GenAI Ethical Hacking Report

Target: github.com
Generated: 2025-08-31 22:00

■ Nmap Results

{'20.207.73.82': {'state': 'up', 'os': [{'name': 'FreeBSD 13.0-RELEASE', 'accuracy': '91'

■ AI Analysis of Nmap

Nmap Scan Result Analysis: 20.207.73.82

Services and Versions:

* **Operating System:** FreeBSD 13.0-RELEASE (91% accuracy)
* **Port 22:** SSH - Version unknown.
* **Port 80:** HAProxy http proxy - Version 2.0.0 or later.
* **Port 443:** HAProxy http proxy - Version 2.0.0 or later.

Possible Vulnerabilities:

The lack of precise version information for both SSH and HAProxy significantly hinders ac

* **SSH (Port 22):** Without a specific version, we cannot pinpoint exact CVEs. However

* **HAProxy (Ports 80 & 443):** HAProxy itself is generally considered secure when proper
    * **Configuration flaws:** Incorrectly configured HAProxy can lead to denial-of-serv
    * **Outdated versions:** Older HAProxy versions are known to have various security fl
    * **Logic vulnerabilities:** Specific vulnerabilities may exist based on the impleme
    * **Improper handling of HTTP requests:** Vulnerabilities related to HTTP request pro

    Risk Level: Medium (to High depending on HAProxy version and configuration)

Exploitation Tools/Methods (Hypothetical, pending version verification):

* **SSH:**
    * **Brute-force attacks:** Tools like Hydra, Medusa, or Ncrack could be used to atte
    * **Exploit databases:** If a specific vulnerable SSH version is identified, Metaspl

* **HAProxy:**
    * **Nessus/OpenVAS scans:** These vulnerability scanners could help identify potenti
    * **Manual review of HAProxy configuration:** This is crucial to identify any misconf
    * **Penetration testing frameworks (Metasploit):** Metasploit may contain exploits i

Overall Risk Assessment:

The overall risk level is **Medium to High**, depending heavily on the exact versions of

■ theHarvester Results

{}

■ AI Analysis of theHarvester

AI analysis unavailable.

■■ Sublist3r Results

- www.github.com
-
- api.github.com
-
- archiveprogram.github.com
-
- atom-installer.github.com
-
- bounty.github.com
-
- branch.github.com
-
- brand.github.com
-
- brandguide.github.com
-
- camo.github.com
-
- central.github.com
-
- cla.github.com
-
- classroom.github.com
-
- cli.github.com
-
- cloud.github.com
-
- f.cloud.github.com
-
- code.github.com
-
- codeload.github.com
-
- codeql.github.com
-

- codespaces.github.com
-
- codespaces-dev.github.com
-
- codespaces-ppe.github.com
-
- communication.github.com
-
- www.communication.github.com
-
- m.communication.github.com
-
- res.communication.github.com
-
- t.communication.github.com
-
- community.github.com
-
- desktop.github.com
-
- dev.github.com
-
- docs.github.com
-
- docs-front-door.github.com
-
- dodgeball.github.com
-
- edu.github.com
-
- education.github.com
-
- emails.github.com
-
- enterprise.github.com
-
- support.enterprise.github.com

-
- www.support.enterprise.github.com
-
- examadmin.github.com
-
- examadmin-uat.github.com
-
- examregistration.github.com
-
- examregistration-api.github.com
-
- examregistration-uat.github.com
-
- examregistration-uat-api.github.com
-
- fast.github.com
-
- forgoodfirstissue.github.com
-
- fortawesome.github.com
-
- galaxy.github.com
-
- garage.github.com
-
- gist.github.com
-
- github.github.com
-
- government.github.com
-
- graphql.github.com
-
- www.graphql.github.com
-
- graphql-stage.github.com
-

- www.graphql-stage.github.com
-
- help.github.com
-
- helpnext.github.com
-
- hq.github.com
-
- hubot.github.com
-
- vpn-ca.iad.github.com
-
- id.github.com
-
- import.github.com
-
- import2.github.com
-
- importer2.github.com
-
- innovationgraph.github.com
-
- jira.github.com
-
- www.jira.github.com
-
- jobs.github.com
-
- lab.github.com
-
- lab-sandbox.github.com
-
- learn.github.com
-
- mac-installer.github.com
-
- mailing.github.com

-
- maintainers.github.com
-
- www.maintainers.github.com
-
- api.mcp.github.com
-
- nonprofits.github.com
-
- octicons.github.com
-
- octodex.github.com
-
- octostatus-production.github.com
-
- octoverse.github.com
-
- offer.github.com
-
- pages.github.com
-
- partner.github.com
-
- partnerportal.github.com
-
- www.partnerportal.github.com
-
- pkg.github.com
-
- maven.pkg.github.com
-
- npm.pkg.github.com
-
- nuget.pkg.github.com
-
- porter.github.com
-

- porter2.github.com
-
- proxima-review-lab.github.com
-
- raw.github.com
-
- registry.github.com
-
- render.github.com
-
- render-lab.github.com
-
- www.render-lab.github.com
-
- resources.github.com
-
- review-lab.github.com
-
- octocaptcha.review-lab.github.com
-
- rs.github.com
-
- schrauger.github.com
-
- api.security.github.com
-
- www.api.security.github.com
-
- securitylab.github.com
-
- skills.github.com
-
- skyline.github.com
-
- www.skyline.github.com
-
- slack.github.com

-
- smtp.github.com
-
- www.smtp.github.com
-
- socialimpact.github.com
-
- staging-lab.github.com
-
- stars.github.com
-
- api.stars.github.com
-
- www.api.stars.github.com
-
- status.github.com
-
- stg.github.com
-
- styleguide.github.com
-
- support.github.com
-
- ws.support.github.com
-
- www.ws.support.github.com
-
- talks.github.com
-
- teams.github.com
-
- training.github.com
-
- visualstudio.github.com
-
- www.visualstudio.github.com
-

- vscode.github.com
-
- vscode-auth.github.com
-
- wellarchitected.github.com
-
- workspaces.github.com
-
- workspaces-dev.github.com
-
- workspaces-ppe.github.com
-

■ AI Analysis of Sublist3r

```
Analysis of Sublist3r Output for github.com

The Sublist3r output reveals a substantial number of subdomains for github.com.  Let's ca

**Interesting/Vulnerable-Looking Subdomains:**

Several subdomains warrant closer examination due to their names or potential for misconf

* **`api.*` subdomains:**  `api.github.com`, `examregistration-api.github.com`, `api.mcp.

* **`staging` and `test` environments:** `codespaces-dev.github.com`, `codespaces-ppe.git

* **`*.github.com` internal subdomains:** Several internal-looking subdomains might revea

* **`*.github.com` subdomains with "admin" in the name:** `examadmin.github.com` and rela


**Risk Level:**

The overall risk level is **Medium to High**. While GitHub is a well-secured platform, th


**Next Recon Steps:**

1. **Prioritize Targets:** Focus initial investigation on the `api.*`, `staging`, `test`,

2. **Passive Reconnaissance:** Perform further passive reconnaissance using tools like:
    * **DNS enumeration:** Use tools beyond Sublist3r (e.g., amass, dnsrecon) to find any
    * **Port scanning:** Identify open ports on each high-priority subdomain using Nmap o
    * **Service versioning:** Identify versions of services running on open ports to chec

3. **Active Reconnaissance (with caution and permission):** If authorized, perform target
    * **Vulnerability scanning:** Use automated vulnerability scanners (e.g., Nessus, Ope
    * **Directory brute-forcing:** Attempt to uncover hidden directories and files on web
    * **Parameter tampering:** Test APIs for vulnerabilities such as SQL injection or Cro

4. **Takeover Checks:** Conduct takeover checks for each subdomain (especially lesser-kno
    * **Takeover tools:** These tools automatically check for vulnerable services (e.g.,

5. **Manual Review:** Examine website source code, HTTP headers, and other metadata for p

**Important Note:**  Always obtain explicit permission before conducting any active recon
```

■ SQL Injection Results

```
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.9.8.5#dev}
|_ -| . [(]     | .'| . |
|___|_  ['']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent

[*] starting @ 21:58:04 /2025-08-31/

[21:58:04] [WARNING] using 'C:\Users\GOWTHAM\Downloads\geminiapi (2)\geminiapi\sqlmap_res
do you want to check for the existence of site's sitemap(.xml) [y/N] N
[21:58:04] [INFO] starting crawler for target URL 'http://github.com'
[21:58:04] [INFO] searching for links with depth 1
got a 301 redirect to 'https://github.com/'. Do you want to follow? [Y/n] Y
do you want to normalize crawling results [Y/n] Y
do you want to store crawling results to a temporary file for event...
```

■ AI Analysis of SQL Injection

Analysis of SQLmap Output:

The SQLmap test against `http://github.com` did *not* find any SQL injection vulnerabilit

Vulnerable Parameters/Endpoints:

None found.

CVEs or Known Exploits:

Not applicable, as no vulnerabilities were identified.

Risk Level:

Low. The absence of SQL injection vulnerabilities in the tested parameters indicates a l

Important Considerations:

* **Limited Scope:** The scan only tested a small subset of parameters on one URL. GitHub
* **False Negatives:** While SQLmap is powerful, it's not infallible. Sophisticated web
* **Parameter Exhaustiveness:** The output doesn't guarantee that *all* GET parameters on
* **Other Vulnerabilities:** The absence of SQL injection doesn't rule out other vulnera

Recommendations:

To obtain a more comprehensive understanding of GitHub's security posture, far more exten

* **Expanding the test scope:** Test a broader range of URLs and parameters.
* **Increasing the testing intensity:** Use higher levels and risk settings within SQLmap
* **Employing other vulnerability scanners:** Use tools designed to detect different type
* **Manual penetration testing:** A skilled security expert performing manual penetratio

In conclusion, based solely on this SQLmap output, the risk of SQL injection on the speci

■ Subdomain Enumeration Results

{'subdomains': ['[i] Enumerating 10 candidates for github.com (DNS available: True, HTTP

■ AI Analysis of Subdomain Enumeration

Analysis of Subdomain Enumeration Output for github.com

The output reveals a partial subdomain enumeration scan for `github.com`. Let's break do

1. Useful Subdomains:

* `www.github.com`: The main website, expected and unsurprising.
* `api.github.com`: The GitHub API endpoint. Crucial for application functionality and a
* `admin.github.com`: This is extremely concerning. If this resolves to an actual admin

2. Risky Infrastructure Exposures:

* **`admin.github.com` Resolution:** The most significant risk. A publicly resolvable `a
* **Unreachable Subdomains:** While many subdomains are unreachable (`ftp`, `dev`, `stagi
* **`test.github.com`:** Although reachable, its IP address (192.0.2.1) is a private IP a

3. Risk Level:

The risk level is **HIGH** due to the presence of a resolvable `admin.github.com` subdoma

4. Suggested Next Actions:

* **Immediate Action:** Investigate `admin.github.com` thoroughly. Try accessing it via
* **Investigate Unreachable Subdomains:** Determine the intended status of the unreachabl
* **Analyze `test.github.com`:** Determine the purpose of the `test.github.com` subdomain
* **Expand Enumeration:** Conduct a more comprehensive subdomain enumeration using variou
* **Port Scanning:** Perform a port scan on the IPs resolved for `www.github.com`, `api.g
* **Vulnerability Scanning:** Conduct a vulnerability scan on identified subdomains and s

Disclaimer: This analysis assumes the provided output is genuine. Analyzing a real-w

■ Final Overall AI Assessment

■ AI Analysis of Overall Findings

GitHub Reconnaissance Results Security Assessment

This report summarizes the security findings from reconnaissance scans of GitHub infrastr

I. Main Security Weaknesses:

* **Extensive Subdomain Exposure:** Multiple tools (`theharvester`, `subdomain_enum`, `s

* **Potential for SQL Injection:** The `sql_injection` scan, while ultimately inconclusiv

* **Open Ports on Production Server:** `nmap` identified open ports 22 (SSH), 80 (HTTP),

II. Attack Paths (Recon → Exploitation):

1. **Subdomain Enumeration → Lateral Movement:** An attacker could use tools like `sublis

2. **Subdomain Enumeration → Credential Harvesting:** Less-secured subdomains could host

3. **Open SSH Port → Unauthorized Access:** The open SSH port on `www.github.com` allows

4. **SQL Injection (Attempted) → Data Breach:** Although the `sql_injection` scan didn't

III. Overall Risk Score:

High. The combination of extensive subdomain exposure, an open SSH port on a producti

IV. Prioritized Mitigation Recommendations:

1. **Reduce Attack Surface:** Conduct a thorough inventory of all subdomains and decommi

2. **Secure SSH Access:** Immediately restrict SSH access to the production server (`20.

3. **Improve Input Validation:** Enhance input validation and sanitization across all ap

4. **Vulnerability Scanning and Penetration Testing:** Conduct regular vulnerability scan

5. **Security Information and Event Management (SIEM):** Implement a SIEM solution to mon

6. **Security Awareness Training:** Train employees on security best practices, including

7. **Regular Security Audits:** Conduct regular internal and external security audits to

These recommendations should be prioritized based on their potential impact and feasibili

■ Consolidated Tool Results

■ Theharvester Results

{'emails': [], 'hosts': ['2Fbrand.github.com', '2Fdocs.github.com', '2Feducation.github.c

■ AI Analysis of Theharvester

This scan result reveals a list of GitHub-related hostnames. The absence of emails and d

Risks Identified:

* **Exposure of Internal Infrastructure:** The inclusion of numerous internal-looking hos
* **Subdomain Takeover:** Some of the less obvious hostnames could be vulnerable to subdo
* **Data Leaks via Misconfigured Services:** The presence of internal-looking names impl
* **Potential for Credential Stuffing:** While no email addresses were found, the sheer v

Exploits and Mitigations:

The exact exploits are impossible to determine without further information. However, gen

* **Intrusion Detection/Prevention Systems (IDS/IPS):** Implement robust IDS/IPS to monit
* **Regular Vulnerability Scanning:** Conduct regular automated vulnerability scans of al
* **Subdomain Enumeration and Takeover Protection:** Regularly scan for potentially vuln
* **Secure Configuration of Services:** Ensure all services running on these hosts are s
* **Network Segmentation:** Implement network segmentation to isolate internal services
* **Web Application Firewall (WAF):** Deploy a WAF to protect against common web applicat
* **Monitoring and Logging:** Implement comprehensive monitoring and logging to track ac

Important Note: This analysis is based solely on the provided scan result. It doesn'

■ Subdomain_enum Results

{'subdomains': ['[i] Enumerating 10 candidates for github.com (DNS available: True, HTTP

■ AI Analysis of Subdomain_enum

```
## Analysis of Subdomain Enumeration Output for github.com

The output reveals a partial subdomain enumeration scan for `github.com`.  Let's break do

**1. Useful Subdomains:**

* `www.github.com`: The main website, expected and unsurprising.
* `api.github.com`: The GitHub API endpoint.  Crucial for application functionality and a
* `admin.github.com`:  This is extremely concerning.  If this resolves to an actual admin

**2. Risky Infrastructure Exposures:**

* **`admin.github.com` Resolution:** The most significant risk.  A publicly resolvable `a
* **Unreachable Subdomains:** While many subdomains are unreachable (`ftp`, `dev`, `stagi
* **`test.github.com`:** Although reachable, its IP address (192.0.2.1) is a private IP a


**3. Risk Level:**

The risk level is **HIGH** due to the presence of a resolvable `admin.github.com` subdoma

**4. Suggested Next Actions:**

* **Immediate Action:** Investigate `admin.github.com` thoroughly.  Try accessing it via '
* **Investigate Unreachable Subdomains:** Determine the intended status of the unreachabl
* **Analyze `test.github.com`:** Determine the purpose of the `test.github.com` subdomain
* **Expand Enumeration:** Conduct a more comprehensive subdomain enumeration using variou
* **Port Scanning:** Perform a port scan on the IPs resolved for `www.github.com`, `api.g
* **Vulnerability Scanning:** Conduct a vulnerability scan on identified subdomains and s


**Disclaimer:** This analysis assumes the provided output is genuine.  Analyzing a real-w
```

■ Sql_injection Results

```
{'output': "        ___\n       __H__\n ___ ___[,]_____ ___ ___  {1.9.8.5#dev}\n|_ -| . [
```

■ AI Analysis of Sql_injection

Analysis of SQLmap Output:

The SQLmap test against `http://github.com` did *not* find any SQL injection vulnerabilit

Vulnerable Parameters/Endpoints:

None found.

CVEs or Known Exploits:

Not applicable, as no vulnerabilities were identified.

Risk Level:

Low. The absence of SQL injection vulnerabilities in the tested parameters indicates a l

Important Considerations:

* **Limited Scope:** The scan only tested a small subset of parameters on one URL. GitHub
* **False Negatives:** While SQLmap is powerful, it's not infallible. Sophisticated web
* **Parameter Exhaustiveness:** The output doesn't guarantee that *all* GET parameters on
* **Other Vulnerabilities:** The absence of SQL injection doesn't rule out other vulnera

Recommendations:

To obtain a more comprehensive understanding of GitHub's security posture, far more exten

* **Expanding the test scope:** Test a broader range of URLs and parameters.
* **Increasing the testing intensity:** Use higher levels and risk settings within SQLmap
* **Employing other vulnerability scanners:** Use tools designed to detect different type
* **Manual penetration testing:** A skilled security expert performing manual penetratio

In conclusion, based solely on this SQLmap output, the risk of SQL injection on the speci

■ Nmap Results

{'20.207.73.82': {'state': 'up', 'os': [{'name': 'FreeBSD 13.0-RELEASE', 'accuracy': '91'

■ AI Analysis of Nmap

```
## Nmap Scan Result Analysis: 20.207.73.82

**Services and Versions:**

* **Operating System:** FreeBSD 13.0-RELEASE (91% accuracy)
* **Port 22:** SSH - Version unknown.
* **Port 80:** HAProxy http proxy - Version 2.0.0 or later.
* **Port 443:** HAProxy http proxy - Version 2.0.0 or later.


**Possible Vulnerabilities:**

The lack of precise version information for both SSH and HAProxy significantly hinders ac

* **SSH (Port 22):**  Without a specific version, we cannot pinpoint exact CVEs.  However

* **HAProxy (Ports 80 & 443):** HAProxy itself is generally considered secure when proper
    * **Configuration flaws:**  Incorrectly configured HAProxy can lead to denial-of-serv
    * **Outdated versions:** Older HAProxy versions are known to have various security fl
    * **Logic vulnerabilities:**  Specific vulnerabilities may exist based on the impleme
    * **Improper handling of HTTP requests:** Vulnerabilities related to HTTP request pro

    **Risk Level: Medium (to High depending on HAProxy version and configuration)**


**Exploitation Tools/Methods (Hypothetical, pending version verification):**

* **SSH:**
    * **Brute-force attacks:**  Tools like Hydra, Medusa, or Ncrack could be used to atte
    * **Exploit databases:**  If a specific vulnerable SSH version is identified, Metaspl


* **HAProxy:**
    * **Nessus/OpenVAS scans:**  These vulnerability scanners could help identify potenti
    * **Manual review of HAProxy configuration:** This is crucial to identify any misconf
    * **Penetration testing frameworks (Metasploit):**  Metasploit may contain exploits i


**Overall Risk Assessment:**

The overall risk level is **Medium to High**, depending heavily on the exact versions of
```

■ Sublist3r Results

```
{'subdomains': ['www.github.com', '', 'api.github.com', '', 'archiveprogram.github.com',
```

■ AI Analysis of Sublist3r

```
Analysis of Sublist3r Output for github.com

The Sublist3r output reveals a substantial number of subdomains for github.com.  Let's ca

**Interesting/Vulnerable-Looking Subdomains:**

Several subdomains warrant closer examination due to their names or potential for misconf

* **`api.*` subdomains:**  `api.github.com`, `examregistration-api.github.com`, `api.mcp.

* **`staging` and `test` environments:** `codespaces-dev.github.com`, `codespaces-ppe.git

* **`*.github.com` internal subdomains:** Several internal-looking subdomains might revea

* **`*.github.com` subdomains with "admin" in the name:** `examadmin.github.com` and rela


**Risk Level:**

The overall risk level is **Medium to High**. While GitHub is a well-secured platform, th


**Next Recon Steps:**

1. **Prioritize Targets:** Focus initial investigation on the `api.*`, `staging`, `test`,

2. **Passive Reconnaissance:** Perform further passive reconnaissance using tools like:
    * **DNS enumeration:** Use tools beyond Sublist3r (e.g., amass, dnsrecon) to find any
    * **Port scanning:** Identify open ports on each high-priority subdomain using Nmap o
    * **Service versioning:** Identify versions of services running on open ports to chec

3. **Active Reconnaissance (with caution and permission):** If authorized, perform target
    * **Vulnerability scanning:** Use automated vulnerability scanners (e.g., Nessus, Ope
    * **Directory brute-forcing:** Attempt to uncover hidden directories and files on web
    * **Parameter tampering:** Test APIs for vulnerabilities such as SQL injection or Cro

4. **Takeover Checks:** Conduct takeover checks for each subdomain (especially lesser-kno
    * **Takeover tools:** These tools automatically check for vulnerable services (e.g.,

5. **Manual Review:** Examine website source code, HTTP headers, and other metadata for p

**Important Note:**  Always obtain explicit permission before conducting any active recon
```