

Task:

“Configuring Firewalls and Intrusion Detection Systems”.

Provided by:

“Digital Empowerment Networks”.

Done by:

“Khawar Amin”.

Objective:

Protect the network by setting up firewalls and IDS.

Description:

Implement firewalls and intrusion detection systems to monitor and control incoming and outgoing network traffic. Detect and prevent unauthorized access and attacks.

Selecting appropriate firewall and IDS solutions.

Firewall: ufw (Uncomplicated Firewall).

IDS: suricata.

Configuring firewall rules and policies.

Firewall Rules:

- Only accept necessary traffic and deny other traffic that is not necessary.
- Allow and Deny Traffic on specific port depending on your needs i.e. HTTP/HTTPS outgoing traffic on Port 80 and 443 if you had set up your webserver or if you wanted to access you must accept incoming HTTP/HTTPS traffic. Also, you can set up rules to accept or deny traffic from specific sources or destination ports.
- Allow and Deny Traffic from specific IP addresses. You can also set rules to accept only traffic from specific IP on specific port.
- Limit incoming connection requests to save yourself for DOS Attack or to limit number of SSH login attempts.
- Allow or Deny traffic for VPN connections to allow remote access securely.
- Allow or Deny traffic from specific services like FTP, SSH, IMAP.
- Enable logging to monitor and analyze blocked or allowed traffic.
- You can make more rules on your specific needs.

Firewall Policies:

Allow Established and Related Connections:

- **Policy:** Deny all traffic by default unless explicitly allowed.

- **Purpose:** Minimizes exposure by ensuring that only specified traffic is permitted.
- **Implementation:** Set the default incoming policy to deny.

Allow Established and Related Connections:

- **Policy:** Allow traffic related to established connections.
- **Purpose:** Ensures that ongoing sessions can continue without interruption.
- **Implementation:** Typically managed automatically by many firewalls but can be explicitly configured if needed.

Least Privilege Access:

- **Policy:** Allow only the minimum required access for services and users.
- **Purpose:** Reduces the attack surface by limiting exposure to only necessary services.
- **Implementation:** Define rules for specific services and restrict access to known, trusted IP addresses.

IP Whitelisting:

- **Policy:** Allow traffic only from specific, trusted IP addresses.
- **Purpose:** Restricts access to services from known sources.
- **Implementation:** Configure rules to allow traffic from specified IP addresses only.

Network Segmentation:

- **Policy:** Create different network segments and control traffic between them.
- **Purpose:** Limits the impact of a security breach by isolating different parts of the network.
- **Implementation:** Define rules to control traffic between network segments.

Rate Limiting:

- **Policy:** Limit the rate of incoming connections to prevent abuse.

- **Purpose:** Mitigates denial-of-service (DoS) attacks and reduces load on services.
- **Implementation:** Use rate-limiting features to restrict connection attempts.

Logging and Monitoring:

- **Policy:** Enable logging of firewall activity and monitor logs regularly.
- **Purpose:** Provides visibility into traffic patterns and helps detect unauthorized access.
- **Implementation:** Enable and review firewall logs.

Application-Level Rules:

- **Policy:** Create rules based on specific applications rather than just ports.
- **Purpose:** Allows more granular control over traffic based on application requirements.
- **Implementation:** Define rules for services by name or application

Secure Management Access:

- **Policy:** Restrict access to firewall management interfaces.
- **Purpose:** Protects the firewall from unauthorized changes and access.
- **Implementation:** Limit management access to specific IP addresses and use strong authentication.

Policy for External and Internal Traffic:

- **Policy:** Differentiate rules for internal (within the organization) and external (internet) traffic.
- **Purpose:** Ensures appropriate controls for different types of network traffic.
- **Implementation:** Set specific rules for traffic based on its origin.

Regular Rule Reviews and Updates:

- **Policy:** Regularly review and update firewall rules and policies.
- **Purpose:** Adapts to changing network environments and emerging threats.

- **Implementation:** Schedule periodic reviews and adjust rules as necessary.

Backup and Recovery:

- **Policy:** Regularly back up firewall configurations and have a recovery plan in place.
- **Purpose:** Ensures quick restoration of configurations in case of failure or misconfiguration.
- **Implementation:** Save backup copies of firewall configurations and test recovery procedures.

Practices:

- Review and update Firewall rules daily.
- Regularly Backup your firewall configuration to avoid data loss and Facilitate recovery. Document firewall rules and policies.
- Integrate it with IDS to detect and remove malicious files or traffic that bypass firewall rules.

Linux commands that I run in my terminal for setting up Rules of Firewall:

```
└─(root@Lt-GH0ST)-[~]
```

```
└─# ufw status numbered
```

Status: active

To	Action	From
--	-----	----
[1] Anywhere	DENY IN	192.168.100.6
[2] 22 (v6)	DENY IN	Anywhere (v6)

```
└─(root@Lt-GH0ST)-[~]
```

```
└─# ufw allow smtp
```

Rule added

Rule added (v6)

└─(root⊕Lt-GH0ST)-[~]

└─# ufw allow imap

Rule added

Rule added (v6)

└─(root⊕Lt-GH0ST)-[~]

└─# ufw allow 80/tcp

Rule added

Rule added (v6)

└─(root⊕Lt-GH0ST)-[~]

└─# ufw limit 22/tcp

Rule added

Rule added (v6)

└─(root⊕Lt-GH0ST)-[~]

└─# ufw deny from 192.168.100.6

Skipping adding existing rule

└─(root⊕Lt-GH0ST)-[~]

└─# ufw status numbered

Status: active

To	Action	From
--	-----	----
[1] Anywhere	DENY IN	192.168.100.6
[2] 25/tcp	ALLOW IN	Anywhere
[3] 143/tcp	ALLOW IN	Anywhere
[4] 80/tcp	ALLOW IN	Anywhere
[5] 22/tcp	LIMIT IN	Anywhere
[6] 22 (v6)	DENY IN	Anywhere (v6)
[7] 25/tcp (v6)	ALLOW IN	Anywhere (v6)
[8] 143/tcp (v6)	ALLOW IN	Anywhere (v6)
[9] 80/tcp (v6)	ALLOW IN	Anywhere (v6)

[10] 22/tcp (v6) LIMIT IN Anywhere (v6)

```
root@Lt-GH0ST: ~  
File Actions Edit View Help  
# ufw status numbered  
Status: active  


| To            | Action  | From          |
|---------------|---------|---------------|
| [ 1] Anywhere | DENY IN | 192.168.100.6 |
| [ 2] 22 (v6)  | DENY IN | Anywhere (v6) |

  
# ufw allow smtp  
Rule added  
Rule added (v6)  
  
# ufw allow imap  
Rule added  
Rule added (v6)  
  
# ufw allow 80/tcp  
Rule added  
Rule added (v6)  
  
# ufw deny from 192.168.100.6  
Skipping adding existing rule  
  
# ufw status numbered  
Status: active  


| To                | Action   | From          |
|-------------------|----------|---------------|
| [ 1] Anywhere     | DENY IN  | 192.168.100.6 |
| [ 2] 25/tcp       | ALLOW IN | Anywhere      |
| [ 3] 143/tcp      | ALLOW IN | Anywhere      |
| [ 4] 80/tcp       | ALLOW IN | Anywhere      |
| [ 5] 22/tcp       | LIMIT IN | Anywhere      |
| [ 6] 22 (v6)      | DENY IN  | Anywhere (v6) |
| [ 7] 25/tcp (v6)  | ALLOW IN | Anywhere (v6) |
| [ 8] 143/tcp (v6) | ALLOW IN | Anywhere (v6) |
| [ 9] 80/tcp (v6)  | ALLOW IN | Anywhere (v6) |
| [10] 22/tcp (v6)  | LIMIT IN | Anywhere (v6) |


```

Setting up IDS to monitor network traffic:

I Successfully Downloaded suricata.

Using: sudo apt install suricata.

Let's configure it.

```
root@Lt-GHOST: /home/lt-gh0st/Downloads/suricata-7.0.6/etc
File Actions Edit View Help
GNU nano 8.1 /etc/suricata/suricata.yaml
%YAML 1.1
4. Configure Suricata

# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://docs.suricata.io/en/latest/configuration/suricata.yaml.html

# This configuration file generated by Suricata 7.0.6
suricata-version: "7.0.6"

##
## Step 1: Inform Suricata about your network
##

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.100.0/24]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
    #EXTERNAL_NET: "any"

    HTTP_SERVERS: "$HOME_NET"
    SMTP_SERVERS: "$HOME_NET"
    SQL_SERVERS: "$HOME_NET"
    DNS_SERVERS: "$HOME_NET"
    TELNET_SERVERS: "$HOME_NET"
    AIM_SERVERS: "$EXTERNAL_NET"
    DC_SERVERS: "$HOME_NET"
```

```
(root@Lt-GHOST)-[/home/lt-gh0st/Downloads/suricata-7.0.6/etc]
# suricata --version
suricata: unrecognized option '--version'
Suricata 7.0.6
USAGE: suricata [OPTIONS] [BPF FILTER]

-c <path> : path to configuration file
-T : test configuration file (use with -c)
-i <dev or ip> : run in pcap live mode
-F <bpffilter file> : bpf filter file
-r <path> : run in pcap file/offline mode
-s <path> : path to signature file loaded in addition to suricata.yaml settings (optional)
-S <path> : path to signature file loaded exclusively (optional)
-l <dir> : default log directory
-D : run as daemon
-k [all|none] : force checksum check (all) or disabled (none)
-V : display Suricata version
-v : be more verbose (use multiple times to increase verbosity)
--list-app-layer-protos : list supported app layer protocols
--list-keywords[=-all|csv|<keyword>] : list keywords implemented by the engine
--list-runmodes : list supported runmodes
--runmode <runmode_id> : specific runmode modification the engine should run. The argument
: supplied should be the ID for the runmode obtained by running
: --list-runmodes
--engine-analysis : print reports on analysis of different sections in the engine and exit.
: Please have a look at the conf parameter engine-analysis on what reports
: can be pulled
--pidfile <file> : write pid to this file
--init-errors-fatal : enable fatal failure on signature init error
--disable-detection : disable detection engine
--dump-config : show the running configuration
--dump-features : display provided features
--build-info : display build information
--pcap[=<dev>] : run in pcap mode, no value select interfaces from suricata.yaml
--pcap-file-continuous : when running in pcap mode with a directory, continue checking directory for pcaps
```

Suricata is running and enabled as we can see in the screenshot below.


```
root@Lt-GH0ST: /home/lt-gh0st/Downloads
File Actions Edit View Help
31/8/2024 -- 17:48:02 -- <Info> -- Testing with suricata -T.
31/8/2024 -- 17:48:32 -- <Info> -- Done.

(root@Lt-GH0ST)-[/home/lt-gh0st/Downloads]
# sudo systemctl start suricata

(root@Lt-GH0ST)-[/home/lt-gh0st/Downloads]
# sudo systemctl enable suricata

Synchronizing state of suricata.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable suricata

(root@Lt-GH0ST)-[/home/lt-gh0st/Downloads]
# sudo systemctl status suricata

● suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/usr/lib/systemd/system/suricata.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-08-31 16:31:45 PKT; 1h 17min ago
 Invocation: b11adb969f384d60a057b77623a36ad1
    Docs: man:suricata(8)
          man:suricatasc(8)
          https://suricata.io/documentation/
 Main PID: 183907 (Suricata-Main)
    Tasks: 14 (limit: 18966)
  Memory: 86.9M (peak: 87.2M)
     CPU: 47.956s
   CGroup: /system.slice/suricata.service
           └─183907 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid

Aug 31 16:31:45 Lt-GH0ST systemd[1]: Starting suricata.service - Suricata IDS/IDP daemon...
Aug 31 16:31:45 Lt-GH0ST suricata[183898]: i: suricata: This is Suricata version 7.0.6 RELEASE running in SYSTEM mode
Aug 31 16:31:45 Lt-GH0ST systemd[1]: Started suricata.service - Suricata IDS/IDP daemon.
```

To See alerts in Suricata:

Use this command to open Suricata configuration and set up alert's methods.

```
# Configure the type of alert (and other) logging you would like
outputs:
  # a line based alerts log similar to Snort's fast.log
  - fast:
    enabled: yes
    filename: fast.log
    append: yes
    filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'

  # Extensible Event Format (nicknamed EVE) event log in JSON format
  - eve-log:
    enabled: yes
    filetype: regular #regular/syslog/unix_dgram/unix_stream/redis
    filename: eve.json
    # Enable for multi-threaded eve.json output; output files are amended with
    # an identifier, e.g., eve.9.json
    #threaded: false
    #prefix: "@cee: " # prefix to prepend to each log entry
    # the following are valid when type: syslog above
    #identity: "suricata"
    #facility: local5
    #level: Info ## possible levels: Emergency, Alert, Critical
    ## Error, Warning, Notice, Info, Debug
    #ethernet: no # log ethernet header in events when available
    #redis:
    #  server: 127.0.0.1
    #  port: 6379
    #  async: true ## if redis replies are read asynchronously
```

My file type is eve.json .

Use tail, less or cat command to see its content like

`sudo tail -f /var/log/suricata/eve.json`

```
(root@Lt-GHOST)-[/home/lt-ghost/Downloads/suricata-7.0.6/etc]
# cat eve.json | jq .

[
  {
    "timestamp": "2024-08-31T19:12:34.567000+0000",
    "event_type": "alert",
    "src_ip": "192.168.1.1",
    "src_port": 12345,
    "dest_ip": "192.168.100.8",
    "dest_port": 80,
    "proto": "TCP",
    "alert": {
      "signature": "Fake alert - TEST",
      "signature_id": 1000001,
      "category": "Potentially Bad Traffic",
      "severity": 1
    }
  },
  {
    "timestamp": "2024-08-31T19:13:45.678000+0000",
    "event_type": "alert",
    "src_ip": "192.168.1.2",
    "src_port": 54321,
    "dest_ip": "192.168.100.8",
    "dest_port": 443,
    "proto": "TCP",
    "alert": {
      "signature": "Another fake alert - TEST",
      "signature_id": 1000002,
      "category": "Suspicious Activity",
      "severity": 2
    }
  }
]
```

Best practices for IDS:

Choose the Right IDS Type:

- **Network-Based IDS (NIDS):** Monitors network traffic for suspicious activity.
- **Host-Based IDS (HIDS):** Monitors and analyzes the internals of a computing system.

Proper Installation and Configuration:

- **Placement:** Install IDS at strategic points in your network, such as the perimeter or key segments.
- **Configuration:** Tailor the IDS rules and settings to match your network environment and security policies.

Regular Updates:

- **Signature Updates:** Ensure that IDS signature databases are regularly updated to recognize the latest threats.
- **Software Updates:** Keep the IDS software itself up-to-date to benefit from new features, improvements, and security patches.

Fine-Tune Rules and Signatures:

- **Custom Rules:** Create and refine custom rules to minimize false positives and enhance detection capabilities.
- **Baseline Normal Activity:** Understand normal network and system behavior to adjust IDS rules and thresholds accordingly.

Integration with Other Security Tools:

- **SIEM Integration:** Integrate IDS with Security Information and Event Management (SIEM) systems for centralized logging, analysis, and correlation of security events.
- **Firewall and IPS:** Coordinate IDS with firewalls and Intrusion Prevention Systems (IPS) for a layered security approach.

Monitoring and Analysis:

- **Real-Time Monitoring:** Continuously monitor alerts and logs to quickly detect and respond to potential threats.
- **Incident Analysis:** Regularly review and analyze detected incidents to improve detection rules and response strategies.

Performance Considerations:

- **Resource Management:** Ensure the IDS has adequate resources (CPU, memory) to handle the volume of traffic and data it needs to process.
- **Network Impact:** Configure the IDS to minimize its impact on network performance.

Documentation and Reporting:

- **Alert Documentation:** Document all alerts, incidents, and responses for future reference and analysis.
- **Regular Reports:** Generate and review regular reports on IDS activity, trends, and incident statistics.

Training and Awareness:

- **Staff Training:** Train IT and security staff on how to use the IDS effectively, interpret alerts, and respond to incidents.
- **Awareness Programs:** Educate users and stakeholders about security best practices and the role of the IDS.

Testing and Validation:

- **Regular Testing:** Periodically test and validate the IDS configuration and rules to ensure they are effective and not causing unintended issues.
- **Simulation Exercises:** Conduct simulation exercises to evaluate the IDS's response to different attack scenarios.

Scalability and Flexibility:

- **Future Growth:** Ensure the IDS can scale with network growth and adapt to new types of threats.
- **Flexible Configuration:** Use a modular approach to easily update and adjust IDS configurations as needed.