

Server Resource Monitoring and Security Documentation

Date: December 09-12-2024

❖ Table of Contents

1. Server Resource Monitoring Setup
2. Email Configuration
3. Resource Monitoring Script
4. Testing Procedures
5. Failed Login Monitoring

1. Server Resource Monitoring Setup

- Update system packages:
#sudo apt update
#sudo apt upgrade -y
- Install required packages:
#sudo apt install mailutils -y

2. Email Configuration

- Configure Gmail SMTP settings:
#sudo nano /etc/mail.rc
- Add the following configuration:
set smtp=smtp://smtp.gmail.com:587
set smtp-auth-user=your-email@gmail.com [Add your mail here]
set smtp-auth-password=your-app-password
[Add your auth pass here (etch otmr ijww uuwa)]
set smtp-auth=login
set from=your-email@gmail.com
- Configure Postfix:
#sudo nano /etc/postfix/main.cf

- Add these settings:

relayhost = [smtp.gmail.com]:587

smtp_sasl_auth_enable = yes

smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd

smtp_sasl_security_options = noanonymous

smtp_tls_security_level = encrypt

smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt

- Set up SASL authentication:

#sudo nano /etc/postfix/sasl_passwd

#[smtp.gmail.com]:587 your-email@gmail.com:your-app-password

- Update password map and restart:

#sudo postmap /etc/postfix/sasl_passwd

#sudo systemctl restart postfix

3. Resource Monitoring Script

- Create monitoring script:

#nano resource_monitor.sh

#!/bin/bash

THRESHOLD=90

EMAIL="your-email@gmail.com"

LOG_FILE="/var/log/resource_monitor.log"

while true; do

Get CPU usage

CPU=\$(top -bn1 | grep "Cpu(s)" | awk '{print \$2}')

Get memory usage

*MEM=\$(free | grep Mem | awk '{print \$3/\$2 * 100.0}')*

Get disk usage

DISK=\$(df / | grep / | awk '{print \$5}' | sed 's/%//')

Check thresholds and send alerts

if ((\$(echo "\$CPU > \$THRESHOLD" | bc -l))); then

echo "CPU usage is above \$THRESHOLD% (\$CPU%)" | mail -s "CPU Alert" \$EMAIL

fi

```

if (( $(echo "$MEM > $THRESHOLD" | bc -l) )); then
echo "Memory usage is above $THRESHOLD% ($MEM%)" | mail -s "Memory Alert" $EMAIL
fi

if (( DISK > THRESHOLD )); then
echo "Disk usage is above $THRESHOLD% ($DISK%)" | mail -s "Disk Alert" $EMAIL
fi

sleep 60

done

```

- Set permissions:

```
#chmod +x resource_monitor.sh
```

4. Testing Procedures

1. CPU Testing:

- Start CPU stress test:

```
#yes > /dev/null &
```

```
#yes > /dev/null &
```

```
#yes > /dev/null &
```

- Stop CPU test:

```
#pkill yes
```

2. Memory Testing:

- Start memory stress:

```
#dd if=/dev/zero of=/dev/null bs=1M count=5000 &
```

- Stop memory test:

```
#kill $(pgrep dd)
```

3. Disk Testing:

- Start disk stress:

```
#dd if=/dev/zero of=/tmp/testfile bs=1M count=500
```

- Clean up:

```
#rm /tmp/testfile
```