Secure Mail Server with YARA and Sigma Rules: Implementation Guide

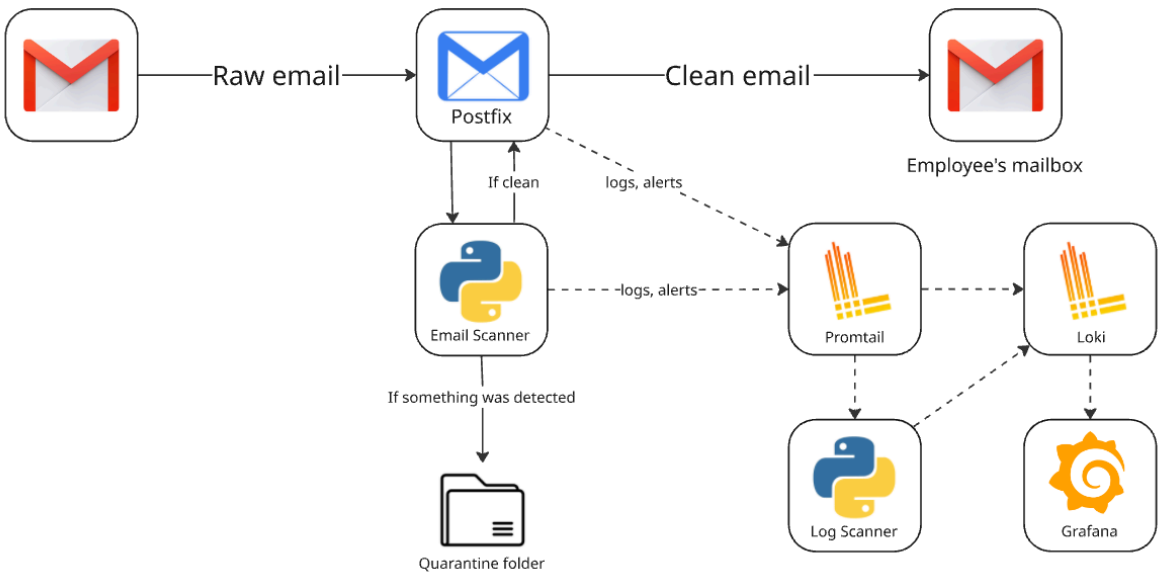
Secure Mail Server with YARA and Sigma Rules: Implementation Guide

This guide provides a comprehensive overview of setting up and using the containerized email threat detection system that leverages YARA and Sigma rules for enhanced security.

System Overview

Your project implements a robust email security solution with these key components:

1. **Postfix** - Lightweight mail transfer agent that receives and routes emails
2. **Python Email Scanner** - Applies YARA rules to email content and attachments
3. **Log Scanner** - Applies Sigma rules to analyze logs for suspicious patterns
4. **Promtail** - Collects and forwards logs to Loki
5. **Loki** - Centralized log storage system
6. **Grafana** - Visualization dashboard for monitoring alerts and logs



Installation Instructions

Prerequisites

- Git

- Docker
- Docker Compose
- Python 3.x
- pip

Setup Steps

1. Clone the repository:

```
git clone git@github.com:CarlosNadal/holbertonshcool-cybersecurity-final-proyect.git
cd holbertonshcool-cybersecurity-final-proyect
```

2. Build and start the containers:

```
docker compose up --build
```

3. Verify all containers are running:

```
docker ps
```

4. Access Grafana dashboard at `http://localhost:3000` (default credentials: admin/changeme)

Troubleshooting

Issue: Postfix container exits with "no such file or directory" error
Solution: Convert line endings in entrypoint.sh:

- On Linux: `dos2unix postfix/entrypoint.sh`
- On Windows (VS Code): Change line endings from CRLF to LF

Usage Guide

Testing the System

1. Send test emails with various payloads:

```
python send_test_mail.py
```

2. Test individual YARA rules:

```
apt install yara
yara rules/yara/your_rule.yar file_to_scan.txt
```

Monitoring

1. **Quarantined emails**: Located in the designated quarantine folder (path specified in scanner configuration)
2. **Logs**: Viewable in Grafana or directly in Loki

3. **Grafana dashboards**: Pre-configured to show:
    - Email scanning statistics
    - Rule matches (both YARA and Sigma)
    - System alerts
    - Traffic patterns

Rule Development

Sigma Rules

1. Create rules in `/rules/sigma` as `.yml` files
2. Example rule structure:

```
title: Suspicious Email Subject
id: abc12345-6789-0def-1234-567890abcdef
description: Detects emails with suspicious subjects
logsource:
  category: email
detection:
  selection:
    subject|contains:
      - "urgent"
      - "invoice"
      - "click here"
  condition: selection
level: medium
```

3. Validate rules:

```
pip install sigmatools
sigma validate rules/sigma/your_rule.yml
# Or validate all rules:
sigma validate rules/sigma/
```

YARA Rules

1. Create rules in `/rules/yara` as `.yar` or `.yara` files
2. Example rule structure:

```
rule Suspicious_Executable
{
    meta:
        description = "Detects a suspicious PE executable"
        author = "Your Name"
        date = "2025-05-14"
```

```
    strings:
        $mz = "MZ"
        $suspicious_string = "cmd.exe"
    condition:
        $mz at 0 and $suspicious_string
}
```

Limitations and Future Enhancements

1. **External Email Server Integration**:
   - Currently limited to localhost testing
   - Future: Configure MX records, TLS, and spam protection for external email processing
2. **Third-Party Notifications**:
   - Future: Integrate Prometheus Alertmanager for Slack/Discord notifications
   - Customizable alert routing and escalation policies
3. **Data Retention**:
   - Future: Implement automatic deletion of logs/quarantined emails
   - Configurable retention periods via environment variables
4. **Additional Enhancements**:
   - Expand YARA rule coverage for emerging threats
   - Add more Sigma rules for comprehensive log analysis
   - Implement machine learning for anomaly detection
   - Add user management for quarantine review/release

Security Considerations

1. Regularly update YARA and Sigma rules to detect new threats
2. Monitor system performance to ensure email delivery isn't delayed
3. Review quarantined emails periodically for false positives
4. Secure Grafana and other management interfaces
5. Consider implementing TLS for all internal communications

This system provides a solid foundation for email security that can be expanded as needs grow. The containerized approach makes it portable and easy to maintain, while the use of standard tools like YARA and Sigma ensures compatibility with broader security ecosystems.