** |
| --- | --- |
| Domain | github-security-check[.]app |
| IP Address | 185.213.211.12 (phishing host) |
| URL Path | /authorize |
| Email Address | noreply@github-security.com *(spoofed)* |

**5. Mitigation Actions Taken**

* Blocked the phishing domain on email gateway and proxy filters.
* Reported the phishing page to hosting provider and GitHub.
* Alerted affected users and reset compromised credentials.
* Reviewed OAuth app authorizations on impacted accounts.
* Conducted internal threat hunt for malicious commits or exfiltration.
* Updated phishing detection rules and shared IOCs with threat intel partners.

**6. Recommendations**

* **User Awareness**: Conduct targeted phishing training focusing on OAuth scams.
* **OAuth Hygiene**: Regularly audit authorized third-party GitHub applications.
* **MFA Enforcement**: Ensure multi-factor authentication is enabled for all users.
* **Email Security**: Improve SPF, DKIM, and DMARC enforcement.
* **Incident Response**: Enhance playbooks to cover OAuth abuse vectors.

**7. Conclusion**

This incident highlights the increasing sophistication of OAuth-based phishing attacks and the need for vigilant user behavior and strong account hygiene on code-hosting platforms like GitHub. Continuous monitoring and user education remain essential to prevent similar threats.