



Ninja Level Infrastructure Monitoring

Defensive Approach to Security Monitoring and Automation

DEFCON 24

06th August 2016, Saturday 10:00 - 14:00

[Madhu Akula](#) & [Riyaz Walikar](#)

Appsecco.com

About

Madhu Akula

- Automation Security Ninja @Appsecco
- Security & DevOps lover
- Trainer & Speaker
- Acknowledged by more than 200 giant companies like Google, Microsoft, Yahoo, Adobe, etc.
- Open source contributor
- Never ending learner !
- [@madhuakula](#)

Riyaz Walikar

- Chief Offensive Security Officer @Appsecco
- Over 9 years of industry experience in breaking infrastructure security and web applications
- Security evangelist, leader for null Bangalore and OWASP chapters
- Trainer/Speaker : BlackHat Asia, BlackHat USA, nullcon Goa, nullcon Delhi, c0c0n, OWASP AppSec USA
- [@riyazwalikar](#)
- [@wincmdfu](#)

Ninja Level Infrastructure Monitoring

What will you learn today?

- Explain the Basics
 - Basics of logs and centralized monitoring
 - Introduction to ELK stack
- Setup
 - Log collections
 - ELK Stack
 - Monitoring Attack Patterns
 - Alerts and Notifications
- Automating the entire setup
 - Ansible Playbook
 - Docker Container

Ninja Level Infrastructure Monitoring

What will you learn today?

- Production best practices
 - Planning and Provisioning
 - Storing and Aggregations
 - Deployment and Post Deployment
- Alternative software options
- Next Steps
- Feedback and Q&A

Prerequisites

- As this workshop is intended for beginner to mid level, we are expecting that participants are comfortable with basic Linux commands
- Laptop with administrative privileges
- 20GB of free space in your hard disk for virtual machines
- Minimum 4GB RAM
- Enthusiasm to learn cool stuff 😊

Basics of logs and centralized logging

What are logs?

- Log is a collection of records or events.
- Logs used to be an (often indecipherable) line of text intended for offline human analysis of what went wrong.
- Logs are a critical part of any system, they give you insight into what a system is doing as well what happened.

Basics of logs and centralized logging

What is centralized logging?

- Virtually every process running on a system generates logs in some form or another.
- When your system grows to multiple hosts, managing the logs and accessing them can get complicated.
- Searching for a particular error across hundreds of log files on hundreds of servers is difficult without good tools.
- A common approach to this problem is to setup a centralized logging solution so that multiple logs can be aggregated in a central location.

Entering in to the ELK Stack

What is ELK?

ELK is a collection of different open source modules like Elasticsearch, Logstash, Kibana. This helps to any user to collect, analyze and visualize data in real-time. Each modules fits based on your use case and environment.

Elasticsearch is a NoSQL database that is based on the Lucene search engine. Logstash is a log pipeline tool that accepts inputs from various sources, executes different transformations, and exports the data to various targets. Kibana is a visualization layer that works on top of Elasticsearch.

Explaining the ELK stack

Elasticsearch

- Written in Java and uses Groovy
- Distributed and Highly available search engine
- Multi Tenant with Multi types
- Various set of APIs
- Document Oriented
- Built on top of Lucene
- (near) Real time search



elastic

Explaining the ELK stack

Logstash

- Written in Ruby and uses JRuby
- Tool for managing events and logs
- Centralized data processing of all types of logs
- Normalize varying schema and formats
- Quickly extend to custom log formats
- Consists of 3 main components

Input : Passing logs to process them into machine understandable format

Filter : Set of conditionals to perform specific action on a event

Output : Decision maker for processed events/logs



logstash

Explaining the ELK stack

Kibana

- Powerful front-end dashboard written in JavaScript
- Browser based analytics and search dashboard for elasticsearch
- Flexible analytics & visualization platform
- Instant sharing and embedding of dashboards
- Provides data in the form of charts, graphs, counts, maps, etc. in real-time

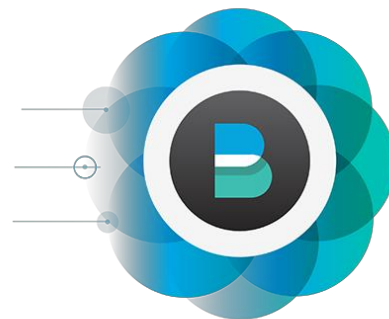


kibana

Explaining the ELK stack

Beats

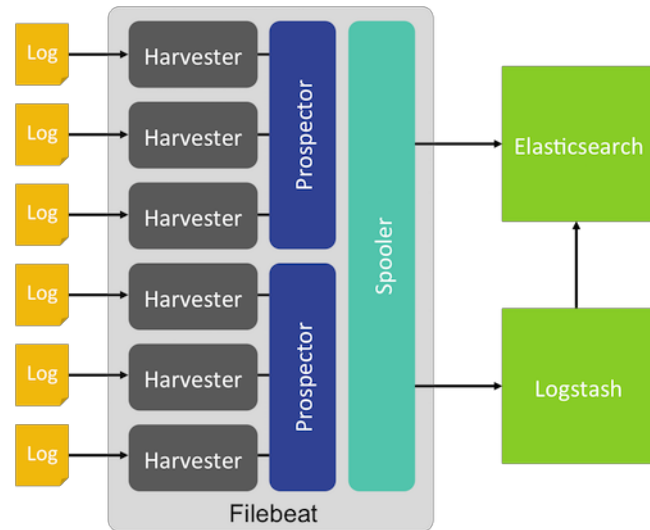
- Written in Go
- Lightweight shippers for Elasticsearch & Logstash
- Capture all sorts of operational data like logs, operating system metrics or network packet data
- It can send logs to either elasticsearch, logstash
- Different kind of beats
 - Libbeat** : The Go framework for creating new Beats
 - Packetbeat** : Tap into your wire data
 - Filebeat** : Lightweight log forwarder to Logstash & Elasticsearch
 - Winlogbeat** : Sends windows event logs



Explaining the ELK stack

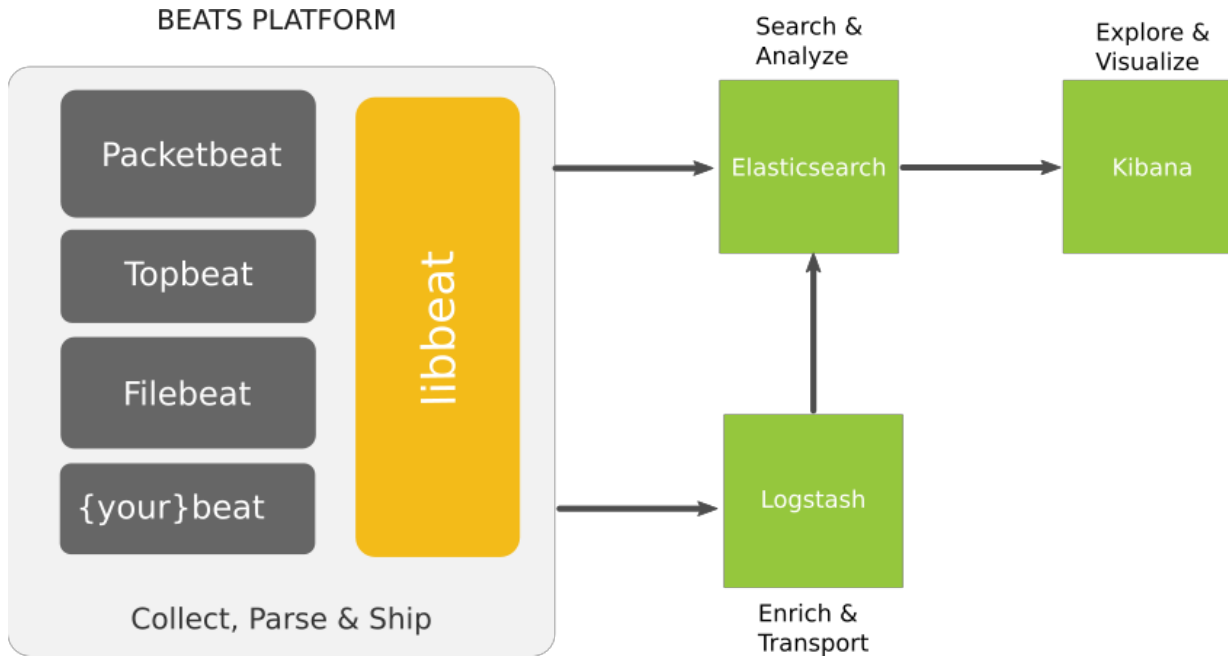
Filebeat

- Lightweight Shipper for Log Data
- Filebeat is an open source file harvester
- Used to fetch logs files and feed them into logstash
- It's replacement for logstash-forwarder
- Resolve problems with real-time kibana dashboards



Explaining the ELK stack

Overview Architecture



Why should we use ELK for Infrastructure Monitoring?

Challenges with traditional monitoring

- No consistency
- No centralization
- Accessibility of log data

Why should we use ELK for Infrastructure Monitoring?

No Consistency (it's difficult to be jack-of-all trades)

- Difficulty in logging for each application, system, device
- Interpreting the various types logs
- Variation in format makes it challenging to search
- Many types of time formats

Why should we use ELK for Infrastructure Monitoring?

No Centralization (simply put, log data is everywhere)

- Logs in many locations on various servers
- SSH + GREP doesn't scale

Solution

The ELK Stack is popular because it fulfills a need in the log analytics space. Splunk's enterprise software has long been the market leader, but its numerous functionalities are increasingly not worth the expensive price — especially for smaller companies such as products and tech startups.

ELK stack can help you manage each of these challenges, and more. ELK is best for time-series data—anything with a time stamp—such as you'll find in most web server logs, transaction logs, and stock data listings. To be intelligible, these logs usually need substantial clean-up.

Why should we use ELK for Infrastructure Monitoring?

Accessibility of Log Data (Much of the data is difficult to locate and manage)

- Access is often difficult
- High expertise to mine data
- Logs can be difficult to find
- Immense size of Log Data

History of ELK & Elastic, the company behind ELK

- **Shay Banon** created compass in 2004, third version of compass is the first version of elasticsearch. Which is released in Feb 2010
- **Jordan Sissel** creator of logstash.
- **Rashid Khan** creator of kibana.
- **Elasticsearch** merged the both logstash and kibana into the elasticsearch organization. Which is [elastic.co](https://www.elastic.co)
- After they included **Beats** which is formerly called as packetbeats
- Now they combined the stack and releasing as Elastic Stack and X-Pack by including commercial plugins and offerings



Let's get started with setup

Terminology before diving into the stack

Near Real-time (NRT)

Elasticsearch is a near real time search platform. What this means is there is a slight latency (normally one second) from the time you index a document until the time it becomes searchable.

Node

A node is a running instance of elasticsearch which belongs to a cluster.

Cluster

Cluster is a collection of one or more nodes that together holds your entire data and provides indexing and search capabilities across all nodes. It shares the same cluster name across nodes. Default name is

Terminology before diving into the stack

Document

A document is a basic unit of information that can be indexed. It is a JSON document which is stored in elasticsearch. It is like a row in a table in a relational database.

Index

An index is a collection of documents that have somewhat similar characteristics. An index is like a database in a relational database. It has a mapping which defines multiple types.

Type

Within an index, you can define one or more types. A type is a logical category/partition of your index whose semantics is completely up to you. In general, a type is defined for documents that have a set of common fields.

Terminology before diving into the stack

Shard & Replica

- Every index can be split into multiple shards to be able to distribute data.
- The shard is the atomic part of an index, which can be distributed over the cluster if you add more nodes.
- It allows you to horizontally split/scale your content volume
- By default 5 primary shards and 1 replica shards are created while starting elasticsearch
- Elasticsearch allows you to make one or more copies of your index's shards into what are called replica shards, or replicas for short.
- Atleast 2 Nodes are required for replicas to be created. It provides high availability in case a shard/node fails.

Hands On

Hands On

Hands On

Hands On

Hands On

Hands On

Hands On



Monitoring Attack Patterns with Dashboards & notifications

Hands On



Automating the entire setup with
Ansible & Docker
for
Production Ready

Hands On

Hands On



Production best practices
for
ELK

Reference Docs



Alternative software options
for
ELK

Reference Docs



Next Steps
and
Road Map

Next steps for your road

Reference Docs

References

- <https://elastic.co>
- <http://logz.io>
- <https://github.com/clay584>
- <http://www.slideshare.net/prajalkulkarni/attack-monitoring-using-elasticsearch-logstash-and-kibana>
- <https://en.wikipedia.org>
- <http://jasonwilder.com/blog/2012/01/03/centralized-logging/>
- <https://qbox.io/blog/welcome-to-the-elk-stack-elasticsearch-logstash-kibana>
- Etc.



THANK YOU

@madhuakula

@riyazwalikar

@appseccouk

www.appsecco.com