Brute Force Attack Detection on SSH Logins

2320030172 - Manikanta[1], 2320030033 - V. Ajay Charan[2], and 2320030003 - K. Manjith Reddy[3]

KL University Bachupally

Abstract. Brute-force attacks on SSH services are a prevalent and dangerous security threat that can compromise the integrity of network systems. This paper presents an effective detection approach using log analysis and threshold-based detection mechanisms to identify and mitigate such attacks before they cause significant damage.

Keywords: Brute Force Attack, SSH Security, Log Analysis, Intrusion Detection, Cybersecurity.

1 Introduction

Secure Shell (SSH) is a widely used protocol for remote system administration and secure file transfers. However, due to its importance, it is frequently targeted by malicious attackers who attempt brute-force attacks to gain unauthorized access. These attacks involve automated scripts attempting numerous login attempts using different password combinations. Detecting and preventing such attacks in real-time is crucial to maintaining the security and confidentiality of sensitive systems.

1.1 Detection and Mitigation of SSH Brute-Force Attacks

Our approach to mitigating brute-force attacks involves real-time monitoring of SSH logs located in `/var/log/auth.log`. By analyzing authentication failures and correlating multiple failed login attempts originating from the same IP address within a short time frame, we can effectively identify potential attack sources. This proactive detection mechanism helps administrators take immediate action, such as blocking the malicious IP or implementing additional authentication layers.

Python Script for SSH Brute-Force Attack Detection The following Python script demonstrates how to scan SSH authentication logs to detect potential brute-force attacks. The script reads log files, extracts failed login attempts, and counts the number of times a particular IP address has attempted to log in. If the number of failed attempts exceeds a defined threshold, the script flags the IP as a possible attacker, enabling administrators to take necessary security measures.

Listing . Python Script for SSH Brute Force Detection

```
import re
import os
import time
from collections import defaultdict

LOG_FILE = "/var/log/auth.log"
THRESHOLD = 5
BAN_TIME = 3600
BLOCKED_IPS_FILE = "/tmp/blocked_ips.log"

ip_attempts = defaultdict(int)

with open(LOG_FILE, "r") as file:
    for line in file:
        match = re.search(r"Failed␣password␣for␣.*␣from␣
            (\d+\.\d+\.\d+\.\d+)", line)
        if match:
            ip = match.group(1)
            ip_attempts[ip] += 1

blocked_ips = {}

if os.path.exists(BLOCKED_IPS_FILE):
    with open(BLOCKED_IPS_FILE, "r") as file:
        for line in file:
            ip, block_time = line.strip().split()
            blocked_ips[ip] = int(block_time)

for ip, count in ip_attempts.items():
    if count >= THRESHOLD and ip not in blocked_ips:
        print(f"[ALERT]␣Blocking␣IP:␣{ip}")
        os.system(f"sudo␣iptables␣-A␣INPUT␣-s␣{ip}␣-j␣
            DROP")
        blocked_ips[ip] = int(time.time())

with open(BLOCKED_IPS_FILE, "w") as file:
    for ip, block_time in blocked_ips.items():
        if int(time.time()) - block_time >= BAN_TIME:
            print(f"[INFO]␣Unblocking␣IP:␣{ip}")
            os.system(f"sudo␣iptables␣-D␣INPUT␣-s␣{ip}␣-j
                ␣DROP")
        else:
            file.write(f"{ip}␣{block_time}\n")
```

Theorem 1. *Brute-force attacks on SSH can be effectively detected by contin-*

uously monitoring system logs and identifying patterns of repeated failed login attempts from a single source IP. This method provides a reliable mechanism to prevent unauthorized access attempts and protect system integrity.

Acknowledgments

The authors would like to extend their gratitude to KL University Bachupally for providing the necessary resources and support to conduct this research. We also acknowledge the contributions of our peers and faculty members, whose valuable feedback has significantly improved the quality of this work.

Disclosure of Interests

The authors declare that they have no competing interests that could have influenced the content or conclusions of this study.

References

1. Trend Micro: What is SSH Brute Force Attack And How To Deal With It. Trend Micro Help Center. Available at: magentahttps://helpcenter.trendmicro.com/en-us/article/tmka-19689
2. CrowdSec: Detect Successful SSH Brute Force Attacks. CrowdSec Blog. Available at: magentahttps://www.crowdsec.net/blog/detecting-successful-ssh-brute-force
3. Jadaptive: Understanding and Mitigating SSH Brute Force Attacks. Available at: magentahttps://jadaptive.com/java-ssh-library/understanding-and-mitigating-ssh-brute-force-attacks/
4. Elastic: Potential Successful SSH Brute Force Attack. Elastic Security Solution [8.17]. Available at: magentahttps://www.elastic.co/guide/en/security/current/potential-successful-ssh-brute-force-attack.html
5. Carnegie Mellon University: Brute-force/Dictionary SSH Attacks. Information Security Office. Available at: magentahttps://www.cmu.edu/iso/aware/be-aware/brute-force_ssh_attack.html
6. Sucuri Blog: How to Prevent SSH Brute Force Login Attacks. Available at: magentahttps://blog.sucuri.net/2023/04/how-to-prevent-ssh-brute-force-login-attacks.html
7. Wikipedia: DenyHosts. Available at: magentahttps://en.wikipedia.org/wiki/DenyHosts
8. Wikipedia (Chinese): DenyHosts. Available at: magentahttps://zh.wikipedia.org/wiki/DenyHosts