GenAI Ethical Hacking Report

Target: pesce.ac.in

Generated: 2025-09-01 11:59

■ Nmap Results

{'68.178.161.81': {'state': 'up', 'os': [{'name': 'Linux 3.10 - 4.11', 'accuracy': '86',

■ AI Analysis of ■ Nmap Results

Nmap Scan Result Analysis: 68.178.161.81 This Nmap scan reveals a Linux server (likely 3.10-4.14 kernel) with several services running, some of which present potential vulnerabilities. **Identified Services and Versions:** * **FTP (Port 21):** ProFTPD (version unspecified). This is a significant risk if not properly configured. * **SSH (Port 22):** OpenSSH 8.0. Requires further investigation for misconfigurations. * **SMTP (Port 25):** Filtered. * **DNS (Port 53):** PowerDNS Authoritative Server 4.9.5. Requires further investigation for misconfigurations. * **HTTP (Port 80):** Apache httpd (version unspecified). Requires further investigation for outdated versions and misconfigurations. * **POP3 (Port 110):** Dovecot pop3d (version unspecified). * **RPCbind (Port 111):** Version 2-4. Older versions have known vulnerabilities. * **IMAP (Port 143):** Dovecot imapd (version unspecified). * **HTTPS (Port 443):** Apache httpd (version unspecified). * **SMTP (Port 465 & 587):** Exim smtpd 4.98.2. Requires checking for known vulnerabilities in this version. * **IMAPS (Port 993):** Version unspecified. * **POP3S (Port 995):** Version unspecified. * **MySQL (Port 3306):** MySQL (version unspecified). A major risk if default credentials or weak passwords are used. **Possible Vulnerabilities and Risk Levels:** * **FTP (Port 21):** ProFTPD has a history of vulnerabilities. Without a version, it's impossible to pinpoint specific CVEs. **Risk Level: Medium to High** (depending on the version and configuration). Exploitation would involve attempting common exploits for ProFTPD or brute-forcing credentials. * **SSH (Port 22):** OpenSSH 8.0 might have vulnerabilities depending on the specific sub-version and configuration. Regular security updates are crucial. **Risk Level: Medium**. Vulnerability checks need further investigation using tools like Nessus or OpenVAS, and potentially exploiting any found vulnerabilities. * **PowerDNS (Port 53):** Requires checking for known vulnerabilities in version 4.9.5 via vulnerability databases (e.g., NVD). **Risk Level: Low to Medium**. Misconfigurations or unpatched versions pose the biggest threat. * **Apache HTTP (Port 80 & 443):** Outdated versions of Apache are susceptible to numerous attacks (e.g., RCE, XSS). Requires version identification and vulnerability scanning. **Risk Level: Medium to High**. Exploitation tools will depend on the found vulnerabilities. * **Dovecot (Port 110, 143, 993, 995):** Version identification is critical to assess vulnerabilities. **Risk Level: Medium**. Exploitation may involve credential stuffing or exploiting known vulnerabilities if outdated. * **RPCbind (Port 111):** Older versions of rpcbind have known vulnerabilities allowing for remote code execution. **Risk Level: Medium to High**. Requires version verification and specific vulnerability checks. * **Exim SMTP (Port 465 & 587):** Version 4.98.2 needs to be checked against CVE databases for any known exploits. **Risk Level: Medium**. * **MySQL (Port 3306):** Default credentials or weak passwords are the biggest risk. **Risk Level: High**. Tools like sqlmap can be used to attempt exploitation if vulnerable. **Suggested Exploitation Tools and Methods:** The specific tools and methods depend heavily on the identified vulnerabilities. The following are general approaches: * **Nmap scripts:** Use Nmap's scripting engine to perform more in-depth service version detection and vulnerability checks. * **Nessus/OpenVAS:** Conduct a full vulnerability scan to identify specific CVEs applicable to the discovered versions. * **Metasploit:** Once vulnerabilities are identified, Metasploit can be used to attempt exploitation. * **Burp Suite:** For web application vulnerabilities (Apache), Burp Suite is invaluable for identifying and exploiting XSS, SQL injection, and other web-related flaws. * **sqlmap:** Specifically for MySQL, sqlmap can be used to check for SQL injection vulnerabilities. * **Hydra/John the Ripper:** For brute-forcing passwords against FTP, SSH, POP3, IMAP, and MySQL. **Next Steps:** 1. **Version Identification:** Determine the precise versions of all the services. 2. **Vulnerability Scanning:** Use Nessus or OpenVAS to identify known vulnerabilities based on the versions. 3. **Manual Testing:** Based on the vulnerabilities identified, conduct manual testing to confirm and exploit them (ethically and with permission). 4. **Credential Brute-forcing:** Only attempt this with proper authorization and after exhausting other options. This analysis provides a starting point. A thorough penetration test requires further investigation and confirmation of the identified vulnerabilities. Remember to always obtain explicit permission before performing any penetration testing activities.

■ theHarvester Results

```
{}
```

■ AI Analysis of ■ theHarvester Results

AI analysis unavailable.

■■ Sublist3r Results

- www.pesce.ac.in

-

- alumni.pesce.ac.in

-

- www.alumni.pesce.ac.in

-

- mail.pesce.ac.in

-

■ AI Analysis of ■■ Sublist3r Results

Analyzing the Sublist3r output: The output shows a small number of subdomains for `pesce.ac.in`. Let's break down each one: * **`www.pesce.ac.in`**: This is the main website. It's expected and requires basic security assessment. * **`alumni.pesce.ac.in`**: This subdomain likely hosts content specifically for alumni. It could have different security configurations than the main site, making it potentially more vulnerable. * **`www.alumni.pesce.ac.in`**: This is a `www` subdomain for the alumni site. Redundant, but needs investigation just like `alumni.pesce.ac.in`. * **`mail.pesce.ac.in`**: This subdomain likely hosts email services. Email servers are frequent targets for attacks, so this is a high-priority target for further investigation. A compromised email server could lead to significant data breaches. * **Empty Strings (`''`):** These are likely artifacts of Sublist3r's output and can be ignored. **Interesting/Vulnerable-Looking Subdomains:** * `mail.pesce.ac.in` (High risk) * `alumni.pesce.ac.in` (Medium risk) **Staging/Test Environments:** None are explicitly identified in this limited output. However, further investigation may reveal test or staging environments using techniques like directory brute-forcing or searching for common naming patterns (e.g., `test.pesce.ac.in`, `stage.pesce.ac.in`, `dev.pesce.ac.in`). **Risk Level:** Overall, the risk is moderate. The presence of a `mail` subdomain raises the risk level significantly due to the potential impact of a compromise. **Suggested Next Recon Steps:** 1. **Port Scanning:** Perform a port scan on all identified subdomains to identify open ports and services running. This will help determine the technologies used and potential vulnerabilities. Nmap is a good tool for this. 2. **Service Versioning:** Identify versions of services running on open ports. Outdated software is often vulnerable. 3. **Vulnerability Scanning:** Conduct automated vulnerability scans on each subdomain using tools like OpenVAS or Nessus. Focus on known vulnerabilities for email servers (if `mail.pesce.ac.in` is vulnerable) and common web application vulnerabilities. 4. **Takeover Checks:** Use tools like `takeover` (GitHub) to check for potential subdomain takeovers. This involves identifying if the subdomain points to a service that is easily compromised. 5. **Directory Brute-forcing:** Attempt to brute-force directories on `alumni.pesce.ac.in` and `www.alumni.pesce.ac.in` to uncover hidden files or directories, potentially revealing sensitive information or vulnerabilities. DirBuster or Gobuster are useful for this. 6. **Subdomain Brute-forcing:** Since the initial enumeration was limited, consider expanding it with a more thorough brute-force subdomain enumeration using tools like Subfinder, amass, or a combination. 7. **Passive Reconnaissance:** Utilize tools like Shodan to gather more information about the services running on these subdomains. Shodan can reveal exposed information and services. By following these steps, you can build a more complete picture of the target's security posture and identify potential vulnerabilities. Remember to always obtain proper authorization before conducting any security testing.

■ SQL Injection Results

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

■ AI Analysis of ■ SQL Injection Results

The provided SQLmap output shows that **no SQL injection vulnerabilities were found**. **Analysis:** The key line is: `[11:57:21] [WARNING] no usable links found (with GET parameters)` This indicates that sqlmap, after crawling the target URL (`http://pesce.ac.in`), did not identify any GET parameters that could be exploited for SQL injection. The crawler examined the site for links with GET parameters (e.g., `http://example.com/page?id=123`) as these are common locations for SQL injection vulnerabilities. The absence of usable links implies that there were no apparent injection points tested. **Vulnerable Parameters/Endpoints:** None identified. The report explicitly states that no usable links (with GET parameters) were found. This does *not* rule out the possibility of vulnerabilities using other attack vectors (POST parameters, blind SQL injection, etc.), but the scan using GET parameters did not reveal any. **CVEs or Known Exploits:** Not applicable, as no vulnerabilities were found in this specific test. **Risk Level:** Based solely on this SQLmap output, the risk level concerning SQL injection via GET parameters is **Low**. However, this is a limited assessment. A comprehensive security assessment would require testing for other types of vulnerabilities, including: * **SQL injection via POST parameters:** SQLmap should be run again testing POST requests. * **Blind SQL injection:** This technique is harder to detect automatically and needs specialized testing. * **Other vulnerabilities:** A complete penetration test would examine XSS, CSRF, authentication flaws, and other potential weaknesses. **Conclusion:** This SQLmap scan provides only partial information. While it didn't find SQL injection vulnerabilities in GET parameters, it's insufficient to conclude the complete absence of vulnerabilities on the target website. Further testing is strongly recommended.

■ Subdomain Enumeration Results

- [i] Enumerating 10 candidates for pesce.ac.in (DNS available: True, HTTP fallback: True)
- [-] staging.pesce.ac.in DNS error: NXDOMAIN | HTTP: unreachable
- [-] beta.pesce.ac.in DNS error: NXDOMAIN | HTTP: unreachable
- [-] dev.pesce.ac.in DNS error: NXDOMAIN | HTTP: unreachable
- [-] portal.pesce.ac.in DNS error: NXDOMAIN | HTTP: unreachable
- [-] test.pesce.ac.in DNS error: NXDOMAIN | HTTP: unreachable
- [-] admin.pesce.ac.in DNS error: NXDOMAIN | HTTP: unreachable
- [-] api.pesce.ac.in DNS error: NXDOMAIN | HTTP: unreachable
- [+] www.pesce.ac.in -> ['68.178.161.81'] | HTTP: 200
- [+] ftp.pesce.ac.in -> ['103.117.212.198'] | HTTP: 200
- [+] mail.pesce.ac.in -> ['68.178.161.81'] | HTTP: 200
- == Summary ==
- DNS-resolved: 3 / 10
- HTTP reachable: 3 / 10

■ AI Analysis of ■ Subdomain Enumeration Results

Analysis of Subdomain Enumeration Output for pesce.ac.in The output reveals a partial success in enumerating subdomains for `pesce.ac.in`. Let's break down the findings: **1. Useful Subdomains:** * **`www.pesce.ac.in`:** The main website. This is crucial for understanding the organization's online presence. * **`ftp.pesce.ac.in`:** An FTP server is exposed. This could potentially allow unauthorized access to files if not properly secured. * **`mail.pesce.ac.in`:** A mail server. This is a critical asset that needs strong security measures. **2. Risky Infrastructure Exposures:** * **FTP Server (`ftp.pesce.ac.in`):** FTP is an insecure protocol. The mere existence of an exposed FTP server is a significant risk, especially if it's not using strong authentication and encryption (like FTPS or SFTP). An attacker could potentially upload malicious files or download sensitive data. * **Multiple Subdomains Resolving to the Same IP (`www.pesce.ac.in` and `mail.pesce.ac.in` sharing `68.178.161.81`):** While not inherently risky, this indicates potential vulnerabilities if misconfiguration leads to compromised access affecting both services. * **Lack of Subdomain Security:** The failure to resolve most attempted subdomains (e.g., `staging`, `beta`, `dev`, `test`, `admin`, `api`) suggests a lack of robust subdomain management. This could indicate a lack of proper security controls and monitoring. While this might seem "safe" in that no assets are directly found, it suggests gaps in the organization's security posture. **3. Risk Level:** I'd rate the overall risk level as **Medium to High**. The presence of an insecure FTP server and the potential for misconfiguration across shared IP addresses are major concerns. The lack of other subdomains also highlights potential blind spots for security monitoring. **4. Suggested Next Actions:** * **Prioritize FTP Server Security:** Conduct a thorough security assessment of the `ftp.pesce.ac.in` server. This includes verifying authentication methods, checking for weak passwords, and exploring the possibility of migrating to a more secure protocol (FTPS or SFTP). Consider disabling the FTP server entirely if not absolutely necessary. * **Investigate Shared IP Address:** Determine why `www.pesce.ac.in` and `mail.pesce.ac.in` share the same IP. This could be a deliberate design choice, but it warrants further investigation to rule out potential vulnerabilities due to misconfiguration or a compromised server affecting both services. * **Expand Subdomain Enumeration:** Use more sophisticated tools and techniques to attempt a more comprehensive subdomain enumeration. This could uncover additional subdomains that might expose further vulnerabilities. Consider using techniques like brute forcing, recursive DNS queries, and crawling the main website for clues. * **Vulnerability Scanning:** Perform vulnerability scans on the identified servers (`68.178.161.81` and `103.117.212.198`) to identify any known vulnerabilities. Tools like Nessus, OpenVAS, or QualysGuard can assist with this. * **Web Application Security Testing:** Conduct a thorough web application security assessment of `www.pesce.ac.in` to identify potential vulnerabilities like Cross-Site Scripting (XSS), SQL Injection, and others. * **Reconnaissance on the Identified IPs:** Perform additional reconnaissance on the identified IP addresses using tools like Nmap to identify open ports and services. This will help pinpoint potential attack vectors. By addressing these issues, the organization can significantly reduce its attack surface and improve its overall security posture. The findings suggest a need for more proactive security management and incident response planning.

■ Final Overall AI Assessment

■ AI Analysis of Overall Findings

Security Assessment of pesce.ac.in Based on the provided reconnaissance data, here's a summary of the security assessment: **1. Main Security Weaknesses:** * **Open and Vulnerable Ports:** Nmap scan reveals numerous open ports on IP 68.178.161.81 (hosting www.pesce.ac.in and mail.pesce.ac.in), including FTP (port 21), SSH (port 22), HTTP (ports 80 & 443), SMTP (ports 25, 465, 587), POP3 (ports 110 & 995), IMAP (ports 143 & 993), and MySQL (port 3306). The lack of version information for some services hinders precise vulnerability assessment, but open ports without proper access controls present significant risks. The presence of older versions (like OpenSSH 8.0) should be investigated for known vulnerabilities. * **Potentially Vulnerable Web Application:** While sqlmap didn't directly identify SQL injection vulnerabilities, the lack of usable links found suggests a poorly structured web application. Further investigation is needed, including manual testing for SQLi, XSS, and other vulnerabilities. * **Lack of Subdomain Protection:** Many attempted subdomain enumerations (e.g., staging, beta, dev, test, admin, api) failed due to NXDOMAIN, suggesting a lack of proper subdomain protection. However, the existence of `alumni.pesce.ac.in` indicates a potential for further subdomain enumeration. This poor subdomain hygiene indicates a potential for overlooked or misconfigured services. **2. Attack Paths (Recon → Exploitation):** * **Path 1 (Port Scanning & Exploitation):** Recon (Nmap) identifies open ports. An attacker could exploit vulnerabilities on these open ports (e.g., outdated SSH server, weak MySQL credentials, known vulnerabilities in Apache or ProFTPD). * **Path 2 (Subdomain Enumeration & Lateral Movement):** Sublist3r and subdomain enumeration tools identify subdomains. Successful exploitation of one subdomain could lead to lateral movement within the network. * **Path 3 (Web Application Attack):** Although sqlmap didn't find SQLi in this run, further manual testing is needed. Successful exploitation of web application vulnerabilities (e.g., SQL injection, Cross-Site Scripting) could allow for data breaches or server compromise. **3. Overall Risk Score:** **Medium** The presence of multiple open ports, potential web application vulnerabilities, and weak subdomain protection warrant a medium-risk rating. While no critical vulnerabilities were definitively identified in this scan, the potential for exploitation is significant. **4. Prioritized Mitigation Recommendations:** * **Prioritize Patching and Hardening:** Immediately update all software on servers with open ports to their latest versions. Implement strong password policies and multi-factor authentication (MFA) where possible. Restrict access to essential ports only. * **Web Application Security Assessment:** Conduct a thorough penetration test of the web application to identify and remediate any vulnerabilities (SQLi, XSS, CSRF, etc.). Implement a robust Web Application Firewall (WAF). * **Subdomain Management:** Implement stricter subdomain management to prevent unauthorized subdomains. Utilize DNSSEC to enhance DNS security. * **Intrusion Detection/Prevention System (IDS/IPS):** Deploy an IDS/IPS to monitor network traffic for malicious activity and block suspicious connections. * **Regular Security Audits:** Conduct regular security audits and penetration testing to proactively identify and mitigate vulnerabilities. * **Vulnerability Scanning:** Implement automated vulnerability scanning tools to continuously monitor for new threats. Consider vulnerability scans covering both web applications and the underlying operating systems and services. * **Log Analysis:** Implement and regularly review security logs to detect unusual activities. **Important Note:** This analysis is based on limited reconnaissance data. A more comprehensive security assessment, including a full penetration test and vulnerability scan, is strongly recommended.

■ Consolidated Tool Results

■ Subdomain_enum Results


```
{'subdomains': ['[i] Enumerating 10 candidates for pesce.ac.in (DNS available: True, HTTP
```


■ AI Analysis of Subdomain_enum

Analysis of Subdomain Enumeration Output for pesce.ac.in The output reveals a partial success in enumerating subdomains for `pesce.ac.in`. Let's break down the findings: **1. Useful Subdomains:** * **`www.pesce.ac.in`:** The main website. This is crucial for understanding the organization's online presence. * **`ftp.pesce.ac.in`:** An FTP server is exposed. This could potentially allow unauthorized access to files if not properly secured. * **`mail.pesce.ac.in`:** A mail server. This is a critical asset that needs strong security measures. **2. Risky Infrastructure Exposures:** * **FTP Server (`ftp.pesce.ac.in`):** FTP is an insecure protocol. The mere existence of an exposed FTP server is a significant risk, especially if it's not using strong authentication and encryption (like FTPS or SFTP). An attacker could potentially upload malicious files or download sensitive data. * **Multiple Subdomains Resolving to the Same IP (`www.pesce.ac.in` and `mail.pesce.ac.in` sharing `68.178.161.81`):** While not inherently risky, this indicates potential vulnerabilities if misconfiguration leads to compromised access affecting both services. * **Lack of Subdomain Security:** The failure to resolve most attempted subdomains (e.g., `staging`, `beta`, `dev`, `test`, `admin`, `api`) suggests a lack of robust subdomain management. This could indicate a lack of proper security controls and monitoring. While this might seem "safe" in that no assets are directly found, it suggests gaps in the organization's security posture. **3. Risk Level:** I'd rate the overall risk level as **Medium to High**. The presence of an insecure FTP server and the potential for misconfiguration across shared IP addresses are major concerns. The lack of other subdomains also highlights potential blind spots for security monitoring. **4. Suggested Next Actions:** * **Prioritize FTP Server Security:** Conduct a thorough security assessment of the `ftp.pesce.ac.in` server. This includes verifying authentication methods, checking for weak passwords, and exploring the possibility of migrating to a more secure protocol (FTPS or SFTP). Consider disabling the FTP server entirely if not absolutely necessary. * **Investigate Shared IP Address:** Determine why `www.pesce.ac.in` and `mail.pesce.ac.in` share the same IP. This could be a deliberate design choice, but it warrants further investigation to rule out potential vulnerabilities due to misconfiguration or a compromised server affecting both services. * **Expand Subdomain Enumeration:** Use more sophisticated tools and techniques to attempt a more comprehensive subdomain enumeration. This could uncover additional subdomains that might expose further vulnerabilities. Consider using techniques like brute forcing, recursive DNS queries, and crawling the main website for clues. * **Vulnerability Scanning:** Perform vulnerability scans on the identified servers (`68.178.161.81` and `103.117.212.198`) to identify any known vulnerabilities. Tools like Nessus, OpenVAS, or QualysGuard can assist with this. * **Web Application Security Testing:** Conduct a thorough web application security assessment of `www.pesce.ac.in` to identify potential vulnerabilities like Cross-Site Scripting (XSS), SQL Injection, and others. * **Reconnaissance on the Identified IPs:** Perform additional reconnaissance on the identified IP addresses using tools like Nmap to identify open ports and services. This will help pinpoint potential attack vectors. By addressing these issues, the organization can significantly reduce its attack surface and improve its overall security posture. The findings suggest a need for more proactive security management and incident response planning.

■ Sql_injection Results


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

■ AI Analysis of Sql_injection

The provided SQLmap output shows that **no SQL injection vulnerabilities were found**. **Analysis:** The key line is: `[11:57:21] [WARNING] no usable links found (with GET parameters)` This indicates that sqlmap, after crawling the target URL (`http://pesce.ac.in`), did not identify any GET parameters that could be exploited for SQL injection. The crawler examined the site for links with GET parameters (e.g., `http://example.com/page?id=123`) as these are common locations for SQL injection vulnerabilities. The absence of usable links implies that there were no apparent injection points tested. **Vulnerable Parameters/Endpoints:** None identified. The report explicitly states that no usable links (with GET parameters) were found. This does *not* rule out the possibility of vulnerabilities using other attack vectors (POST parameters, blind SQL injection, etc.), but the scan using GET parameters did not reveal any. **CVEs or Known Exploits:** Not applicable, as no vulnerabilities were found in this specific test. **Risk Level:** Based solely on this SQLmap output, the risk level concerning SQL injection via GET parameters is **Low**. However, this is a limited assessment. A comprehensive security assessment would require testing for other types of vulnerabilities, including: * **SQL injection via POST parameters:** SQLmap should be run again testing POST requests. * **Blind SQL injection:** This technique is harder to detect automatically and needs specialized testing. * **Other vulnerabilities:** A complete penetration test would examine XSS, CSRF, authentication flaws, and other potential weaknesses. **Conclusion:** This SQLmap scan provides only partial information. While it didn't find SQL injection vulnerabilities in GET parameters, it's insufficient to conclude the complete absence of vulnerabilities on the target website. Further testing is strongly recommended.

■ Theharvester Results


```
{'emails': [], 'hosts': ['alumni.pesce.ac.in', 'mail.pesce.ac.in'], 'domains': []}
```

■ AI Analysis of Theharvester

This scan result reveals a limited but potentially concerning picture. Let's break down the risks and potential mitigations: **Analysis:** The scan only found two hostnames: `alumni.pesce.ac.in` and `mail.pesce.ac.in`. The absence of emails and domains in the results suggests a limited scope of the scan, possibly focusing only on hostname discovery. This doesn't mean there are *no* emails or domains associated with `pesce.ac.in`; it just means they weren't discovered by this particular scan. **Risks:** * **Limited Information:** The biggest risk is the incompleteness of the scan. Without more information (e.g., open ports, services running on those hosts, vulnerabilities), it's impossible to assess the true security posture. This limited data only provides a starting point for further investigation. * **Potential for email compromise:** `mail.pesce.ac.in` is likely a mail server. If this server is vulnerable (e.g., outdated software, weak passwords, lack of proper security patching), it could be compromised leading to data breaches (email leaks, phishing attacks). * **Alumni Network Vulnerability:** `alumni.pesce.ac.in` could expose sensitive information about alumni if not properly secured. This could lead to identity theft or targeted attacks against alumni. * **Further Reconnaissance:** The limited data suggests the need for a more comprehensive security scan. An attacker could leverage this information to perform further reconnaissance, identify additional targets, and launch more sophisticated attacks. **Suggested Exploits (Hypothetical, based on limited data):** Without more information, these are purely *hypothetical* exploits based on the potential vulnerabilities of mail and alumni servers. They should not be attempted without explicit permission from the organization responsible for the network. * **Mail Server Exploitation:** An attacker could attempt to exploit known vulnerabilities in the mail server software (if outdated or misconfigured) to gain unauthorized access. This could involve buffer overflows, SQL injections, or exploiting known zero-day exploits. * **Phishing:** Compromising the mail server could enable phishing campaigns targeting alumni or staff. **Mitigations:** * **Comprehensive Security Scan:** Conduct a thorough vulnerability scan using a reputable tool that examines open ports, services, and common vulnerabilities and exposures (CVEs). This should include network scans, web application scans, and potentially penetration testing. * **Mail Server Hardening:** Ensure the `mail.pesce.ac.in` server is up-to-date with all security patches, employs strong password policies, and uses appropriate security measures like SPF, DKIM, and DMARC to prevent email spoofing and phishing. Regular backups are essential. * **Alumni Portal Security:** Review the security of the `alumni.pesce.ac.in` portal. Implement strong authentication, access controls, and regularly audit user access privileges. Ensure data is encrypted both in transit and at rest. * **Intrusion Detection/Prevention System (IDS/IPS):** Deploy an IDS/IPS to monitor network traffic and detect suspicious activity, potentially alerting to intrusion attempts. * **Regular Security Audits:** Regular security audits and penetration testing are crucial to identify vulnerabilities and weaknesses before attackers can exploit them. * **Employee Security Training:** Educate employees and alumni about phishing and social engineering tactics to prevent them from becoming victims of attacks. In summary, this scan provides a very limited view of the potential security risks. A much more thorough investigation is required to accurately assess the situation and implement appropriate security measures. The identified hosts are critical infrastructure and require immediate attention.

■ Nmap Results


```
{'68.178.161.81': {'state': 'up', 'os': [{'name': 'Linux 3.10 - 4.11', 'accuracy': '86',
```

■ AI Analysis of Nmap

Nmap Scan Result Analysis: 68.178.161.81 This Nmap scan reveals a Linux server (likely 3.10-4.14 kernel) with several services running, some of which present potential vulnerabilities. **Identified Services and Versions:** * **FTP (Port 21):** ProFTPD (version unspecified). This is a significant risk if not properly configured. * **SSH (Port 22):** OpenSSH 8.0. Requires further investigation for misconfigurations. * **SMTP (Port 25):** Filtered. * **DNS (Port 53):** PowerDNS Authoritative Server 4.9.5. Requires further investigation for misconfigurations. * **HTTP (Port 80):** Apache httpd (version unspecified). Requires further investigation for outdated versions and misconfigurations. * **POP3 (Port 110):** Dovecot pop3d (version unspecified). * **RPCbind (Port 111):** Version 2-4. Older versions have known vulnerabilities. * **IMAP (Port 143):** Dovecot imapd (version unspecified). * **HTTPS (Port 443):** Apache httpd (version unspecified). * **SMTP (Port 465 & 587):** Exim smtpd 4.98.2. Requires checking for known vulnerabilities in this version. * **IMAPS (Port 993):** Version unspecified. * **POP3S (Port 995):** Version unspecified. * **MySQL (Port 3306):** MySQL (version unspecified). A major risk if default credentials or weak passwords are used. **Possible Vulnerabilities and Risk Levels:** * **FTP (Port 21):** ProFTPD has a history of vulnerabilities. Without a version, it's impossible to pinpoint specific CVEs. **Risk Level: Medium to High** (depending on the version and configuration). Exploitation would involve attempting common exploits for ProFTPD or brute-forcing credentials. * **SSH (Port 22):** OpenSSH 8.0 might have vulnerabilities depending on the specific sub-version and configuration. Regular security updates are crucial. **Risk Level: Medium**. Vulnerability checks need further investigation using tools like Nessus or OpenVAS, and potentially exploiting any found vulnerabilities. * **PowerDNS (Port 53):** Requires checking for known vulnerabilities in version 4.9.5 via vulnerability databases (e.g., NVD). **Risk Level: Low to Medium**. Misconfigurations or unpatched versions pose the biggest threat. * **Apache HTTP (Port 80 & 443):** Outdated versions of Apache are susceptible to numerous attacks (e.g., RCE, XSS). Requires version identification and vulnerability scanning. **Risk Level: Medium to High**. Exploitation tools will depend on the found vulnerabilities. * **Dovecot (Port 110, 143, 993, 995):** Version identification is critical to assess vulnerabilities. **Risk Level: Medium**. Exploitation may involve credential stuffing or exploiting known vulnerabilities if outdated. * **RPCbind (Port 111):** Older versions of rpcbind have known vulnerabilities allowing for remote code execution. **Risk Level: Medium to High**. Requires version verification and specific vulnerability checks. * **Exim SMTP (Port 465 & 587):** Version 4.98.2 needs to be checked against CVE databases for any known exploits. **Risk Level: Medium**. * **MySQL (Port 3306):** Default credentials or weak passwords are the biggest risk. **Risk Level: High**. Tools like sqlmap can be used to attempt exploitation if vulnerable. **Suggested Exploitation Tools and Methods:** The specific tools and methods depend heavily on the identified vulnerabilities. The following are general approaches: * **Nmap scripts:** Use Nmap's scripting engine to perform more in-depth service version detection and vulnerability checks. * **Nessus/OpenVAS:** Conduct a full vulnerability scan to identify specific CVEs applicable to the discovered versions. * **Metasploit:** Once vulnerabilities are identified, Metasploit can be used to attempt exploitation. * **Burp Suite:** For web application vulnerabilities (Apache), Burp Suite is invaluable for identifying and exploiting XSS, SQL injection, and other web-related flaws. * **sqlmap:** Specifically for MySQL, sqlmap can be used to check for SQL injection vulnerabilities. * **Hydra/John the Ripper:** For brute-forcing passwords against FTP, SSH, POP3, IMAP, and MySQL. **Next Steps:** 1. **Version Identification:** Determine the precise versions of all the services. 2. **Vulnerability Scanning:** Use Nessus or OpenVAS to identify known vulnerabilities based on the versions. 3. **Manual Testing:** Based on the vulnerabilities identified, conduct manual testing to confirm and exploit them (ethically and with permission). 4. **Credential Brute-forcing:** Only attempt this with proper authorization and after exhausting other options. This analysis provides a starting point. A thorough penetration test requires further investigation and confirmation of the identified vulnerabilities. Remember to always obtain explicit permission before performing any penetration testing activities.

■ Sublist3r Results


```
{'subdomains': ['www.pesce.ac.in', '', 'alumni.pesce.ac.in', '', 'www.alumni.pesce.ac.in'
```


■ AI Analysis of Sublist3r

Analyzing the Sublist3r output: The output shows a small number of subdomains for `pesce.ac.in`. Let's break down each one: * **`www.pesce.ac.in`**: This is the main website. It's expected and requires basic security assessment. * **`alumni.pesce.ac.in`**: This subdomain likely hosts content specifically for alumni. It could have different security configurations than the main site, making it potentially more vulnerable. * **`www.alumni.pesce.ac.in`**: This is a `www` subdomain for the alumni site. Redundant, but needs investigation just like `alumni.pesce.ac.in`. * **`mail.pesce.ac.in`**: This subdomain likely hosts email services. Email servers are frequent targets for attacks, so this is a high-priority target for further investigation. A compromised email server could lead to significant data breaches. * **Empty Strings (`''`):** These are likely artifacts of Sublist3r's output and can be ignored. **Interesting/Vulnerable-Looking Subdomains:** * `mail.pesce.ac.in` (High risk) * `alumni.pesce.ac.in` (Medium risk) **Staging/Test Environments:** None are explicitly identified in this limited output. However, further investigation may reveal test or staging environments using techniques like directory brute-forcing or searching for common naming patterns (e.g., `test.pesce.ac.in`, `stage.pesce.ac.in`, `dev.pesce.ac.in`). **Risk Level:** Overall, the risk is moderate. The presence of a `mail` subdomain raises the risk level significantly due to the potential impact of a compromise. **Suggested Next Recon Steps:** 1. **Port Scanning:** Perform a port scan on all identified subdomains to identify open ports and services running. This will help determine the technologies used and potential vulnerabilities. Nmap is a good tool for this. 2. **Service Versioning:** Identify versions of services running on open ports. Outdated software is often vulnerable. 3. **Vulnerability Scanning:** Conduct automated vulnerability scans on each subdomain using tools like OpenVAS or Nessus. Focus on known vulnerabilities for email servers (if `mail.pesce.ac.in` is vulnerable) and common web application vulnerabilities. 4. **Takeover Checks:** Use tools like `takeover` (GitHub) to check for potential subdomain takeovers. This involves identifying if the subdomain points to a service that is easily compromised. 5. **Directory Brute-forcing:** Attempt to brute-force directories on `alumni.pesce.ac.in` and `www.alumni.pesce.ac.in` to uncover hidden files or directories, potentially revealing sensitive information or vulnerabilities. DirBuster or Gobuster are useful for this. 6. **Subdomain Brute-forcing:** Since the initial enumeration was limited, consider expanding it with a more thorough brute-force subdomain enumeration using tools like Subfinder, amass, or a combination. 7. **Passive Reconnaissance:** Utilize tools like Shodan to gather more information about the services running on these subdomains. Shodan can reveal exposed information and services. By following these steps, you can build a more complete picture of the target's security posture and identify potential vulnerabilities. Remember to always obtain proper authorization before conducting any security testing.