Below is a **step-by-step** guide with example configuration files to achieve the following goals:

1. **Collect all Windows Event logs from Windows2 and Windows3** (both Windows 11).
2. **Send these logs to a Logstash instance on Windows-LOGSTASH** and store them locally in a rolling/rotating file (up to 20GB).
3. **Use Filebeat on Windows-LOGSTASH** to forward the locally stored logs to the Wazuh stack on linux-WAZUH (Ubuntu 24.04) for indexing and visualization.
4. **Ensure resilience** so that if Windows-LOGSTASH is disconnected from linux-WAZUH, logs will be cached locally and forwarded once connectivity is restored.
5. **Distinguish** logs from Windows2 and Windows3 in Wazuh.
6. **Do not install the Wazuh agent** on Windows2 and Windows3 (use Winlogbeat or another forwarder instead).

---

# Overall Architecture

pgsql
Copy

```
+-------------------+      +--------------------+      +-------------------
--+
|  Windows2 (Win11) |      |                    |      |  linux-WAZUH
(Ubuntu|
|  - Winlogbeat     | ---> |  Windows-LOGSTASH  | ---> | 24.04)
|
|  - Forwards logs  |      |  - Logstash        |      | - Wazuh indexer
|
|                   |      |  - Rolling local   |      | - Wazuh server
|
+-------------------+      |    storage (20GB)  |      | - Wazuh dashboard
|
                           |  - Filebeat -->    |      +-------------------
--+
+-------------------+      |    forward to      |
|  Windows3 (Win11) |      |    Wazuh           |
|  - Winlogbeat     | ---> +--------------------+
|  - Forwards logs  |
+-------------------+
```

- **Windows2 & Windows3** each run Winlogbeat, sending logs to `Windows-LOGSTASH` on port 5044.
- **Windows-LOGSTASH** uses **Logstash** to receive logs, store them in a rolling file (max 20GB), and also runs **Filebeat** to ship those logs to the Wazuh cluster on `linux-WAZUH`.
- If the network link between `Windows-LOGSTASH` and `linux-WAZUH` goes down, Filebeat on `Windows-LOGSTASH` will queue/hold logs until connectivity is restored.
- **No Wazuh agent** is deployed on the Windows endpoints.

---

# 1. Install and Configure Winlogbeat on Windows2 and Windows3

Use the official Winlogbeat (from Elastic) on each Windows machine. Below is an example Winlogbeat configuration for Windows2. The same applies to Windows3 with slight modifications (especially the `name` or additional fields).

## 1.1 Download and Install Winlogbeat

1. Download Winlogbeat (matching your Logstash/Filebeat/Elasticsearch major version, e.g., 7.x or 8.x) from Elastic's downloads page.
2. Extract/unzip and place it in a suitable directory (e.g., `C:\Program Files\Winlogbeat`).

## 1.2 Example `winlogbeat.yml` for Windows2

```
yaml
Copy
# winlogbeat.yml

winlogbeat.event_logs:
  - name: Application
  - name: System
  - name: Security

# You can add other channels if needed.
# e.g.:
#   - name: Microsoft-Windows-PowerShell/Operational
#   - name: ...

# Configure output to Logstash
output.logstash:
  hosts: ["WINDOWS-LOGSTASH:5044"]
  # If you have DNS issues, you can use IP, e.g. ["192.168.1.10:5044"]

# (Optional) Include some metadata to distinguish logs
name: "Windows2"
fields:
  os: "Windows11"
  source_host: "Windows2"

fields_under_root: true

# Set logging level (optional)
logging.level: info
logging.to_files: true
logging.files:
  path: C:/ProgramData/winlogbeat/logs
```

**Notes:**

- Change `WINDOWS-LOGSTASH` to the actual hostname (or IP) of your Logstash server.
- For Windows3, simply change `name: "Windows3"` and `source_host: "Windows3"`.

### 1.3 Install Winlogbeat as a Service

Open PowerShell or Command Prompt (as Administrator), navigate to the Winlogbeat directory, and run:

```
powershell
Copy
winlogbeat.exe install
winlogbeat.exe start
```

This installs Winlogbeat as a Windows service and starts it automatically.

---

## 2. Configure Logstash on Windows-LOGSTASH

Logstash needs to listen for incoming beats (Winlogbeat) on port 5044, store logs locally in a **rotating file** up to 20GB, and also make them available for Filebeat to pick up.

### 2.1 Install Logstash

1. Download and install the same major version of Logstash from Elastic's downloads page.
2. Once installed, locate your Logstash configuration directory (commonly `C:\logstash\config` or `C:\Program Files\Logstash\config`).

### 2.2 Add the Rolled File Plugin (if needed)

By default, the built-in `file` output in Logstash **does not** rotate by size. You have a few options to achieve a "cyclic" max-size approach:

1. Use an external solution (e.g., [logrotate via scheduled tasks]) to rotate the file once it hits 20GB.
2. Use a community or third-party plugin like logstash-output-rolled_file.

Below is an example using the **rolled_file** plugin:

```
powershell
Copy
# From PowerShell, install the plugin if needed:
logstash-plugin install logstash-output-rolled_file
```

### 2.3 Create Logstash Pipeline Configuration

Create a file, e.g. `C:\logstash\config\pipelines\winbeats.conf`:

```
conf
Copy
# winbeats.conf

input {
  beats {
    port => 5044
    # If using SSL, configure ssl_certificate/ssl_key here
  }
}

filter {
  # Optional filters if you want to parse or manipulate fields
  # Example: add tags or rename fields
  mutate {
    add_tag => ["from_winlogbeat"]
  }
}

output {
  # 1) Output to a rolling file up to ~20GB
  rolled_file {
    path => "C:/logstash-data/logstash-output"
    # The actual filename pattern
    filename_pattern => "logstash-%{+yyyyMMdd_HH}.log"
    # This is the size limit in bytes (20GB)
    file_size_limit => 21474836480
    # The number of rotated files to keep before overwriting the oldest
    # If you only want a single file up to 20GB, then set:
    max_roll_count => 1
    # If you allow more than 1, you can keep multiple older files
    # e.g. max_roll_count => 5
    # Then you get up to 5 x 20GB = 100GB max
  }

  # 2) (Optional) If you want to do something else with logs
  #    or debug them before Filebeat picks them up, you can
  #    add more outputs here.
}
```

**Important**: The above configuration means **Logstash** will continuously append to a file named with a pattern like `logstash-20250219_15.log` until it reaches 20GB, then roll it. If `max_roll_count => 1`, it will overwrite the old file once the new file also hits 20GB. This effectively caps your storage usage.

You must ensure that the directory `C:/logstash-data` exists (or whichever path you use) and that Logstash has permission to write there.

## 2.4 Run Logstash

To run Logstash (PowerShell, as admin):

```
powershell
Copy
cd "C:\Program Files\Logstash\bin"
.\logstash.bat -f "C:\logstash\config\pipelines\winbeats.conf"
```

Or set it up as a Windows service, depending on your preference.

---

# 3. Configure Filebeat on Windows-LOGSTASH to Forward to Wazuh

Next, **Filebeat** will read the rolled/rotating log files created by Logstash and forward them to your Wazuh cluster on `linux-WAZUH`. Because we want resilience, Filebeat will keep track of what has already been sent. If the link to Wazuh is lost, Filebeat will buffer and catch up when the connection is back.

## 3.1 Install Filebeat

On `Windows-LOGSTASH`, install Filebeat similarly (matching the major version of your Wazuh/Elastic components).

## 3.2 Example `filebeat.yml`

Create or modify `C:\Program Files\Filebeat\filebeat.yml`:

```
yaml
Copy
filebeat.inputs:
  - type: log
    enabled: true
    # Point to the rolling files from Logstash
    paths:
      - "C:/logstash-data/*.log"
    # If logs can have multiline messages, you might configure multiline
here
    # multiline:
    #   pattern: '^\['
    #   negate: true
    #   match: after

    # Additional fields to help identify
    fields:
      source: "logstash-rolled"
      environment: "production"
    fields_under_root: true

# Output to Wazuh/Elasticsearch
output.elasticsearch:
  # Replace with the actual IP or domain of your Wazuh cluster’s
Elasticsearch
  # Typically, if Wazuh is using an embedded Elasticsearch, it might be
  # accessible via port 9200 on the same server or a load balancer.
  hosts: ["linux-WAZUH:9200"]
  # If needed, specify username/password or certificates

  # This ensures Filebeat keeps data until it can deliver them
  # (default is usually 10-15 seconds)
  bulk_max_size: 50
  worker: 1

# Enable/adjust logging if desired
logging.level: info
logging.to_files: true
logging.files:
  path: C:/ProgramData/filebeat/logs
```

**Notes:**

- If your Wazuh cluster does not run on port `9200`, adjust accordingly.
- If you have TLS enabled on Wazuh, you must configure the SSL settings here (e.g., `ssl.certificate_authorities`).
- If you have a separate index or pipeline for Wazuh, you may need to specify them (e.g., `setup.template.name`, `setup.template.pattern`, `index`, etc.).

### 3.3 Start Filebeat

As with Winlogbeat:

```
powershell
Copy
filebeat.exe install
filebeat.exe start
```

This will launch Filebeat as a service, tail the rolling log files from Logstash, and forward them to `linux-WAZUH`.

---

# 4. Verify in Wazuh

- On `linux-WAZUH`, access your Wazuh Dashboard (Kibana, or the dedicated Wazuh UI).
- You should see incoming logs from the Windows machines.
- The logs should contain fields like `host.name` (from Winlogbeat) or the custom fields you added, allowing you to distinguish Windows2 vs. Windows3.
- If the link from `Windows-LOGSTASH` to `linux-WAZUH` goes down, check the Filebeat logs on `Windows-LOGSTASH` to confirm it is retrying. Once connectivity is restored, logs are forwarded automatically.

---

# 5. Resilience and Catch-Up

1. **Winlogbeat** on Windows2/Windows3 will automatically queue logs briefly if it can't reach `Windows-LOGSTASH`. Once connectivity is restored, logs continue.
2. **Logstash** writes all logs to local rolling files up to 20GB. You'll have a local repository of logs on `Windows-LOGSTASH`.
3. **Filebeat** on `Windows-LOGSTASH` will read from those files and keep an internal registry of what has been sent to Wazuh. If the link to Wazuh is down, logs remain in local files. Once Wazuh is reachable again, Filebeat continues reading from where it left off.

Because of the 20GB rolling limit, older logs will be overwritten after you hit the file size limit (depending on how quickly you generate logs). This satisfies your “cyclic log” requirement. If you want a different approach (like time-based rotation plus size-based retention), you can tune the plugin or use an external tool (logrotate for Windows, or custom scripts).

---

# 6. Distinguishing Windows2 and Windows3 in Wazuh

You have several ways to differentiate logs:

- **Via `Winlogbeat` name**:

  ```
  yaml
  Copy
  name: "Windows2"
  fields:
    source_host: "Windows2"
    os: "Windows11"
  ```

- **By default**, Winlogbeat also sends the hostname in `host.name` or `agent.name`.
- **Logstash** can also add or mutate fields/tags, but typically you don’t need to if Winlogbeat is already sending them.
- **Filebeat** can add further fields or tags in its input config.

When you view these logs in the Wazuh/Kibana interface, you can filter by `host.name: Windows2` or whichever fields you set.

---

# 7. Summary of Key Files

1. **Windows2 / Windows3**: `C:\Program Files\Winlogbeat\winlogbeat.yml`
2. **Windows-LOGSTASH**:
   - **Logstash pipeline**: `C:\logstash\config\pipelines\winbeats.conf`
   - **Logstash rolled file output** stored in: `C:\logstash-data\` (or your chosen path)
   - **Filebeat config**: `C:\Program Files\Filebeat\filebeat.yml`

---

# 8. Tips and Considerations

1. **Firewall**: Ensure port 5044 is open on Windows-LOGSTASH for inbound from Windows2/Windows3. Ensure any relevant ports (e.g., 9200) are open from Windows-LOGSTASH to linux-WAZUH.
2. **SSL/TLS**: In production, it is strongly recommended to encrypt traffic (using SSL certificates) especially from Winlogbeat to Logstash and from Filebeat to Wazuh.

3. **Monitoring**: Keep an eye on disk usage on Windows-LOGSTASH to make sure you aren't inadvertently running out of space, especially if logs come in fast and you have large bursts.
4. **Index Management in Wazuh**: You may want to create specific indices or index patterns for Windows logs in Wazuh. Alternatively, you can rely on the default index.

---

# Complete End-to-End Flow

1. **Winlogbeat** (Windows2, Windows3) → port 5044 → **Logstash** (Windows-LOGSTASH).
2. **Logstash** writes logs to rolling files up to 20GB.
3. **Filebeat** monitors those rolling files → sends to **Wazuh** (linux-WAZUH).
4. Wazuh/Elasticsearch indexes the logs, Wazuh Dashboard visualizes them.

You now have a reliable pipeline with local caching/rolling, no Wazuh agent on the Windows endpoints, and clear distinction between Windows2 and Windows3 logs in Wazuh.

---

Anything ELSE

Here are **some additional tips** and **best practices** you might consider. These aren't strictly required, but can help improve reliability, security, and manageability of your logging pipeline:

---

### 1. Use TLS/SSL for Encrypted Communication

- **Between Winlogbeat → Logstash**:
  - Generate or obtain TLS certificates (self-signed or from an internal CA).
  - In `winlogbeat.yml`, enable SSL by adding:

```
yaml
Copy
output.logstash:
  hosts: ["WINDOWS-LOGSTASH:5044"]
  ssl.certificate_authorities: ["C:/Program 
Files/Winlogbeat/ca.crt"]
  # optionally ssl.certificate, ssl.key if mutual auth is 
required
```

  - In the Logstash `beats` input configuration:

```
conf
Copy
input {
  beats {
    port => 5044
    ssl => true
    ssl_certificate => "C:/logstash/config/certs/logstash.crt"
```

```
        ssl_key => "C:/logstash/config/certs/logstash.key"
      }
    }
```

- **Between Filebeat → Wazuh/Elasticsearch**:
  - If Wazuh/Elasticsearch is secured with TLS on port 9200, include:

    ```
    yaml
    Copy
    output.elasticsearch:
      hosts: ["linux-WAZUH:9200"]
      ssl.certificate_authorities: ["/path/to/ca.crt"]
      # If using basic auth or tokens, provide credentials
    ```

- Encryption helps ensure logs in transit aren't exposed to eavesdropping or tampering.

---

## 2. Consider Persistent Queues in Logstash

If you want an additional layer of **resilience** (for example, in case Logstash crashes or the system reboots), you can enable **Logstash persistent queues**. This way, if Logstash receives data from Winlogbeat but can't write to the output file (or if there is a crash), logs remain in an on-disk queue. To enable persistent queues in Logstash, add something like this in `logstash.yml`:

```
yaml
Copy
queue.type: persisted
path.queue: "C:/logstash-queue"
queue.max_bytes: 4gb
```

This ensures data is retained on disk until Logstash can write it to the output. It's especially useful in higher-volume or mission-critical environments.

---

## 3. Monitor Storage Usage

Because you're storing up to **20GB** per log file in a rolling manner, be sure to:

1. **Check free disk space** on `Windows-LOGSTASH`.
2. Consider using **time-based** plus **size-based** rotation if you want more predictable retention. For instance, rotate daily or weekly (or if 20GB is reached, whichever comes first). This approach can be managed either by:
   - A plugin like `rolled_file` with advanced options, or
   - Using an external solution like `logrotate` (on Windows, you can script a scheduled task to rename/compress/delete old files).

---

## 4. Customize Winlogbeat for Specific Event IDs or Channels

If you only need specific logs (e.g., Security events with certain IDs, or Sysmon logs, or PowerShell logs), you can refine your Winlogbeat configuration:

```
yaml
Copy
winlogbeat.event_logs:
  - name: Security
    event_id: 4624, 4625, 4634
  - name: Microsoft-Windows-Sysmon/Operational
  - name: Microsoft-Windows-PowerShell/Operational
```

This can **reduce noise** and disk usage, focusing only on the most relevant events. However, if your goal is truly *all logs*, you can leave it broad.

---

## 5. Validate the Data Flow

Once you have everything set up:

1. **From Windows2/Windows3**:
    - Check the Winlogbeat logs (`C:\ProgramData\winlogbeat\logs\winlogbeat…`) to see if it's successfully connecting and shipping events.
    - You can also force an event to appear (e.g., generate a test event in Event Viewer).
2. **On Windows-LOGSTASH**:
    - Check Logstash logs (`C:\logstash\logs` or wherever they're stored). Ensure there are no pipeline errors.
    - Confirm that new files are appearing in `C:\logstash-data`.
    - Check the Filebeat logs (`C:\ProgramData\filebeat\logs\filebeat…`) to confirm it sees the rolled files and is sending them out.
3. **On linux-WAZUH**:
    - Check Wazuh/Elasticsearch logs for indexing errors.
    - Use the Wazuh Dashboard (or Kibana) to query logs (e.g., filter by `host.name: Windows2`) to make sure they're arriving.

---

## 6. Keep Your Beats & Logstash Versions in Sync

Elastic Beats (Winlogbeat, Filebeat) and Logstash typically should be on the **same major version** to avoid potential protocol mismatches. For example, if you're using 8.x Beats, make sure Logstash is also 8.x (or at least compatible).

---

## 7. Assign Clear Host or Agent Names

In each `winlogbeat.yml`, you can explicitly set:

```
yaml
Copy
name: "Windows2"
fields:
  source_host: "Windows2"
  os: "Windows11"
fields_under_root: true
```

And in Windows3:

```
yaml
Copy
name: "Windows3"
fields:
  source_host: "Windows3"
  os: "Windows11"
fields_under_root: true
```

This makes it trivial to differentiate the logs once they land in Wazuh. You can filter in the Wazuh dashboard (or Kibana) by `source_host` or `host.name`.

---

## 8. If You Need More Advanced Parsing

If you'd like to parse or enrich the events (e.g., parse event fields, map them to ECS fields, or drop low-severity events), you could expand the **Logstash pipeline** with filters:

```
conf
Copy
filter {
  if [event_id] == "4624" {
    # Add a tag, parse further, etc.
    mutate {
      add_tag => ["windows_logon_event"]
    }
  }
}
```

Or use Logstash's Dissect/Grok filters if you want to break out fields from message strings.

---

## 9. Use a Separate Index (Optional)

If you prefer not to mix these logs with other Wazuh data or want separate retention policies, you can direct Filebeat to write to a custom index:

```
yaml
Copy
output.elasticsearch:
  hosts: ["linux-WAZUH:9200"]
  index: "windows-custom-%{+yyyy.MM.dd}"
```

Then you can manage index lifecycle policies or retention in Wazuh/Elasticsearch specifically for these logs.

---

### 10. Security Hardening

Finally, ensure all the machines involved are **firewalled** appropriately:

- Windows2/Windows3 only need to send outbound traffic to `Windows-LOGSTASH` on port 5044 (plus any ephemeral return ports).
- `Windows-LOGSTASH` needs outbound access to `linux-WAZUH` on port 9200 (or whichever port your Wazuh/Elasticsearch is on).

Lock down or secure any other open ports. Additionally, ensure Windows-LOGSTASH is patched, and consider running these services with **least-privilege** accounts.

---

# Final Thoughts

With the above setup and these additional considerations, you should have a **robust, resilient, and secure** logging pipeline. You'll have:

- **Local caching** on each Windows machine (Winlogbeat will hold events briefly if it can't connect).
- **Local storage** on Windows-LOGSTASH in rolled files (up to 20GB).
- **Automatic catch-up** with Filebeat once the network link to linux-WAZUH is restored.

This design meets your requirements for **resilience**, **distinguishing logs** between Windows2 and Windows3, **non-Wazuh-based forwarding** from Windows, and ensuring logs are not lost if connections go down.

---

Filebeat to Wazuh using certs and user/pass in more detail

Below is a **detailed walkthrough** on how to connect Filebeat (running on your Windows-LOGSTASH machine) to the **Wazuh stack** (which uses its built-in Elasticsearch) when you have a **default Wazuh installation** that automatically generates certificates and credentials.

---

# 1. Understand the Default Wazuh Setup

When you install Wazuh using the all-in-one installation wizard or the standard `wazuh-indexer`, `wazuh-server`, and `wazuh-dashboard` packages, Wazuh typically:

1. Installs and configures an **Elasticsearch**-compatible service (`wazuh-indexer`).

2. Enables **TLS** on port `9200` by default.
3. Generates a **CA certificate**, a server certificate/key, and configures them in the Wazuh Indexer.
4. Creates **default credentials** (like `elastic` user, or a specialized user for the Wazuh services).

**Note:** Depending on your Wazuh version, the default user might be called `elastic`, or it could be a user dedicated for Wazuh. In many recent releases, the default user for Elasticsearch is `elastic`. You should have been given or prompted to set a password during installation. If you’re not sure, check your Wazuh docs or the `/etc/wazuh-indexer/opensearch/.opensearch_passwords` file on the Wazuh server.

---

# 2. Locate the Wazuh CA Certificate

To enable **secure (TLS) connections** from Filebeat to Wazuh Indexer, you need to trust the **Certificate Authority (CA)** that Wazuh generated. You can typically find this CA certificate on the Wazuh server in one of these directories:

- `/etc/wazuh-indexer/certs/`
- `/etc/opensearch/certs/`
- `/etc/wazuh-dashboard/certs/`

Common file names for the CA certificate might be:

- `root-ca.pem`
- `wazuh-cafile.pem`
- `chain-ca.pem`
- Or something similar.

Check the documentation or look in those directories for a PEM file labeled something like “ca” or “root-ca.” For example:

```
swift
Copy
/etc/wazuh-indexer/certs/root-ca.pem
```

Once you find it, **copy** that file to your Windows-LOGSTASH machine, for example to:

```
makefile
Copy
C:\Program Files\Filebeat\certs\wazuh-ca.pem
```

Make sure to place it in a directory that Filebeat can read (e.g., `C:\Program Files\Filebeat\certs\`).

---

# 3. Gather Your Wazuh/Elasticsearch Credentials

During or after the default Wazuh setup, you typically have (or have set) a **username** and **password** for accessing Elasticsearch. Often the default user is `elastic`. You can verify:

1. **Check** the `/etc/wazuh-indexer/opensearch/.opensearch_passwords` file or run a command like:

   ```
   bash
   Copy
   sudo /usr/share/wazuh-indexer/opensearch-passwords interactive
   ```

   (Equivalent commands may differ based on your Wazuh version.)

2. This should display or allow you to reset the `elastic` user's password if needed.

**Example**:

- Username: `elastic`
- Password: `someSuperSecretPassword`

---

# 4. Configure Filebeat to Output to Wazuh Indexer Over TLS

On your **Windows-LOGSTASH** machine, open or create the Filebeat configuration file (usually in `C:\Program Files\Filebeat\filebeat.yml`).
Under the `output.elasticsearch:` section, configure:

1. **hosts**: Point to your Wazuh indexer's hostname or IP (port **9200**, typically).
2. **protocol**: Use `https` (or Filebeat sets it automatically if you specify `https://` in the hosts).
3. **username** and **password**: The credentials from step 3 above.
4. **ssl.certificate_authorities**: Path to the CA file you copied.

For example:

```
yaml
Copy
###################### Filebeat Configuration Example
######################

filebeat.inputs:
  - type: log
    enabled: true
    paths:
      - "C:/logstash-data/*.log"
    # ...

output.elasticsearch:
  # Use the internal IP or DNS name of your Wazuh server here:
  hosts: ["https://wazuh-server.example.local:9200"]
```

```
  # If you prefer the IP:
  # hosts: ["https://192.168.1.100:9200"]

  # Specify your credentials
  username: "elastic"
  password: "someSuperSecretPassword"

  # TLS/SSL settings
  ssl:
    certificate_authorities: ["C:/Program Files/Filebeat/certs/wazuh-
ca.pem"]
    verification_mode: "full"  # or "certificate" if you have CN mismatch
issues

# (Optional) Additional logging configurations
logging.level: info
logging.to_files: true
logging.files:
  path: "C:/ProgramData/filebeat/logs"
```

### 4.1 If You Run Into Certificate Name Mismatch

If the hostname on the Wazuh server certificate (the `CN` or `SAN` field) does not match the hostname you used in `hosts:`, Filebeat might refuse the connection with an error like `x509: certificate is not valid for the name`.

You have two main options:

1. **Use the exact hostname that matches the certificate** in the `hosts` field (ideal).
2. Lower the SSL verification level:

   ```
   yaml
   Copy
   ssl.verification_mode: "certificate"
   ```

   or even

   ```
   yaml
   Copy
   ssl.verification_mode: "none"
   ```

   (not recommended in production, but it will skip name matching).

---

## 5. Test the Connection

1. **Restart Filebeat** on Windows-LOGSTASH:

   ```
   powershell
   Copy
   Stop-Service filebeat
   Start-Service filebeat
   ```

Or run:

```
powershell
Copy
filebeat.exe test output
```

to check connectivity to Wazuh.

2. **Check the Filebeat logs** at `C:\ProgramData\filebeat\logs\filebeat` or wherever you configured logs to be written. You should see something like “Connection to backoff(elasticsearch(https://wazuh-server.example.local:9200)) established”.
3. **Verify** in the Wazuh Dashboard or Kibana interface that new logs are being indexed. If your pipeline is correct, logs from `C:/logstash-data/*.log` should appear in the Wazuh index (often named something like `wazuh-alerts-4.x-*` or, if you used a custom index, that index name).

---

## 6. Common Troubleshooting Tips

1. **Invalid Credentials**: If you see `401 Unauthorized` or `403 Forbidden` in the Filebeat logs, verify the username and password are correct.
2. **Certificate Issues**: If you see `x509 certificate signed by unknown authority`, ensure you copied the correct CA file from the Wazuh server and specified the correct path.
3. **Port Blocking**: Make sure port `9200` (TCP) is allowed in your firewall settings from Windows-LOGSTASH to Wazuh.
4. **Timeouts**: If you see connection timeouts, confirm you can ping the Wazuh server or do a quick test with:

   ```
   powershell
   Copy
   curl -k https://wazuh-server.example.local:9200
   ```

   from PowerShell (with `-k` ignoring certificate, just to test connectivity).

---

## 7. Example of a Full `filebeat.yml`

Below is a more complete sample config, showing both the **input** (tailing Logstash’s rolled files) and the **secured output** (to Wazuh’s Elasticsearch):

```
yaml
Copy
filebeat.inputs:
  - type: log
    enabled: true
    paths:
      - "C:/logstash-data/*.log"
```

```
    fields:
      source: "logstash-rolled"
      environment: "production"
    fields_under_root: true

# (Optional) If multiline needed:
# multiline:
#   pattern: '^\['
#   negate: true
#   match: after

output.elasticsearch:
  hosts: ["https://wazuh-server.example.local:9200"]
  username: "elastic"                    # or your dedicated user
  password: "someSuperSecretPassword"
  ssl.certificate_authorities: ["C:/Program Files/Filebeat/certs/wazuh-
ca.pem"]
  ssl.verification_mode: "full"

logging.level: info
logging.to_files: true
logging.files:
  path: "C:/ProgramData/filebeat/logs"
  name: filebeat
  keepfiles: 7
  rotateeverybytes: 10485760  # 10 MB per log file
```

---

# 8. Final Checklist

1. **Copy Wazuh CA** (PEM file) to Windows.
2. **Use correct credentials** (`elastic` user or equivalent) in `filebeat.yml`.
3. **Set the correct host** for `hosts` in Filebeat (with `https://...:9200`).
4. **Restart Filebeat** and confirm logs in Wazuh Dashboard.

With this, your **Filebeat** on Windows-LOGSTASH is properly secured and authenticated against the Wazuh Indexer, leveraging the **default** TLS certificates and credentials generated by the Wazuh installer.

---

What user name and password do I use that Wazuh generated

When you do a **default Wazuh installation** (using the Wazuh packages or the all-in-one installer), Wazuh typically sets up an **internal Elasticsearch-compatible service** called Wazuh Indexer (port 9200) with **TLS** enabled and default credentials.

## 1. The Default Username

In most Wazuh Indexer installations (especially recent versions), the **default user** is:

```
nginx
Copy
elastic
```

## 2. Finding or Resetting the Password

During or after the installation process, you should have been **provided** or **prompted** for the `elastic` user’s password. If you are not sure or do not recall the password, you can do either of the following on the Wazuh server:

1. **Check the .opensearch_passwords file**
   Often located at:

   ```
   swift
   Copy
   /etc/wazuh-indexer/opensearch/.opensearch_passwords
   ```

   This file (or a similarly named file) may contain the automatically generated passwords for `elastic` and any other default accounts.

2. **Run the password utility**
   Depending on your Wazuh version, you can run:

   ```
   bash
   Copy
   sudo /usr/share/wazuh-indexer/opensearch-passwords interactive
   ```

   This will let you **reset** the password for `elastic` or other users in an interactive prompt.

### 3. Using the Credentials in Filebeat

Once you know the username (`elastic`) and the password (either found or reset), **plug them** into your `filebeat.yml` under the `output.elasticsearch:` section. For example:

```
yaml
Copy
output.elasticsearch:
  hosts: ["https://YOUR_WAZUH_INDEXER_IP:9200"]
  username: "elastic"
  password: "YOUR_ELASTIC_PASSWORD"

  ssl.certificate_authorities: ["C:/Program Files/Filebeat/certs/wazuh-ca.pem"]
  ssl.verification_mode: "full"
```

- **username**: `elastic`
- **password**: the one you obtained via `.opensearch_passwords` or set via the password utility

Make sure you’ve **copied the Wazuh CA certificate** to your Windows machine (e.g., `wazuh-ca.pem`) and referenced it in the `ssl.certificate_authorities` path.

---

## Summary

- **Default user**: `elastic`

- **Password**: Provided during installation or found/reset in `/etc/wazuh-indexer/opensearch/.opensearch_passwords` or via the `opensearch-passwords` utility.
- Place those credentials in your Filebeat configuration along with the Wazuh Indexer CA certificate path to enable a secure connection to port 9200.

---

STAMUS central server setup

Below is a **step-by-step** guide showing how to forward events/logs from a **Stamus Central Server** to your **Windows-LOGSTASH** machine (which runs Logstash), and then have **Filebeat** forward those logs to Wazuh. The high-level flow looks like this:

```
pgsql
Copy
Stamus Central Server
     | (TCP/UDP Syslog, or EVE JSON over TCP)
     v
Windows-LOGSTASH (Logstash)
     | (local rotating file)
     v
Filebeat (on Windows-LOGSTASH)
     | (TLS/HTTPS to Wazuh Indexer)
     v
Wazuh (Indexer / Elasticsearch / Dashboard)
```

**Note**: Stamus solutions may support multiple output methods (e.g., EVE JSON over TCP, syslog, Kafka, etc.). This guide shows a **typical approach** using Syslog (or EVE JSON over TCP) to Logstash. You should adapt paths, ports, and configuration details per your specific Stamus product version and environment.

---

# 1. Configure Stamus Central Server to Send Logs to Logstash

## 1.1 Choose a Transport and Format

1. **Syslog** (UDP or TCP)
2. **EVE JSON** (TCP)

**Syslog** is a common approach, but if you need full Suricata EVE JSON with rich fields, you might prefer **EVE JSON** output. Either works for Logstash ingestion.

## 1.2 Example: Configure Syslog Output in Stamus

Depending on your Stamus version, you may have a UI or configuration file (e.g., `/etc/stamus-central/stamus-central.yaml`, or something similar). A generic example to output Syslog to a remote server (Windows-LOGSTASH) might look like:

```
yaml
Copy
```

```
logs:
  outputs:
    - type: syslog
      name: logstash-syslog
      enabled: true
      protocol: tcp            # or udp
      server: "WINDOWS-LOGSTASH"  # or the IP, e.g. "192.168.1.10"
      port: 5000
      format: rfc5424          # Possibly "rfc3164" or "structured"
      # Additional Stamus parameters or features as needed
```

### 1.3 Example: Configure EVE JSON over TCP in Stamus

If your Stamus installation (or Suricata-based sensor within Stamus) supports direct EVE JSON forwarding:

```
yaml
Copy
logs:
  outputs:
    - type: eve-json
      name: logstash-eve
      enabled: true
      protocol: tcp
      server: "WINDOWS-LOGSTASH"
      port: 5000
      # Potentially specify SSL: false/true, CA, certificate, etc. if
supported
```

**Important**: Consult the official Stamus documentation to confirm the exact configuration syntax and path for your version.

---

## 2. Configure Logstash on Windows-LOGSTASH to Receive Stamus Logs

### 2.1 Create a Logstash Pipeline

Create (or edit) a pipeline file, for example:

```
arduino
Copy
C:\logstash\config\pipelines\stamus.conf
```

#### Option A: Syslog Input

```
conf
Copy
input {
  tcp {
    port => 5000
    type => "stamus_syslog"
    # If your Stamus server sends syslog via UDP, use:
    # udp { port => 5000 type => "stamus_syslog" }
    codec => plain  # Typically syslog will be plain text
  }
}

filter {
  # Here you can parse the syslog lines further if you want,
  # e.g., using the syslog_pri or grok filters.
  # For a typical syslog parse:
  syslog_pri { }
  # Or:
  # grok {
  #   match => { "message" => "%{SYSLOGLINE}" }
  # }
}

output {
  # Write to a rolling file (local retention up to 20GB, for instance)
  # We'll reuse the 'rolled_file' plugin from earlier instructions.
  rolled_file {
    path => "C:/logstash-data/stamus-logs"
    filename_pattern => "stamus-%{+yyyyMMdd_HH}.log"
    file_size_limit => 21474836480  # 20GB
    max_roll_count => 1            # Keep 1 file, overwrite when new 20GB is reached
  }

  # (Optional) debug output to console:
  # stdout { codec => rubydebug }
}
```

**Option B: EVE JSON Input**

If you chose EVE JSON forwarding from Stamus (instead of syslog):

```
conf
Copy
input {
  tcp {
    port => 5000
    type => "stamus_eve"
    codec => json  # <--- Important! EVE logs are JSON
  }
}

filter {
  # Potential additional EVE field parsing or enrichment if desired
}

output {
  rolled_file {
    path => "C:/logstash-data/stamus-logs"
    filename_pattern => "stamus-%{+yyyyMMdd_HH}.log"
    file_size_limit => 21474836480  # 20GB
    max_roll_count => 1
  }
  # (Optional) debug:
  # stdout { codec => rubydebug }
}
```

**Note**:

- Adjust the **port** if you prefer something else (e.g., 5045) to avoid conflicts.
- Make sure your Windows Firewall or any other firewall on Windows-LOGSTASH allows inbound traffic on that port from the Stamus Central Server.

### 2.2 Run or Restart Logstash

If running Logstash as a service, ensure your pipeline config is included in its pipeline configuration path. Then **restart** Logstash so it picks up the new config:

```
powershell
Copy
# Example:
Stop-Service logstash
Start-Service logstash
```

Or if running in the foreground:

```
powershell
Copy
.\logstash.bat -f "C:\logstash\config\pipelines\stamus.conf"
```

---

## 3. Verify Logs Are Written Locally in Logstash

After a few moments, if Stamus is actively sending logs:

1. Check `C:\logstash-data\` for newly created files named something like `stamus-20250219_14.log` (timestamped).
2. Open the file in a text editor or run `Get-Content` in PowerShell to see if logs are indeed arriving.

---

## 4. Configure Filebeat on Windows-LOGSTASH to Forward Stamus Logs to Wazuh

Use the same approach as you did with the other logs (Windows event logs). Now you just need **another** input in `filebeat.yml`, or you can reuse the same if you wish to collect all logs from `C:\logstash-data\`.

### 4.1 Example Filebeat Config for Stamus Rotating Logs

In:

```
makefile
Copy
C:\Program Files\Filebeat\filebeat.yml
```

you can add or modify:

```
yaml
Copy
filebeat.inputs:
  - type: log
    enabled: true
    paths:
      - "C:/logstash-data/stamus-*.log"
    fields:
      source: "stamus"
      environment: "production"
    fields_under_root: true

    # If logs can be multi-line JSON, consider multiline settings:
    # multiline:
    #   pattern: '^{'
    #   negate: true
    #   match:  "after"

output.elasticsearch:
  # Connect to your Wazuh Indexer/Elasticsearch
  hosts: ["https://WAZUH-SERVER:9200"]
  username: "elastic"
  password: "YOUR_ELASTIC_PASSWORD"
  ssl.certificate_authorities: ["C:/Program Files/Filebeat/certs/wazuh-
ca.pem"]
  ssl.verification_mode: "full"
```

Adjust the **hosts**, **username**, **password**, and **ssl** settings as appropriate for **your** Wazuh installation.

### 4.2 Start Filebeat

```
powershell
Copy
Stop-Service filebeat
Start-Service filebeat
```

Or in a console:

```
powershell
Copy
filebeat.exe -c "C:\Program Files\Filebeat\filebeat.yml"
```

### 4.3 Verify in Wazuh

1. Go to your Wazuh Dashboard (or Kibana, if that’s the interface used).
2. Look for new logs with `fields.source: stamus` or parse them by `host.name`, `log.file.path`, etc.
3. Confirm the data from Stamus (Suricata/EVE or syslog events) is now visible in Wazuh indices.

---

## 5. Tips & Best Practices

1. **Firewall & Networking**
    - Open your chosen **Logstash port** on Windows-LOGSTASH (e.g., TCP/5000 or 5045).
    - Ensure your Stamus server can reach that port.
    - Make sure **port 9200** (or your Wazuh Indexer port) is open from Windows-LOGSTASH to Wazuh.
2. **Use TLS If Possible**
    - If Stamus supports sending EVE or Syslog over TLS, you can secure that traffic.
    - If not, at least use TLS from Filebeat to Wazuh.
3. **Rotate or Manage Disk Usage**
    - You already set up a rolling file with `rolled_file` plugin (max size = 20GB).
    - Confirm you have enough free disk space, or set `max_roll_count` to keep more than 1 file if you want a longer local history.
4. **Stamus Parsing**
    - Suricata EVE JSON logs can be richly parsed. If you want to do advanced parsing, you can expand your Logstash pipeline with `json` filters or a Suricata/EVE log pipeline. Or you can rely on Wazuh's indexing/mapping.
5. **Monitoring**
    - Check the Logstash logs (`C:\logstash\logs\`) and Filebeat logs (`C:\ProgramData\filebeat\logs\`) for errors or warnings.
    - In Wazuh, confirm logs appear in the correct index.
    - If using Kibana, you might create an index pattern for `filebeat-*` or whichever index name you defined in your output.

---

## 6. Putting It All Together

1. **Stamus Central Server**:
    - Configure a Syslog or EVE JSON output → point to `WINDOWS-LOGSTASH:5000`.
2. **Windows-LOGSTASH**:
    - Logstash input (TCP or UDP Syslog, or TCP JSON) on port 5000.
    - Store logs locally in `C:\logstash-data\stamus-*.log` with a rolling file up to 20GB.
3. **Filebeat**:
    - Watches `C:\logstash-data\stamus-*.log`.
    - Forwards to Wazuh Indexer (`https://WAZUH-SERVER:9200`) with the appropriate credentials and CA certificate.
4. **Wazuh**:
    - Receives logs, indexes them.
    - You can search and visualize them in the Wazuh dashboard.

---

# Conclusion

With these configurations:

- **Stamus** → (Syslog/EVE) → **Logstash** → (rolling file) → **Filebeat** → (TLS) → **Wazuh**.
- You get local resilience (rolling file up to 20GB) and a robust path to your Wazuh stack.
- If the link to Wazuh goes down, Filebeat will queue logs locally until it reconnects.
- You can parse, enrich, or filter the Stamus data in Logstash as needed.

This fulfills the requirement of forwarding **Stamus Central Server** logs/events to your existing pipeline (Logstash) and then on to **Wazuh** for indexing and visualization.