**🔹 2. Filebeat & Logstash**

**Scenario 1: Filebeat Setup and Log Ingestion**

**Q1:** How do you configure Filebeat to ship logs from a Linux server to Logstash?

**Answer:**

* Install Filebeat and configure filebeat.inputs for the log file path.
* Under output.logstash, specify Logstash host and port.

filebeat.inputs:

- type: log

paths:

- /var/log/auth.log

output.logstash:

hosts: ["localhost:5044"]

**Q2:** How do you test if Filebeat is sending logs correctly?

**Answer:**

* Use filebeat test output to test connectivity.
* Monitor Logstash logs and use filebeat -e for debug output.

**Q3:** Why would you prefer using Filebeat over Logstash for log collection on endpoints?

**Answer:**

* Lightweight and low resource usage.
* Easier deployment and management on multiple endpoints.
* Built-in modules for common log formats.

**Scenario 2: Logstash Pipeline Design**

**Q4:** How do you design a pipeline in Logstash to parse firewall logs and send to Elasticsearch?

**Answer:**

* Use input (beats), filter (grok, date, mutate), and output (Elasticsearch).

input {

beats { port => 5044 }

}

filter {

grok { match => { "message" => "%{IP:src\_ip} %{IP:dst\_ip} %{INT:port}" } }

date { match => ["timestamp", "ISO8601"] }

}

output {

elasticsearch { hosts => ["http://localhost:9200"] index => "firewall-logs" }

}

**Q5:** How do you handle logs that do not match the Grok pattern?

**Answer:**

* Use tag\_on\_failure => ["\_grokparsefailure"]
* Route failed logs to a separate index or file for analysis.

**Q6:** What is the benefit of using conditional logic in Logstash?

**Answer:**

* Allows routing logs based on source type, tags, content.
* Optimizes parsing and reduces processing load.

if [source] =~ /auth.log/ {

grok { match => { "message" => "%{SYSLOGBASE}" } }

}

**Scenario 3: Performance Tuning**

**Q7:** How do you ensure Filebeat does not overwhelm Logstash with data?

**Answer:**

* Use bulk\_max\_size and queue.mem settings in Filebeat.
* Use persistent queues and pipeline workers in Logstash.

**Q8:** What are persistent queues in Logstash and when are they useful?

**Answer:**

* Provide disk-based buffering.
* Prevent data loss if Elasticsearch is temporarily down.

queue.type: persisted

**Q9:** How do you debug slow processing in Logstash?

**Answer:**

* Enable --config.debug or use monitoring APIs.
* Analyze filter plugin timing.
* Check JVM heap usage and queue stats.

**Scenario 4: Cybersecurity-Specific Use Cases**

**Q10:** How do you enrich log data with threat intelligence in Logstash?

**Answer:**

* Use translate or elasticsearch filter plugin.
* Match IPs against known threat feeds.

**Q11:** How do you mask or remove sensitive fields from logs?

**Answer:**

* Use mutate to remove or replace fields.

mutate {

remove\_field => ["password"]

gsub => ["user\_email", "@.\*", "@redacted"]

}

**Q12:** How do you tag logs based on log source or type?

**Answer:**

* Use add\_tag in Filebeat inputs or Logstash filters.

- type: log

paths: ["/var/log/auth.log"]

tags: ["auth"]

**Scenario 5: Pipeline Management and Monitoring**

**Q13:** How can you test a Logstash pipeline configuration before deploying?

**Answer:**

* Use --config.test\_and\_exit to check for syntax errors.
* Use sample input files and stdin {} for dry run.

**Q14:** What is the use of multiple pipelines in Logstash?

**Answer:**

* Isolate log sources and processing logic.
* Improve performance and fault isolation.

**Q15:** How do you monitor Filebeat and Logstash health?

**Answer:**

* Filebeat: Use filebeat monitoring and logs.
* Logstash: Use X-Pack monitoring or REST API /\_node/stats.