

Network Security and Administrative Software Implementation

FINAL REPORT

BY

Maninder Kaur

Funmilayo Ogabi

Sirjan Pun Magar

Vijay Cyril Yeruva

Reshmaben Shaileshbhai Patel

Henny Jayeshbhai Patel

TABLE CONTENT

Executive Summary.....	1
Introduction	2
Scope	2
Server Infrastructure Policy.....	3
Problem Statement.....	13
Problem Solution Debrief.....	16
Software Details.....	17
Implementation software	19
Configuration In Sshd_Conf	22
Implementation Plan	25

Executive summary

The Auxilium Group is a web-based technology and online data management platform focused on offering accessible solutions to organizations. Auxilium Group focuses on enhancing network security by addressing vulnerabilities that can compromise digital assets. Their primary focus is to identify and mitigate these vulnerabilities to safeguard the organization's sensitive information and ensure uninterrupted operations. They conduct a thorough analysis to identify potential weaknesses and entry points in network infrastructure and employ industry best practices, advanced tools, and innovative technologies to fortify the network's defenses.

The primary goal of our study is to detect and strengthen network weaknesses. During our network inspection, our team discovers brute force attack vulnerabilities. We use the FAIL2BAN program to address this issue, which efficiently mitigates these attacks. This program integrates smoothly into the network, boosting security. Our initiative focuses on detecting vulnerabilities and applying smart solutions, such as FAIL2BAN, to strengthen network security against brute force assaults.

Introduction:

Auxilium Group is committed to guaranteeing the security and reliability of its network infrastructure. Our major focus is thoroughly investigating the network to identify potential vulnerabilities. During our investigation, we discovered a serious vulnerability in the form of a brute-force attack. To solve this, we carefully chose to integrate the FAIL2BAN software, a powerful solution known for combating such attacks. This program was effortlessly incorporated into the server, strengthening the network's defenses. In summary, our mission is to secure the network by discovering vulnerabilities and then strategically implementing solutions like FAIL2BAN to improve its overall security posture.

SCOPE

The purpose of this project is to formulate a comprehensive policy for the server infrastructure and develop a security strategy to mitigate common threats. We will implement the planned hardening strategy across our network, addressing vulnerabilities with appropriate security measures. Our goal is to improve network security and reduce risks. Our team will ensure that every implementation follows the plan, delivering the desired outcomes. Project Team: The following team members will be involved in the project:

Project Manager – Reshma Patel

Researcher - Sirjan Pun Magar & Vijay Cyril Reddy

Developer - Vijay Cyril Reddy

Presenter - Maninder Kaur & Funmilayo Ogabi YU

Technical Writer – Maninder Kaur

Technical Lead - Henny Patel

Course Coordinator – Sirjan Pun Magar

Server Infrastructure Policy

Document Control

Organization	Auxilium Group
Title	Server Infrastructure Policy Document
Filename	Server_Infrastructure_Policy.txt
Owner	Alixander Greganti
Subject	Server Infrastructure Policy

Purpose: The Server Infrastructure Policy establishes guidelines and best practices for the secure design, implementation, and management of the organization's server environment. This policy aims to ensure the availability, integrity, and confidentiality of data and services hosted on the servers.

This policy applies to all servers owned or managed by the organization, including physical and virtual servers deployed both on-premises and in the cloud.

Roles and Responsibilities

Server Administration Team: The Server Administration Team is responsible for the proper setup, configuration, and maintenance of servers. They ensure adherence to security standards and monitor server performance and availability.

Information Security Team: The Information Security Team defines security measures, conducts risk assessments, and ensures compliance with industry standards and regulations. They provide guidance on security controls and incident response.

IT Support Team: The IT Support Team addresses user issues related to server access, performance, and troubleshooting. They collaborate with other teams to maintain a secure and well-functioning server environment.

Hardware and Software Standards

Servers must adhere to established hardware and software standards to maintain consistency, security, and compatibility. These standards cover server hardware specifications, operating systems, and software applications.

Access Controls

Role-Based Access: Access to servers is based on job roles and responsibilities. Users are granted permissions necessary for their tasks, following the principle of least privilege.

Access Request and Approval: Access to servers requires proper authorization from designated personnel. Access requests are reviewed and approved based on job function and need.

Data Management

Data Classification: Data is classified based on sensitivity and confidentiality. This classification determines access controls, encryption requirements, and retention policies.

Encryption: Sensitive data in transit and at rest is encrypted using approved cryptographic methods to protect against unauthorized access and data breaches.

Security Controls

Antivirus and Malware Protection: All servers must have up-to-date antivirus and anti-malware software to detect and mitigate malicious software.

Intrusion Detection and Prevention: Intrusion Detection and Prevention Systems (IDPS) are deployed to monitor and respond to unauthorized access attempts and suspicious activities.

Firewall Configuration: Firewalls are configured to restrict unauthorized network traffic and prevent unauthorized access to servers.

Monitoring and Logging

Real-time Monitoring: Servers are continuously monitored in real-time for unusual activities and potential security breaches.

Log Collection and Analysis: Logs of server activities and events are collected, stored, and regularly reviewed for security monitoring and forensic analysis.

Incident Response

Incident Reporting: All personnel must promptly report any security incidents or suspicious activities to the Information Security Team.

Incident Escalation: Incidents are escalated based on their severity and potential impact. The Incident Response Team coordinates and implements incident response procedures.

Patch and Vulnerability Management

Patching: Servers and software applications must be regularly patched with security updates to address known vulnerabilities.

Vulnerability Assessment: Regular vulnerability assessments are conducted to identify and mitigate potential security weaknesses.

Change management

Changes to server configurations and settings must be authorized and documented to maintain accountability.

Compliance and Regulation

Changes are tested in a controlled environment before implementation to prevent disruptions and security risks.

Training and Awareness

Security Training: Personnel involved in server management must undergo regular security training to stay informed about cybersecurity threats and best practices.

User Awareness: User awareness campaigns are conducted to educate personnel about security risks and the importance of compliance.

Threat Assessment Report:

Detailed Analysis of Attack Vectors

This detailed analysis delves into the common attack vectors that malicious actors may exploit to compromise the server infrastructure, as identified in the Threat Assessment Report.

Phishing Attacks:

Description: Phishing attacks are deceptive attempts to trick users into divulging sensitive information, such as usernames, passwords, credit card details, or personal data. These attacks often involve emails, websites, or other communication channels that appear legitimate but are controlled by malicious actors.

Exploitation: Malicious actors craft convincing emails that mimic legitimate organizations, urging recipients to click on malicious links, download attachments, or provide confidential information.

Mitigation:

- **Employee Training:** Regularly educate users about recognizing phishing attempts, including checking sender email addresses, verifying links, and avoiding suspicious requests.
- **Email Filters:** Implement advanced email filters and spam detection mechanisms to identify and block phishing emails.

- **Multi-Factor Authentication (MFA):** Require MFA for sensitive accounts to add an extra layer of security against stolen credentials.

Malware Infections:

Description: Malware infections involve the introduction of malicious software onto a system, often through infected files, downloads, or compromised applications. Malware can include viruses, worms, Trojans, ransomware, and spyware.

Exploitation: Malicious actors may use various methods to distribute malware, such as disguising it as legitimate software, exploiting software vulnerabilities, or enticing users to download infected files.

Mitigation:

- **Regular Updates:** Keep operating systems and applications up to date with the latest security patches to prevent the exploitation of known vulnerabilities.
- **Endpoint Protection:** Deploy robust antivirus and anti-malware software to detect and remove malicious code.
- **User Education:** Train users to avoid downloading files from untrusted sources and to exercise caution when clicking on links.

Brute Force Attacks:

Description: Brute force attacks involve repeated and automated attempts to guess passwords or gain unauthorized access by exploiting weak authentication mechanisms.

Exploitation: Malicious actors use automated tools to systematically guess passwords or credentials until they gain access to a system. Weak passwords and inadequate authentication mechanisms make this attack vector viable.

Mitigation:

- **Strong Password Policies:** Enforce complex password requirements, including length, character variety, and periodic changes.
- **Account Lockouts:** Implement account lockout mechanisms after a certain number of failed login attempts.
- **Multi-Factor Authentication (MFA):** Require MFA to add an extra layer of protection even if passwords are compromised.

DDoS Attacks:

Description: Distributed Denial of Service (DDoS) attacks involve overwhelming a server or network with a flood of traffic, rendering it unable to respond to legitimate requests.

Exploitation: Malicious actors deploy botnets or other resources to generate a massive volume of traffic aimed at the target server or network.

Mitigation:

- **Traffic Filtering:** Implement traffic filtering mechanisms to identify and block malicious traffic patterns.
- **DDoS Protection Services:** Utilize specialized DDoS protection services and solutions to absorb and mitigate attack traffic.
- **Scalability:** Design server infrastructure to handle traffic spikes and distribute loads effectively.

Injection Attacks:

Description: Injection attacks involve exploiting vulnerabilities in software by injecting malicious code, such as SQL injection or cross-site scripting, into the system.

Exploitation: Malicious actors insert malicious input, often in the form of scripts or code snippets, into user inputs or data fields. This can lead to the execution of unintended actions by the system.

Software Vulnerabilities:

Outdated Software:

- **Description:** Running outdated operating systems and applications may expose the server to known vulnerabilities that have been patched in newer versions.
- **Impact:** Attackers could exploit unpatched vulnerabilities to gain unauthorized access, execute malicious code, or disrupt services.

Unpatched Security Vulnerabilities:

- **Description:** Failing to apply timely security patches and updates leaves the server susceptible to known vulnerabilities that threat actors can exploit.
- **Impact:** Attackers may exploit unpatched vulnerabilities to breach the server, compromise data, or execute remote code.

Insecure Third-Party Software:

- Description: Incorporating insecure third-party software or libraries may introduce vulnerabilities that attackers can target.
- Impact: Exploitation of third-party software vulnerabilities can lead to data breaches, unauthorized access, and potential system compromise.

Weak Authentication Mechanisms:

- Description: Inadequate authentication processes, weak password policies, and lack of multi-factor authentication (MFA) can be exploited by attackers.
- Impact: Weak authentication can result in unauthorized access, account compromise, and data leakage.

Hardware Vulnerabilities:

Firmware Vulnerabilities

- Outdated or unpatched firmware in server hardware components could contain vulnerabilities that attackers exploit.
- Attackers may compromise firmware to gain unauthorized access, bypass security controls, or execute malicious code.

Inadequate Hardware Security Configuration:

- Improper configuration of hardware security features, such as Trusted Platform Modules (TPMs), can expose the server to attacks.
- Inadequate hardware security configurations may lead to unauthorized access or data theft.

Physical Security Lapses:

- Poor physical security measures, such as unsecured server rooms or lack of access controls, can expose hardware to unauthorized manipulation.
- Attackers with physical access may tamper with hardware components, compromising server integrity and confidentiality.

Supply Chain Vulnerabilities:

- Components sourced from compromised supply chains may introduce vulnerabilities in server hardware.

- Supply chain compromise could lead to compromised hardware, unauthorized access, or backdoor entry points.

Potential Entry Points

Threat actors often target various entry points to compromise a server's security and gain unauthorized access. These entry points are based on server infrastructure and setup. Here are some potential malicious entry points that attackers may exploit:

- **Unpatched Software:** Exploiting known vulnerabilities in operating systems, applications, or software components that have not been updated with the latest security patches.
- **Weak Authentication:** Gaining unauthorized access by exploiting weak or default passwords, inadequate password policies, or a lack of multi-factor authentication (MFA).
- **Insecure Network Connections:** Attacking the server through unsecured network connections, such as open ports, exposed services, or poorly configured firewalls.
- **Phishing Attacks:** Trickling users into disclosing credentials or sensitive information through deceptive emails, websites, or social engineering.
- **Malware Infections:** Introducing malicious software through infected files, downloads, or compromised applications that can provide attackers with unauthorized access.
- **Brute Force Attacks:** Repeatedly attempting to guess passwords or access credentials by systematically trying different combinations.
- **Injection Attacks:** Exploiting vulnerabilities in software by injecting malicious code, such as SQL injection or cross-site scripting, to gain unauthorized access.
- **Third-Party Integrations:** Exploiting vulnerabilities in third-party applications or integrations connected to the server.
- **Insider Threats:** Exploiting malicious actions taken by insiders with legitimate access, including employees, contractors, or partners.
- **Misconfigured Security Settings:** Gaining access due to misconfigured access controls, permissions, or security settings.
- **Physical Access:** Attacking the server by gaining physical access to the data centre or server room where the hardware is located.
- **Social Engineering:** Manipulating individuals into providing access to sensitive information through psychological manipulation.
- **File Upload Vulnerabilities:** Uploading malicious files that can be executed on the server if not properly validated.
- **Exposed API Endpoints:** Attacking the server by exploiting vulnerabilities in exposed APIs that may lack proper authentication and authorization.

- **Physical Hardware Attacks:** Tampering with or compromising the physical server hardware to gain unauthorized access.

Impact of Successful Attacks

- Successful phishing attacks can lead to unauthorized access to sensitive systems, data breaches, identity theft, and financial loss. Phishing attacks are also frequently used as an initial entry point for broader attacks.
- Malware infections can result in data theft, unauthorized access, system disruptions, and financial losses. Ransomware attacks can encrypt critical data and demand payment for its release.
- Successful brute force attacks can lead to unauthorized access, data breaches, and compromised accounts.
- DDoS attacks can cause service disruptions, downtime, decreased user experience, and financial losses.
- Injection attacks can lead to data leakage, unauthorized access, data manipulation, and the compromise of sensitive information.

Security Controls and Procedures:

SSH DDoS Attack Prevention

Rate Limiting and Connection Throttling

- Implement rate limiting to restrict the number of SSH connection requests from a single IP address within a specified time frame.
- Configure connection throttling mechanisms to prevent rapid, successive connection attempts.

IP Whitelisting and Blacklisting

- Maintain an IP whitelist of trusted sources allowed to access the SSH service.
- IP blacklisting for IP addresses showing suspicious behavior or excessive failed login attempts.

Fail2Ban Implementation

- Install and configure Fail2Ban to monitor SSH logs for patterns of failed login attempts.
- Automate the blocking of IP addresses with a history of multiple failed login attempts.

Network Traffic Shaping and Filtering:

- Implement network-level traffic shaping and filtering to regulate the volume of incoming SSH traffic from specific IP ranges. Employ traffic filtering techniques to identify and block traffic from known malicious sources.

Security is a continuous process that does not stop with one solution; it requires time to time protection and an update with an upgrade. Here we list out some other implementations to minimize the risk and challenges that can be possible. The other recommendations are:

SSH Key Authentication:

- Encourage SSH key authentication over password-based authentication. Promote the use of strong, passphrase-protected SSH keys to mitigate the risk of password-related attacks, including SSH DDoS attacks.

Real-time Traffic Analysis

- Deploy real-time traffic analysis tools to monitor incoming SSH connections.
- Identify unusual traffic patterns that may indicate a DDoS attack.

Two-Factor Authentication (2FA)

- Enforce the use of two-factor authentication for SSH logins.
- Combine SSH key authentication with an additional authentication factor.

Scalability and Load Balancing:

- Design the server infrastructure for scalability and load balancing. Distribute incoming SSH traffic across multiple servers to minimize the impact of DDoS attacks on individual systems.

Monitoring And Incident Response Plan

Auxilium Group is committed to offering comprehensive network security solutions, and our Monitoring and Incident Response Plan, when combined with Fail2ban, provides a strong defense against possible attacks. We maintain the security and stability of your network by combining constant monitoring with quick reaction measures.

Monitoring

- **Logs:** Keep an eye on the Fail2ban log (/var/log/fail2ban.log) for information regarding discovered patterns and blacklisted IP addresses. This log keeps track of detected incidents, blacklisted IPs, and un-banned IPs.
- **Intrusion Detection Systems (IDS):** Integrating Fail2ban with an IDS such as Snort or Suricata improves monitoring capabilities. An IDS can perform more advanced network traffic analysis and identify more complicated assaults.
- Fail2ban, an advanced intrusion prevention system, continually monitors logs for suspicious behaviour, such as repeated unsuccessful login attempts.

- Real-time alerts inform our specialists of possible hazards in a timely manner, enabling fast investigation and response.

Incident Response Plan

- Determine the types of occurrences you wish to handle by defining them. Failed login attempts, SQL injection attempts, and other scenarios are examples.
- Response categories: Divide issues into categories based on severity. This aids in the prioritization of replies and the efficient allocation of resources.
- Actions:
 - Tier 1: Log and monitor for any escalation of small occurrences that do not constitute an immediate hazard.
 - Tier 2: For moderate events, temporarily ban the IP address using Fail2ban to prevent additional unauthorized access attempts.
 - Tier 3: Permanently prohibit the IP address and undertake a more complete investigation into the source and impact of the assault.
- Notification: Decide who should be notified for each tier. Make certain that there is a clear communication route for incident warnings. Email, messaging applications, or specialist crisis response systems might all be used to send notifications.
- Documentation: Keep accurate records of all incidents. Timestamps, pertinent log entries, actions taken, and any follow-up investigations or resolutions should all be included in this record.

Review and Continuous Improvement

- Set up frequent evaluations of event logs, and replies. Look for patterns or trends that may suggest developing attack methodologies or possible flaws in security posture.
- Adjustment: Based on the findings of evaluations, make changes to Fail2ban rules, incident response methods, or even system settings to better handle new threats.
- Regularly train IT and security personnel on the incident response strategy and processes. Conduct simulated drills to ensure that everyone understands their duties and responsibilities during a security issue.

It is critical to emphasize that the success of our Monitoring and Incident Response Plan is dependent on our dedication to remaining ahead of the curve. We must remain attentive by remaining up to date on the newest security trends, continually improving our procedures, and cultivating a security-conscious culture throughout the organization. This dynamic strategy guarantees that we not only respond successfully to emerging attacks but also develop effectively

to the rapidly changing cybersecurity landscape. We strengthen our commitment to protecting our network and digital assets by cultivating this mentality.

PROBLEM STATEMENT:

During the process of evaluating Auxilium Group's network security, our dedicated teams encountered a notable concern: a multitude of open ports were found accessible. To resolve this issue, we sought the necessary authority to perform extensive testing, with a particular focus on the Montreal server.

This proactive approach enabled us to go deep into the security architecture, painstakingly evaluating possible weaknesses. During this investigation, our researchers discovered a significant vulnerability: a brute force attack was observed on the Montreal server. This revelation emphasized the relevance of our review by highlighting the real-world hazards that the organization's network faced.

The combination of these studies emphasizes the significance of thorough network security testing. Our method not only revealed flaws, but it also acted as a catalyst for strengthening the organization's defenses. By dealing with unsecured ports and identifying the brute force assault, we helped Auxilium Group build a more robust and secure network environment.

Nmap search on the Montreal location IP address 69.90.202.100

```
File Actions Edit View Help
(henny@kali):[~]
$ nmap -A 69.90.202.100
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-27 19:06 EDT
Nmap scan report for auxilium.world (69.90.202.100)
Host is up (0.043s latency).
Not shown: 891 filtered tcp ports (no-response), 102 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   256 00:aa:00:99:9b:91:ef:13:17:b4:80:4a:39:0f:2c:23 (ECDSA)
|   256 0f:17:9f:a9:0f:dc:79:f4:be:97:3f:c9:3a:da:ac:fa (ED25519)
80/tcp    open  http     nginx
|_ http-title: 404 Not Found
443/tcp   open  ssl/http nginx
|_ http-title: 404 Not Found
|_ ssl-cert: Subject: commonName=api.datalynk.ca
|_ Subject Alternative Name: DNS:api.datalynk.ca
|_ Not valid before: 2023-05-31T00:00:00
|_ Not valid after: 2023-08-29T23:59:59
|_ ssl-date: TLS randomness does not represent time
8080/tcp   open  ssl/http nginx
|_ ssl-date: TLS randomness does not represent time
|_ ssl-cert: Subject: commonName=api.datalynk.ca
|_ Subject Alternative Name: DNS:api.datalynk.ca
|_ Not valid before: 2023-05-31T00:00:00
|_ Not valid after: 2023-08-29T23:59:59
|_ http-title: 404 Not Found
8081/tcp   open  ssl/http nginx
|_ ssl-date: TLS randomness does not represent time
|_ http-title: 502 Bad Gateway
|_ ssl-cert: Subject: commonName=api.datalynk.ca
|_ Subject Alternative Name: DNS:api.datalynk.ca
|_ Not valid before: 2023-05-31T00:00:00
|_ Not valid after: 2023-08-29T23:59:59
8082/tcp   open  ssl/http nginx
|_ ssl-date: TLS randomness does not represent time
|_ http-title: 502 Bad Gateway
|_ ssl-cert: Subject: commonName=api.datalynk.ca
|_ Subject Alternative Name: DNS:api.datalynk.ca
|_ Not valid before: 2023-05-31T00:00:00
|_ Not valid after: 2023-08-29T23:59:59
9000/tcp   open  ssl/http nginx
|_ ssl-cert: Subject: commonName=storage.auxilium.world
|_ Subject Alternative Name: DNS:storage.auxilium.world
|_ Not valid before: 2023-05-31T00:00:00
```

```

File Actions Edit View Help
443/tcp open  ssl/http nginx
|_http-title: 404 Not Found
|_ssl-cert: Subject: commonName=api.datalynk.ca
|_Subject Alternative Name: DNS:api.datalynk.ca
|_Not valid before: 2023-05-31T00:00:00
|_Not valid after: 2023-08-29T23:59:59
|_ssl-date: TLS randomness does not represent time
8080/tcp open  ssl/http nginx
|_ssl-date: TLS randomness does not represent time
|_ssl-cert: Subject: commonName=api.datalynk.ca
|_Subject Alternative Name: DNS:api.datalynk.ca
|_Not valid before: 2023-05-31T00:00:00
|_Not valid after: 2023-08-29T23:59:59
|_http-title: 404 Not Found
8081/tcp open  ssl/http nginx
|_ssl-date: TLS randomness does not represent time
|_http-title: 502 Bad Gateway
|_ssl-cert: Subject: commonName=api.datalynk.ca
|_Subject Alternative Name: DNS:api.datalynk.ca
|_Not valid before: 2023-05-31T00:00:00
|_Not valid after: 2023-08-29T23:59:59
8082/tcp open  ssl/http nginx
|_ssl-date: TLS randomness does not represent time
|_http-title: 502 Bad Gateway
|_ssl-cert: Subject: commonName=api.datalynk.ca
|_Subject Alternative Name: DNS:api.datalynk.ca
|_Not valid before: 2023-05-31T00:00:00
|_Not valid after: 2023-08-29T23:59:59
9000/tcp open  ssl/http nginx
|_ssl-cert: Subject: commonName=storage.auxilium.world
|_Subject Alternative Name: DNS:storage.auxilium.world
|_Not valid before: 2023-05-31T00:00:00
|_Not valid after: 2023-08-29T23:59:59
|_ssl-date: TLS randomness does not represent time
|_http-title: Did not follow redirect to http://auxilium.world:9001
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 39.24 seconds

```

```

(henny@kali)~$

```

```

File Actions Edit View Help
nsirjan@balancer2:~$ sudo nmap
nmapscpe ssh:notty 134.17.94.27 Tue Jul 25 19:26 - 19:26 (00:00)
nmapscpe ssh:notty 134.17.94.27 Tue Jul 25 19:26 - 19:26 (00:00)
root ssh:notty 289.38.225.238 Tue Jul 25 19:26 - 19:26 (00:00)
jenkins ssh:notty 122.14.214.24 Tue Jul 25 19:25 - 19:25 (00:00)
jenkins ssh:notty 122.14.214.24 Tue Jul 25 19:25 - 19:25 (00:00)
sx ssh:notty 161.132.219.115 Tue Jul 25 19:25 - 19:25 (00:00)
admin ssh:notty 181.115.145.34 Tue Jul 25 19:25 - 19:25 (00:00)
sx ssh:notty 181.132.229.115 Tue Jul 25 19:25 - 19:25 (00:00)
admin ssh:notty 181.115.145.34 Tue Jul 25 19:25 - 19:25 (00:00)
root ssh:notty 43.135.156.188 Tue Jul 25 19:25 - 19:25 (00:00)
user ssh:notty 134.17.94.27 Tue Jul 25 19:25 - 19:25 (00:00)
user ssh:notty 134.17.94.27 Tue Jul 25 19:25 - 19:25 (00:00)
harish ssh:notty 212.178.240.195 Tue Jul 25 19:25 - 19:25 (00:00)
harish ssh:notty 212.178.240.195 Tue Jul 25 19:25 - 19:25 (00:00)
autcon ssh:notty 289.38.225.238 Tue Jul 25 19:24 - 19:24 (00:00)
root ssh:notty 218.92.0.76 Tue Jul 25 19:24 - 19:24 (00:00)
autcon ssh:notty 289.38.225.238 Tue Jul 25 19:24 - 19:24 (00:00)
root ssh:notty 218.92.0.76 Tue Jul 25 19:24 - 19:24 (00:00)
root ssh:notty 218.92.0.76 Tue Jul 25 19:24 - 19:24 (00:00)
root ssh:notty 218.92.0.76 Tue Jul 25 19:24 - 19:24 (00:00)
root ssh:notty 218.92.0.76 Tue Jul 25 19:24 - 19:24 (00:00)
root ssh:notty 139.59.255.59 Tue Jul 25 19:24 - 19:24 (00:00)
root ssh:notty 218.92.0.34 Tue Jul 25 19:24 - 19:24 (00:00)
root ssh:notty 218.92.0.34 Tue Jul 25 19:24 - 19:24 (00:00)
root ssh:notty 218.92.0.34 Tue Jul 25 19:24 - 19:24 (00:00)
admin ssh:notty 181.132.219.115 Tue Jul 25 19:24 - 19:24 (00:00)
root ssh:notty 218.92.0.34 Tue Jul 25 19:24 - 19:24 (00:00)
admin ssh:notty 181.115.145.34 Tue Jul 25 19:24 - 19:24 (00:00)
gitlab-r ssh:notty 181.115.145.34 Tue Jul 25 19:24 - 19:24 (00:00)
mailtest ssh:notty 43.156.42.52 Tue Jul 25 19:24 - 19:24 (00:00)
mailtest ssh:notty 43.156.42.52 Tue Jul 25 19:24 - 19:24 (00:00)
lailva ssh:notty 43.135.156.188 Tue Jul 25 19:24 - 19:24 (00:00)
lailva ssh:notty 43.135.156.188 Tue Jul 25 19:24 - 19:24 (00:00)
stacey ssh:notty 194.209.191.243 Tue Jul 25 19:23 - 19:23 (00:00)
stacey ssh:notty 194.209.191.243 Tue Jul 25 19:23 - 19:23 (00:00)
root ssh:notty 122.14.214.24 Tue Jul 25 19:23 - 19:23 (00:00)
user ssh:notty 114.34.95.216 Tue Jul 25 19:23 - 19:23 (00:00)
user ssh:notty 114.34.95.216 Tue Jul 25 19:23 - 19:23 (00:00)
root ssh:notty 43.156.42.52 Tue Jul 25 19:23 - 19:23 (00:00)
root ssh:notty 43.252.61.98 Tue Jul 25 19:23 - 19:23 (00:00)

```



```
File Actions Edit View Help
root sshnotty 43.156.42.52 Tue Jul 25 19:23 - 19:23 (00:00)
root sshnotty 43.252.61.98 Tue Jul 25 19:23 - 19:23 (00:00)
ftpuser1 sshnotty 154.232.219.215 Tue Jul 25 19:23 - 19:23 (00:00)
ftpuser1 sshnotty 161.152.219.115 Tue Jul 25 19:23 - 19:23 (00:00)
root sshnotty 194.209.191.243 Tue Jul 25 19:22 - 19:22 (00:00)
root sshnotty 165.22.98.246 Tue Jul 25 19:22 - 19:22 (00:00)
ku sshnotty 43.135.156.160 Tue Jul 25 19:22 - 19:22 (00:00)
root sshnotty 114.34.95.216 Tue Jul 25 19:22 - 19:22 (00:00)
ku sshnotty 43.135.156.160 Tue Jul 25 19:22 - 19:22 (00:00)
sysadmin sshnotty 222.255.115.237 Tue Jul 25 19:22 - 19:22 (00:00)
sysadmin sshnotty 222.255.115.237 Tue Jul 25 19:22 - 19:22 (00:00)
root sshnotty 43.156.42.52 Tue Jul 25 19:21 - 19:21 (00:00)
kafka sshnotty 43.252.61.98 Tue Jul 25 19:21 - 19:21 (00:00)
kafka sshnotty 43.252.61.98 Tue Jul 25 19:21 - 19:21 (00:00)
root sshnotty 14.32.8.74 Tue Jul 25 19:21 - 19:21 (00:00)
postgres sshnotty 194.209.191.243 Tue Jul 25 19:21 - 19:21 (00:00)
postgres sshnotty 194.209.191.243 Tue Jul 25 19:21 - 19:21 (00:00)
test sshnotty 165.22.98.246 Tue Jul 25 19:21 - 19:21 (00:00)
test sshnotty 165.22.98.246 Tue Jul 25 19:21 - 19:21 (00:00)
front sshnotty 114.34.95.216 Tue Jul 25 19:21 - 19:21 (00:00)
front sshnotty 114.34.95.216 Tue Jul 25 19:21 - 19:21 (00:00)
admin sshnotty 122.14.214.24 Tue Jul 25 19:21 - 19:21 (00:00)
admin sshnotty 122.14.214.24 Tue Jul 25 19:21 - 19:21 (00:00)
root sshnotty 218.92.0.112 Tue Jul 25 19:21 - 19:21 (00:00)
root sshnotty 218.92.0.112 Tue Jul 25 19:21 - 19:21 (00:00)
root sshnotty 218.92.0.112 Tue Jul 25 19:20 - 19:20 (00:00)
ptest sshnotty 222.255.115.237 Tue Jul 25 19:20 - 19:20 (00:00)
root sshnotty 218.92.0.112 Tue Jul 25 19:20 - 19:20 (00:00)
root sshnotty 222.255.115.237 Tue Jul 25 19:20 - 19:20 (00:00)
ubnt sshnotty 190.107.30.117 Tue Jul 25 19:20 - 19:20 (00:00)
ubnt sshnotty 190.107.30.117 Tue Jul 25 19:20 - 19:20 (00:00)
admin sshnotty 165.22.98.246 Tue Jul 25 19:20 - 19:20 (00:00)
ubnt sshnotty 134.228.197.28 Tue Jul 25 19:20 - 19:20 (00:00)
admin sshnotty 165.22.98.246 Tue Jul 25 19:20 - 19:20 (00:00)
duke sshnotty 14.32.8.74 Tue Jul 25 19:20 - 19:20 (00:00)
root sshnotty 194.209.191.243 Tue Jul 25 19:20 - 19:20 (00:00)
ubnt sshnotty 134.228.197.28 Tue Jul 25 19:20 - 19:20 (00:00)
duke sshnotty 14.32.8.74 Tue Jul 25 19:20 - 19:20 (00:00)
ftpuser sshnotty 134.17.94.27 Tue Jul 25 19:20 - 19:20 (00:00)
bot1 sshnotty 43.129.216.151 Tue Jul 25 19:20 - 19:20 (00:00)
ftpuser sshnotty 134.17.94.27 Tue Jul 25 19:20 - 19:20 (00:00)
bot1 sshnotty 43.129.216.151 Tue Jul 25 19:20 - 19:20 (00:00)
song sshnotty 212.178.240.195 Tue Jul 25 19:19 - 19:19 (00:00)
```

← → ↺

https://www.virustotal.com/gui/ip-address/43.156.42.52

🔍 ⬆️ 📄

URL, IP address, domain, or file hash

🔍 ⬆️ 📄

14

/ 88

📌

📌

🔴 14 security vendors flagged this IP address as malicious

Similar ▾

43.156.42.52 (43.156.0.0/15)

SG

AS 132203 (Tencent Building, Kejizhongyi Avenue)

🇨🇳

📉

Community Score

📈

DETECTION

DETAILS

RELATIONS

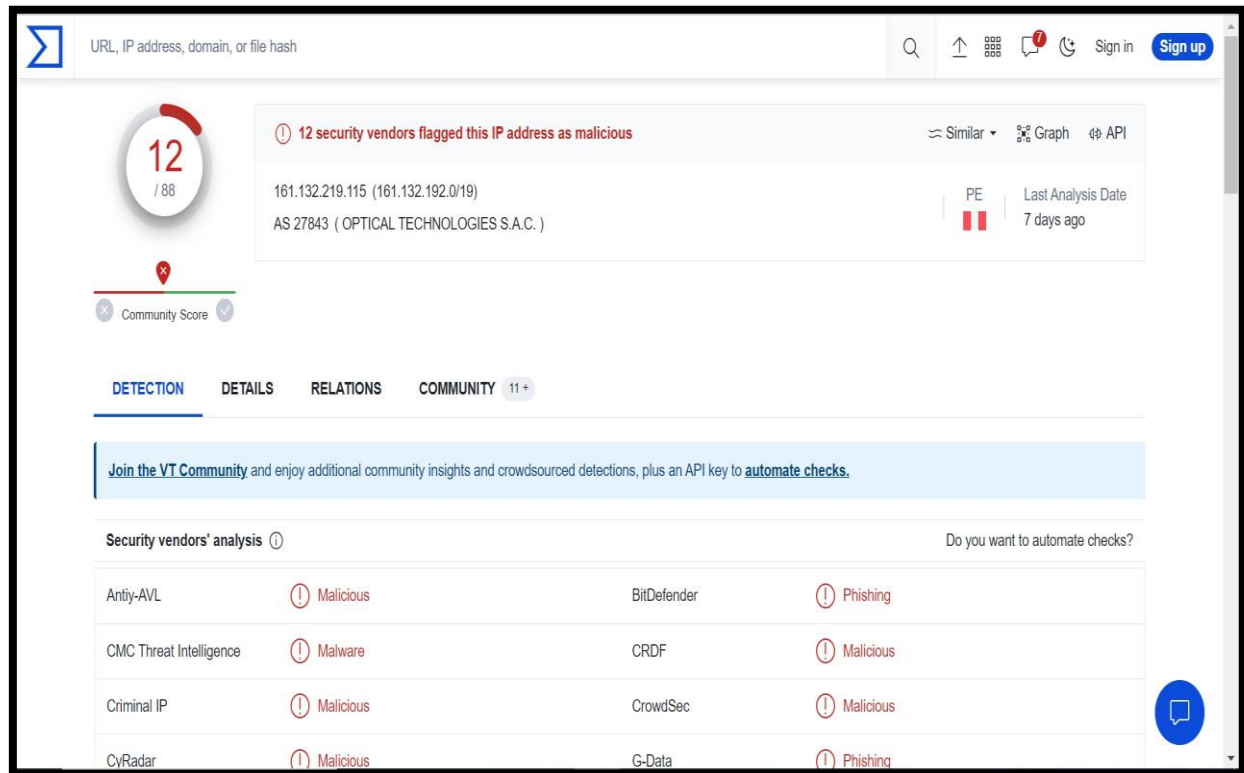
COMMUNITY 10

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis ⓘ

Do you want

Antiy-AVL	🔴 Malicious	BitDefender	🔴 Phishing
Cluster25	🔴 Malicious	CMC Threat Intelligence	🔴 Malware
CRDF	🔴 Malicious	Criminal IP	🔴 Malicious
CrowdSec	🔴 Malicious	Cyble	🔴 Malicious



PROBLEM SOLUTION DEBRIEF

The network infrastructure of the server is subject to brute force assaults, which might result in unauthorized access, data breaches, and resource waste. This compromises data integrity, operational continuity, and the reputation of the organization.

- **Vulnerability Assessment:** Perform a thorough examination of the server's network to detect potential weak points and entry points vulnerable to brute force assaults. This evaluation will offer a clear picture of the degree of vulnerability.
- **Detecting Brute Force Attacks:** Use powerful intrusion detection systems and monitoring tools to detect and report abnormal login attempts. This proactive technique will allow for real-time detection of possible brute force assaults.
- **FAIL2BAN implementation:** Use the FAIL2BAN software as a proactive defense measure. This program detects and bans IP addresses that make several failed login attempts, essentially blocking brute-force assaults.
- **Multi-Factor Authentication (MFA):** Use MFA to improve user account security. Even if an attacker obtains login credentials, MFA adds an additional authentication step, making unauthorized access far more difficult.
- **Regular Monitoring and Updates:** Keep an eye on the server's network for any strange behavior or signals of an assault. Update the server's software and security protocols often to remain on top of new threats.

- **Employee Training:** Educate server users and workers on the need of using strong, unique passwords and practicing good security hygiene to reduce the chance of successful brute-force attacks.
- **Develop a detailed incident response plan** that describes the measures to follow in the event of a detected brute force assault. This strategy should contain protocols for communication, isolation measures, and rehabilitation techniques.
- **Configuration and Integration:** Configure FAIL2BAN to meet the unique security needs of the server. Integrate the program into the server's architecture easily to ensure effective functioning without interfering with usual tasks.

DETAILS OF SOFTWARE FAIL2BAN

HOW ITS WORK

Fail2Ban operates by continually monitoring log files created by various server services such as SSH, FTP, web servers, and others. It searches these log files for unsuccessful login attempts or other suspicious activity patterns and then takes predetermined measures to stop or mitigate the risks discovered. Here's a detailed breakdown of how Fail2Ban works:

- **Monitoring of Log Files:** Fail2Ban continually monitors log files in real-time. These log files provide data regarding different server actions such as login attempts, authentication failures, and other occurrences.
- **Pattern Detection:** Fail2Ban analyses log files for certain patterns of behavior using regular expressions and specified rules. For example, it may search for a pattern of failed login attempts from the same IP address over a short period of time.
- **Issue Identification:** When Fail2Ban detects a pattern that meets one of its established rules, it considers it a security issue. These occurrences might involve frequent unsuccessful login attempts, repeated access to non-existent resources, or other questionable activity.
- **Triggering Actions:** When an event is recognized, Fail2Ban acts depending on the rules provided in its configuration. This usually entails blocking the IP address connected with questionable behavior. The block can be implemented using firewall rules to restrict further communication from that IP address.
- **Fail2Ban imposes a temporary ban** on the offending IP address for a preset length of time. This temporary ban is intended to deter automated assaults while still allowing legitimate users to utilize the service once the ban expires.

BENEFITS OF FAILED2BAN:

- **Threat Mitigation:** Fail2ban's automatic blocking of malicious IPs helps prevent unauthorized access and reduces the risk of successful attacks.
- **Prevents Brute-Force Attacks:** FAIL2BAN mitigates brute-force attacks by automatically banning IP addresses attempting to guess passwords or exploit vulnerabilities.
- **Reduced Workload:** By automating the detection and response to threats, Fail2ban alleviates the burden on system administrators, allowing them to focus on other critical tasks.
- **Customizable:** The ability to create custom filters and rules empowers you to tailor Fail2ban to your server's specific security requirements.
- **Cost-Effective:** Fail2ban is open-source software, meaning there are no licensing fees. This makes it a cost-effective solution for enhancing server security.
- **Proactive Defense:** Fail2ban doesn't just detect attacks; it actively blocks them in real-time, minimizing the window of opportunity for attackers.
- **Protection Against DoS Attacks:** It provides some level of protection against Denial of Service (DoS) attacks by blocking IPs that repeatedly generate excessive traffic.

WHY WE ARE USING FAIL2BAN

FAIL2BAN to bolster the security of our systems and services. Its proactive approach helps in preventing brute-force attacks, protecting against DoS attacks to some extent, and reducing the risk of unauthorized access. By utilizing this software, we add an extra layer of defense to our infrastructure, improving overall security and minimizing the chances of successful cyber-attacks. The software's ability to handle multiple services and its ease of configuration make it a popular choice for enhancing security on servers and services.

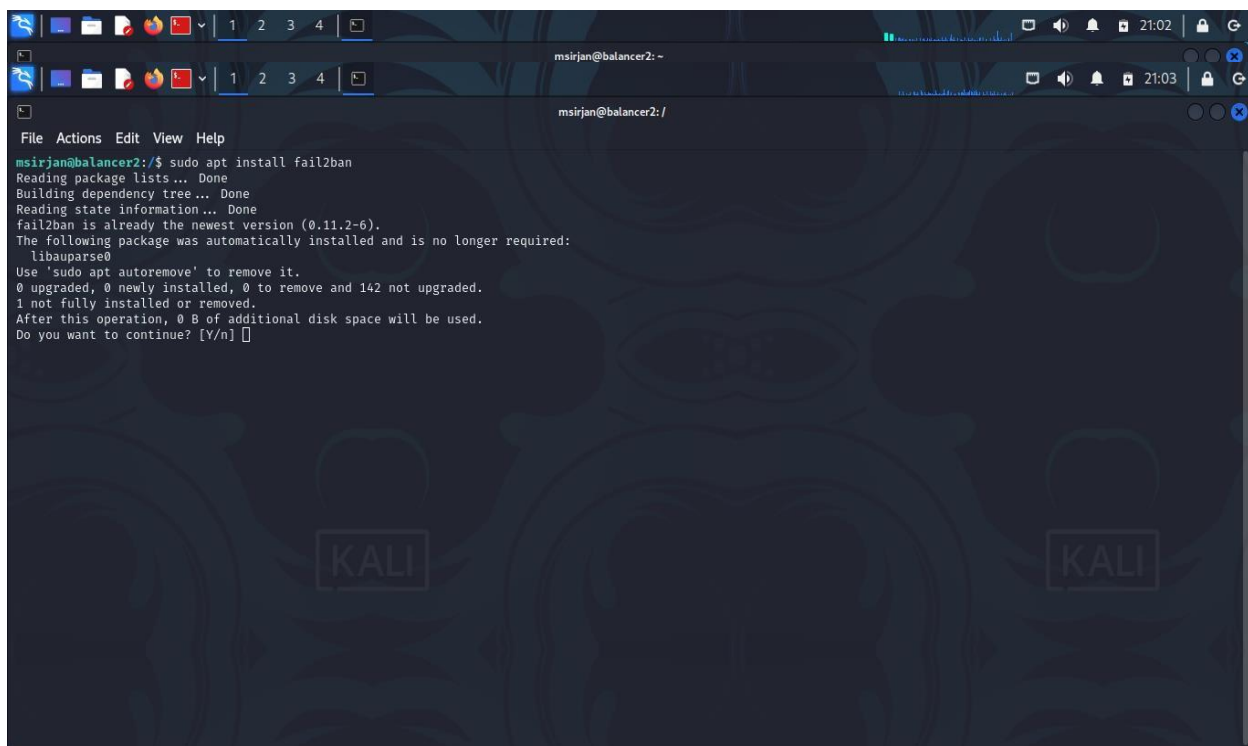
FAIL2BAN is a valuable tool that proactively defends against common cyber threats, adds an extra layer of security to systems, and reduces the risk of unauthorized access to critical services. However, it's essential to keep the software updated and fine-tune the rules periodically to stay ahead of evolving threats.

WHY IS BETTER THAN OTHERS

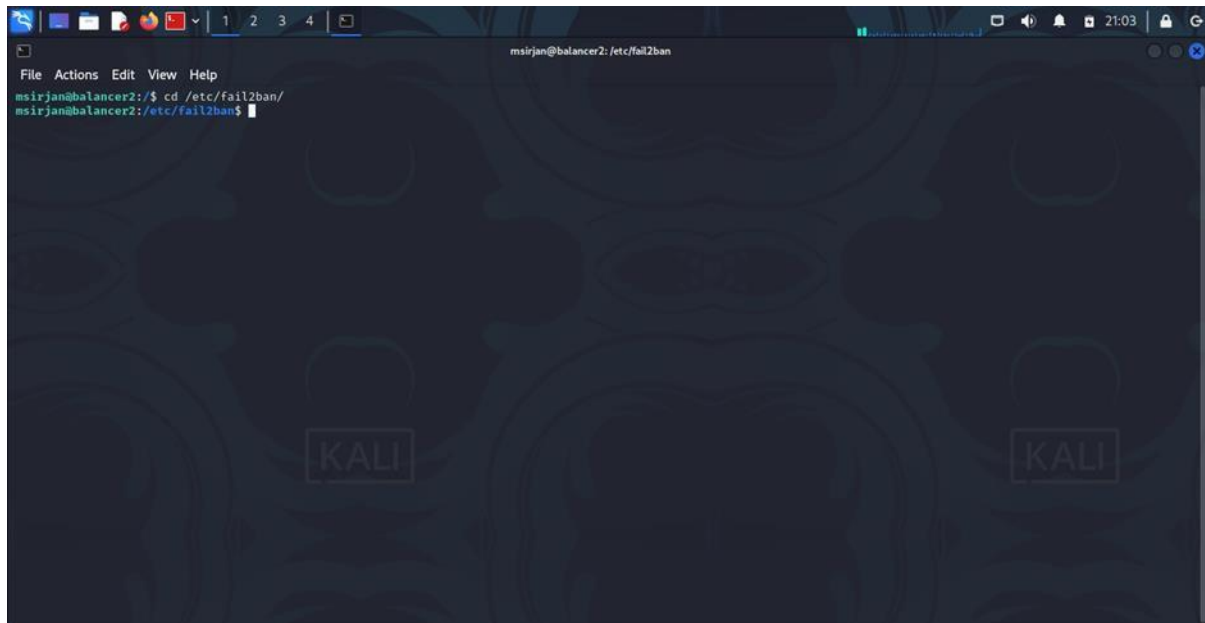
- **Automation:** FAIL2BAN automates the process of detecting and blocking malicious activities, reducing the need for manual intervention. This automation saves time and resources for system administrators.
- **Lightweight:** It is a relatively lightweight software that does not consume excessive system resources, making it suitable for use on various types of servers and systems.

- **Flexibility:** Administrators can customize and fine-tune the rules according to their specific security needs, providing a high level of flexibility and adaptability to different environments.
- **Wide Range of Services:** FAIL2BAN supports a wide range of services and protocols, making it versatile in protecting various server applications, such as SSH, FTP, Apache, Nginx, and more.
- **Effective Deterrent:** The software acts as a strong deterrent against potential attackers, as they know their IP addresses will be automatically blocked if they attempt unauthorized access.
- **Real-time Response:** FAIL2BAN responds in real-time to security threats, promptly blocking suspicious IPs, thereby reducing the window of opportunity for attackers.

IMPLEMENTATION OF SOFTWARE FAIL2BAN

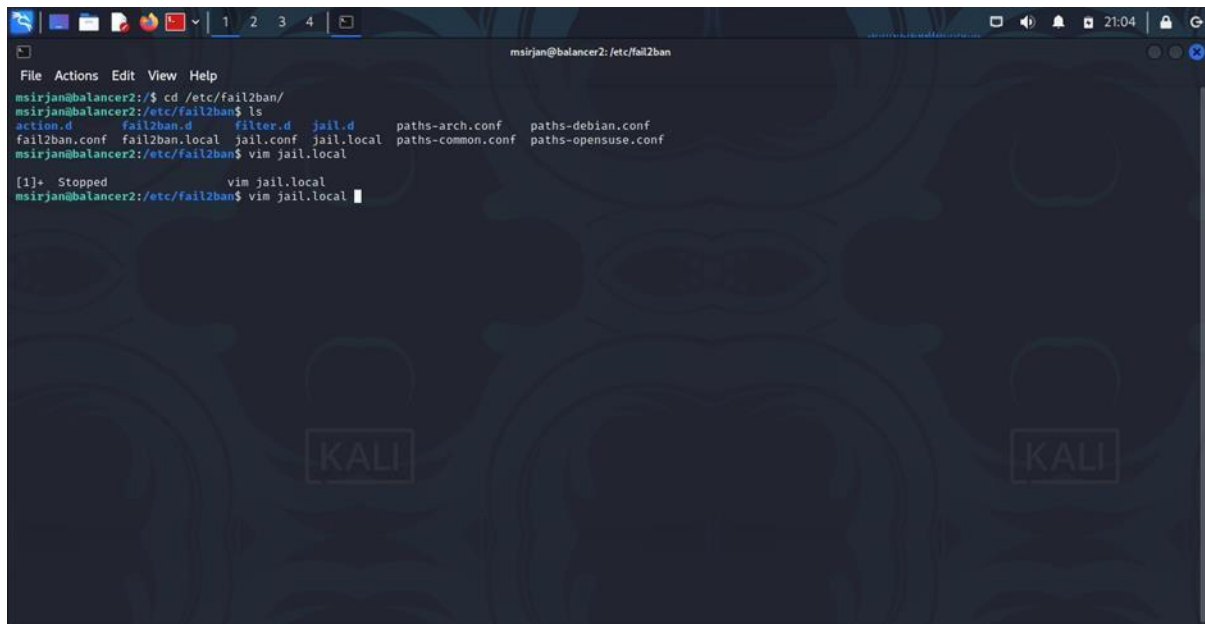


```
msirjan@balancer2: ~  
msirjan@balancer2: /  
File Actions Edit View Help  
msirjan@balancer2:/$ sudo apt install fail2ban  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
fail2ban is already the newest version (0.11.2-6).  
The following package was automatically installed and is no longer required:  
  libauparse0  
Use 'sudo apt autoremove' to remove it.  
0 upgraded, 0 newly installed, 0 to remove and 142 not upgraded.  
1 not fully installed or removed.  
After this operation, 0 B of additional disk space will be used.  
Do you want to continue? [Y/n]
```



A terminal window titled "msirjan@balancer2: /etc/fail2ban" with a menu bar (File, Actions, Edit, View, Help). The terminal shows the user navigating to the /etc/fail2ban directory. The background features a Kali Linux watermark.

```
msirjan@balancer2:/$ cd /etc/fail2ban/  
msirjan@balancer2:/etc/fail2ban$
```



A terminal window titled "msirjan@balancer2: /etc/fail2ban" with a menu bar (File, Actions, Edit, View, Help). The terminal shows the user listing files in the /etc/fail2ban directory, then opening jail.local with vim. The background features a Kali Linux watermark.

```
msirjan@balancer2:/$ cd /etc/fail2ban/  
msirjan@balancer2:/etc/fail2ban$ ls  
action.d      fail2ban.d      filter.d      jail.d          paths-arch.conf  paths-debian.conf  
fail2ban.conf fail2ban.local  jail.conf     jail.local      paths-common.conf paths-opensuse.conf  
msirjan@balancer2:/etc/fail2ban$ vim jail.local  
[1]+  Stopped                  vim jail.local  
msirjan@balancer2:/etc/fail2ban$ vim jail.local
```

```
msirjan@balancer2: /etc/ fail2ban
File Actions Edit View Help
[DEFAULT]
ignoreip = 127.0.0.1/8 10.0.0.0/8 69.90.74.129/27 69.90.202.96/27 72.38.111.130/32
bantime = 500
findtime = 500
maxretry = 2

[sshd]

port = ssh
logpath = %(sshd_log)s
enabled = true

"jail.local" [readonly] 11L, 1998
4,12 All

"jail.local" [readonly] 11L, 1998
5,1 All
```

```
root@kali: ~
File Actions Edit View Help
root@kali: ~
$ ssh msirjan@69.90.202.100
msirjan@69.90.202.100's password:
Permission denied, please try again.
msirjan@69.90.202.100's password:
Permission denied, please try again.
msirjan@69.90.202.100's password:
msirjan@69.90.202.100: Permission denied (publickey,password).

root@kali: ~
$ ssh msirjan@69.90.202.100
msirjan@69.90.202.100's password:
Permission denied, please try again.
msirjan@69.90.202.100's password:
Permission denied, please try again.
msirjan@69.90.202.100's password:
msirjan@69.90.202.100: Permission denied (publickey,password).

root@kali: ~
$ ssh msirjan@69.90.202.100
ssh: connect to host 69.90.202.100 port 22: Connection refused

root@kali: ~
```

CONGREGATION IN SSHD_CONF

We changed PubkeyAuthentication and permitrootlogin. We disable the KerberosAuthentication and GSSAPIAuthentication. Further, we enable IgnoreUserKnownHosts and StrictModes. In addition to improving the server security, we disable some forwarding such as AllowAgentForwarding, AllowTcpForwarding, and X11Forwarding. Also, we disabled PrintMotd and Permit User Environment.

```
ChallengeResponseAuthentication no
UsePAM yes
PubkeyAuthentication no
PermitRootLogin no
PasswordAuthentication yes
PidFile /run/sshd.pid
#   $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
```



```
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin prohibit-password
StrictModes yes
MaxAuthTries 3
#MaxSessions 10

PubkeyAuthentication no

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
HostbasedAuthentication
IgnoreUserKnownHosts yes
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
```

Kerberos options

KerberosAuthentication yes

#KerberosOrLocalPasswd yes

#KerberosTicketCleanup yes

#KerberosGetAFSToken no

GSSAPI options

GSSAPIAuthentication yes

#GSSAPICleanupCredentials yes

#GSSAPIStrictAcceptorCheck yes

#GSSAPIKeyExchange no

Set this to 'yes' to enable PAM authentication, account processing,

and session processing. If this is enabled, PAM authentication will

be allowed through the ChallengeResponseAuthentication and

PasswordAuthentication. Depending on your PAM configuration,

PAM authentication via ChallengeResponseAuthentication may bypass

the setting of "PermitRootLogin without-password".

If you just want the PAM account and session checks to run without

PAM authentication, then enable this but set PasswordAuthentication

and ChallengeResponseAuthentication to 'no'.

AllowAgentForwarding no

AllowTcpForwarding no

#GatewayPorts no

X11Forwarding no

#X11DisplayOffset 10

#X11UseLocalhost yes

#PermitTTY yes

PrintMotd no

#PrintLastLog yes

#TCPKeepAlive yes

PermitUserEnvironment no

#Compression delayed

#ClientAliveInterval 0

#ClientAliveCountMax 3

#UseDNS no

```
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
    X11Forwarding no
    AllowTcpForwarding no
#    PermitTTY no
#    ForceCommand cvs server
```

IMPLEMENTATION PLAN

Week 1: July 18 - July 25

Project Kick-Off and Client meetup

- Sign off Scope with the client about the project.
- Getting information about the project from the client
- Getting access to GitLab and doing the Nmap search
- Clearly define the target hosts or networks using IP addresses or domain names.

Week 2: July 25 - July 29

- Monitor login attempts and successful logins.
- Identify unusual or suspicious login behavior.
- Enhance our understanding of SSH usage patterns.
- Investigating Vulnerabilities through Last Failed Login Analysis

- Identify common targets or specific accounts that are frequently targeted.
- Determine whether failed login attempts correspond with any known security vulnerabilities.

Week 3 July 29 – August 05

- Finalizing software failed2ban (Tool) to stop brute force attacks.
- Filtered the open port.
- Install and configure Fail2Ban, a tool that monitors log files for failed login attempts and can block IP addresses temporarily.

Week 4: August 5 – August 10

- Set up Fail2Ban rules to work in conjunction with MaxTries to provide an additional layer of protection.
- Consider implementing time-based restrictions, such as allowing only a certain number of logins attempts within a specific time frame (e.g., 3 attempts in 5 minutes).
- We will seamlessly integrate 2FA mechanisms into our authentication processes, requiring users to provide a second form of verification.
- We leverage TCP wrappers to meticulously manage incoming connection requests, permitting access solely from specified hosts or networks.
- Effective traffic control is instrumental in optimizing network performance and safeguarding against potential threats.

Week 5: August 10- August 14

- Go live with the new system.
- Provide post-implementation support to address any issues or concerns.

Alixander Greganti

Alixander Greganti

08/14/2023



Project Topic: **Network Security and Administrative Software Implementation**

SCOPE

To creating a server infrastructure policy, implementing a security strategy, and conducting network hardening to enhance security and minimize risks.

PROBLEM STATEMENT

The problem statement highlights the discovery of multiple open ports and a brute force attack vulnerability in Auxilium Group's network, underscoring the need for comprehensive security testing and reinforcing the organization's defenses.

OPTIONS REVIEWED

Conducting a comprehensive Vulnerability Assessment, implementing intrusion detection systems, utilizing the FAIL2BAN software, incorporating Multi-Factor Authentication (MFA), ensuring regular monitoring and updates, providing employee training, creating an incident response plan, and configuring/integrating FAIL2BAN to match server security requirements.

THE FUTURE STUDY AND DIRECTION INVOLVE THE CONTINUED IMPLEMENTATION AND UTILIZATION OF FAIL2BAN TO ENHANCE SYSTEM SECURITY BY PREVENTING BRUTE-FORCE AND DOS ATTACKS, REDUCING UNAUTHORIZED ACCESS RISKS, AND ENSURING AN ADDITIONAL LAYER OF DEFENSE, WITH A FOCUS ON REGULAR UPDATES AND RULE ADJUSTMENTS TO STAY EFFECTIVE AGAINST EVOLVING CYBER THREATS.

FUTURE DIRECTIONS