Report: SSH Reverse Shell Using IPFire Firewall

Objective

This report details the implementation of an SSH reverse shell setup using IPFire as a firewall to establish secure communication between two devices. The setup ensures that the devices communicate through the firewall without a direct connection between them. The devices use the firewall as an intermediary to establish and maintain the reverse SSH connection.

Components Involved

1. IPFire Firewall:

Acts as the network management gateway.
Provides a secure environment for communication.
Facilitates NAT and port forwarding for SSH reverse connections.

2. Device A(VM):

Initiates the reverse SSH connection to the IPFire firewall.

3. Device B(VM)

Connects to the IPFire firewall to establish communication with Device A.

4. Network Topology:

Devices are configured to connect to the firewall's IP address.
No direct connection exists between Device A and Device B.

Methodology

1. Setting up IPFire Firewall

Network Configuration:

Assign static IPs to the firewall's interfaces (e.g., GREEN and RED zones).
Enable SSH access on the IPFire firewall.

Port Forwarding:

Configure port forwarding rules to allow incoming SSH connections from Device A and Device B.

Firewall Rules:

Create rules to permit SSH traffic (port 22) between the devices and the firewall.

2. Configuring Devices

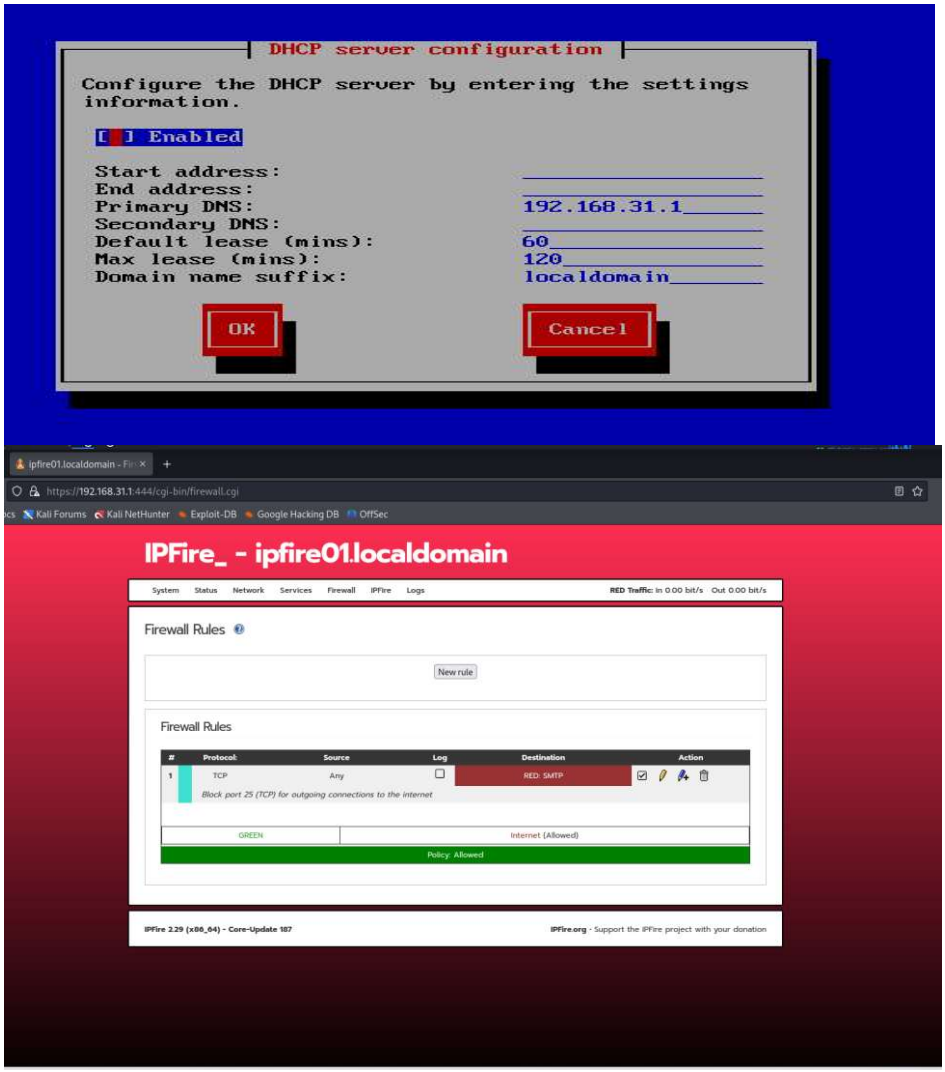
Device A (Reverse SSH Client):
ssh -R 2222:localhost:22 user@<IPFire_IP>

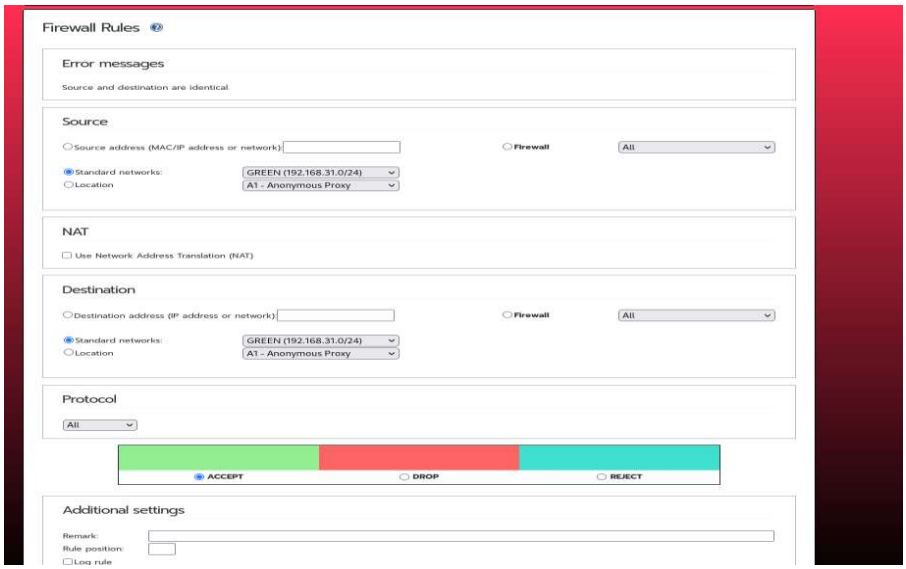
Device B (Forward Connection)
ssh -p 2222 user@<IPFire_IP>

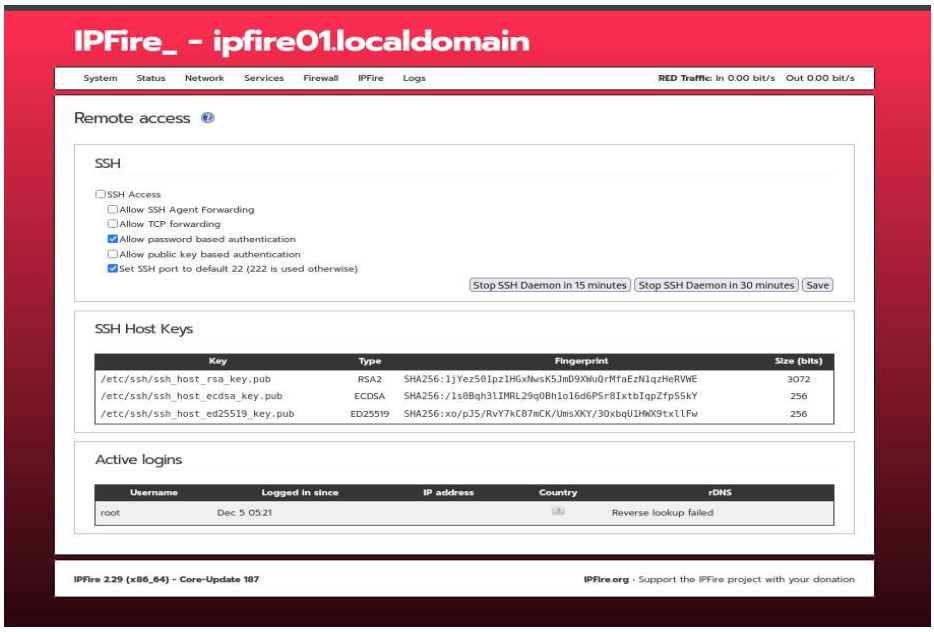
Workflow

1. Initiation by Device A:
   Device A establishes a reverse SSH tunnel to the IPFire firewall.
2. Device B Connection:
   Device B connects to the firewall on the port mapped by Device A's reverse SSH tunnel.
3. Secure Communication:
   All communication between Device A and Device B is routed through the IPFire firewall.

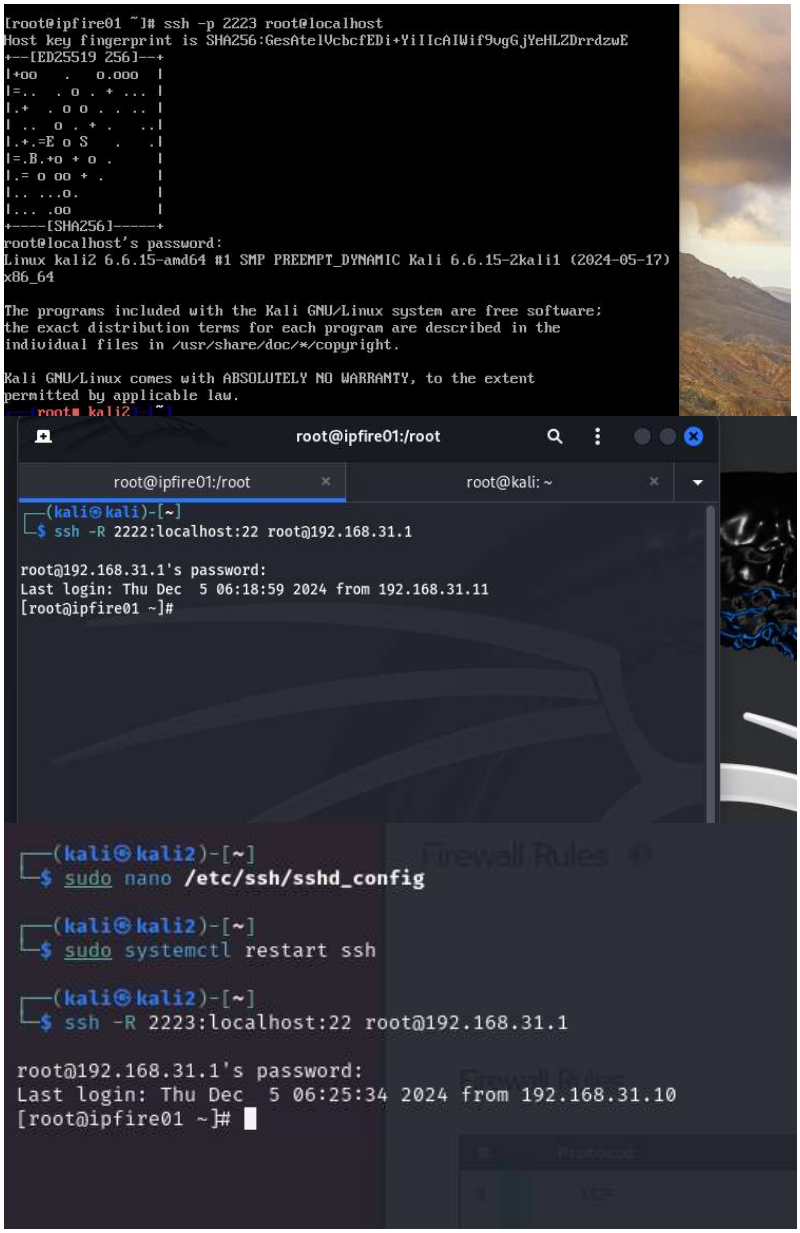
Proof of concept




Firewall Rules
Error messages
Source and destination are identical.
Source
Source address (MAC/IP address or network):
Firewall
All
Standard networks:
GREEN (192.168.31.0/24)
Location
A1 - Anonymous Proxy
NAT
Use Network Address Translation (NAT)
Destination
Destination address (IP address or network):
Firewall
All
Standard networks:
GREEN (192.168.31.0/24)
Location
A1 - Anonymous Proxy
Protocol
All
ACCEPT
DROP
REJECT
Additional settings
Remark:
Rule position:
Log rule


IPFire_ - ipfire01.localdomain
System
Status
Network
Services
Firewall
IPFire
Logs
RED Traffic: In 0.00 bit/s Out 0.00 bit/s
Remote access
SSH
SSH Access
Allow SSH Agent Forwarding
Allow TCP forwarding
Allow password based authentication
Allow public key based authentication
Set SSH port to default 22 (222 is used otherwise)
Stop SSH Daemon in 15 minutes
Stop SSH Daemon in 30 minutes
Save
SSH Host Keys
Key
Type
Fingerprint
Size (bits)
/etc/ssh/ssh_host_rsa_key.pub
RSA2
SHA256:1jYez501pz1HGxNwsK5JmD9XWuQrMfaEzN1qzHeRVWE
3072
/etc/ssh/ssh_host_ecdsa_key.pub
ECDSA
SHA256:/1s0Bqh3lIMRL29q0Bhlo16d6PSr8IxtbIqpZfpSSkY
256
/etc/ssh/ssh_host_ed25519_key.pub
ED25519
SHA256:xo/pJ5/RvY7kC87mCK/UmsXKY/3OxbqU1HWX9txllFw
256
Active logins
Username
Logged in since
IP address
Country
rDNS
root
Dec 5 05:21
Reverse lookup failed
IPFire 2.29 (x86_64) - Core-Update 187
IPFire.org - Support the IPFire project with your donation


SSH
SSH Access
Allow SSH Agent Forwarding
Allow TCP forwarding
Allow password based authentication
Allow public key based authentication
Set SSH port to default 22 (222 is used otherwise)



Conclusion

Using IPFire as a firewall for SSH reverse shell communication provides a robust and secure method to link devices without direct exposure. This configuration is particularly useful in scenarios where device isolation and centralized control are priorities. Further

optimization can include implementing VPN tunnels and automated connection scripts for streamlined operations.