Relatório Intrusion Prevention System (IPS) Contra Ataques em Rede Local

Objetivo

O objetivo deste relatório é apresentar o desenvolvimento e a implementação de um Intrusion Prevention System (**IPS)** em Python, integrado ao **Fail2Ban**, com foco na detecção e mitigação automatizada de ataques direcionados a serviços **SSH e Telnet via SSH**. O IPS visa identificar e bloquear permanentemente tentativas de brute force, conexões suspeitas e atividades anômalas, garantindo uma camada adicional de segurança para sistemas baseados em Linux.

A implementação busca oferecer um sistema eficiente e aplicável no dia a dia, capaz de neutralizar ameaças instantaneamente e impedir que atacantes continuem suas tentativas após a detecção. O relatório detalha o funcionamento do IPS, suas regras de detecção e a integração com o **Fail2Ban**, demonstrando como essa abordagem pode ser utilizada para fortalecer a segurança de redes locais contra invasões e acessos não autorizados.

Autor: Enzo Arrue Juan Fuso
Instituição: Fatec São Caetano do Sul
Curso: Segurança da Informação

Introdução

Com o aumento constante das ameaças cibernéticas, a proteção de sistemas contra ataques de força bruta e acessos não autorizados tornou-se uma prioridade em ambientes de rede. Entre as técnicas mais comuns de invasão, os ataques de brute force SSH e tentativas de exploração de Telnet via SSH são amplamente utilizados, visando a obtenção de credenciais de acesso através de tentativas repetidas e automatizadas. Essas ameaças representam um risco significativo para servidores e dispositivos em redes locais, especialmente em sistemas com serviços críticos expostos à internet.

Neste contexto, a implementação de um Intrusion Prevention System (IPS) é fundamental para a defesa ativa contra esses ataques. O uso do Fail2Ban, uma ferramenta amplamente reconhecida para mitigação de tentativas de acesso indevido, pode ser potencializado quando integrado a soluções customizadas que permitam uma resposta mais rápida e adaptável às ameaças em tempo real. O presente trabalho propõe a criação de um IPS em Python, com foco na automação da detecção e bloqueio de ataques SSH e Telnet, utilizando a integração com o Fail2Ban para aumentar a eficácia do sistema de segurança.

A proposta de um sistema inteligente e reativo visa garantir não apenas a identificação das tentativas de ataque, mas também a bloqueio permanente de IPs que realizam tentativas maliciosas, evitando acessos futuros, mesmo quando as credenciais estejam corretas. O sistema desenvolvido será analisado em detalhes, com ênfase em sua eficácia na prevenção de ataques em tempo real e na importância de soluções como o Fail2Ban para a proteção de sistemas baseados em Linux.

Ferramentas/Softwares Utilizados

VirtualBox, Terminal Linux (Comandos), Fail2Ban

Pré-requisitos:

Comandos no terminal:

- sudo apt update && sudo apt install fail2ban -y
- sudo systemctl enable fail2ban
- sudo systemctl start fail2ban

IPS e Teste do Sistema

OBS: O código pode não funcionar se usar a função de “copiar e colar”, pois cada máquina/SO possui suas características, como nomes de diretórios diferentes, arquivos, etc… Para isso faça um pequeno ajuste, se necessário.

```python
import subprocess
import time
import re

# Arquivo de log do SSH
LOG_FILE = "/var/log/auth.log"

# Expressão regular para identificar tentativas de login falhas
SSH_FAIL_REGEX = r"Failed password for .* from (\d+\.\d+\.\d+\.\d+)"

# Configuração do Fail2Ban
FAIL2BAN_JAIL = "sshd"

# Lista para evitar bloqueios repetidos
blocked_ips = set()

def monitor_ssh_log():
    """Monitora o log do SSH e detecta tentativas de Brute Force."""
    with subprocess.Popen(["tail", "-F", LOG_FILE], stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True) as proc:
        for line in proc.stdout:
            match = re.search(SSH_FAIL_REGEX, line)
            if match:
                ip = match.group(1)
                if ip not in blocked_ips:
                    print(f"[ALERTA] Ataque de Brute Force detectado! IP: {ip}")
                    block_ip(ip)
                    blocked_ips.add(ip)
```

```python
def block_ip(ip):
    """Bloqueia o IP usando Fail2Ban."""
    try:
        subprocess.run(["fail2ban-client", "set", FAIL2BAN_JAIL, "banip", ip], check=True)
        print(f"[BLOQUEADO] IP {ip} foi banido com sucesso!")
    except subprocess.CalledProcessError as e:
        print(f"[ERRO] Falha ao banir {ip}: {e}")

if __name__ == "__main__":
    print("[IPS SSH ATIVO] Monitorando tentativas de Brute Force...")
    monitor_ssh_log()
```

Após finalizado o código em sua máquina cliente no editor do Ubuntu, devemos testar. Para isso, salve o código com um nome que deseja com o final “.py”, e execute o comando no terminal “sudo python3 {nome_do_arquivo.py}”, se tudo ocorrer conforme o esperado deverá ser exibido uma mensagem na tela dizendo que o IPS está monitorando o tráfego na rede.

Agora em outra máquina virtual, abra uma máquina atacante. Vamos realizar um comando já visto, no terminal:

- sudo apt update && sudo apt install hydra -y
- sudo apt install wordlists -y
- gunzip /usr/share/wordlists/rockyou.txt.gz
- sudo hydra -l {nome_do_usuario} -P /usr/share/wordlists/rockyou.txt ssh://{IP}

Conclusão

A implementação de um Intrusion Prevention System (IPS) para proteger redes locais mostrou que é uma solução eficiente contra ameaças como brute force em serviços SSH e Telnet via SSH. O uso do Fail2Ban ajudou bastante, permitindo bloquear automaticamente IPs maliciosos e garantir a segurança do sistema sem necessidade de intervenção manual.

A ideia principal foi analisar os logs do sistema para identificar padrões suspeitos e bloquear automaticamente conexões potencialmente perigosas. Com a integração do IPS com o Fail2Ban, conseguimos criar uma lista dinâmica de bloqueios, evitando que ataques repetidos continuem. Como o sistema foi desenvolvido em Python, também fica mais fácil de adaptar e expandir caso novas ameaças surjam no futuro.

Nos testes, observamos que ataques de força bruta usando ferramentas como Hydra foram detectados e bloqueados com sucesso. Depois do bloqueio, o atacante não conseguiu mais tentar logar, mesmo que usasse credenciais corretas. Isso é essencial para sistemas que precisam de conexão remota sem comprometer a segurança.

A prevenção ativa desse tipo de ataque é muito importante, já que tentativas de força bruta ainda são bastante comuns. Usar logs para tomar decisões automáticas ajuda a diminuir os riscos e dificulta bastante a vida dos invasores.

No futuro, seria interessante adicionar recursos mais avançados, como análise de comportamento com machine learning para detectar ataques de forma ainda mais precisa. Também poderíamos expandir o IPS para proteger outros protocolos e serviços.

Com isso, conseguimos provar que um IPS baseado em regras pode ser muito eficaz para proteger redes locais contra ataques. Além disso, o projeto reforça a importância da automação e da prevenção na segurança cibernética nos dias de hoje.