IT366

Lab 3

Security Incident and Event Management (SIEMs)

Dan Kindt

# Table of Contents

[Executive Summary 3](#_Toc1390894)

[Questions 4](#_Toc1390895)

[Conclusion 7](#_Toc1390896)

[References 8](#_Toc1390897)

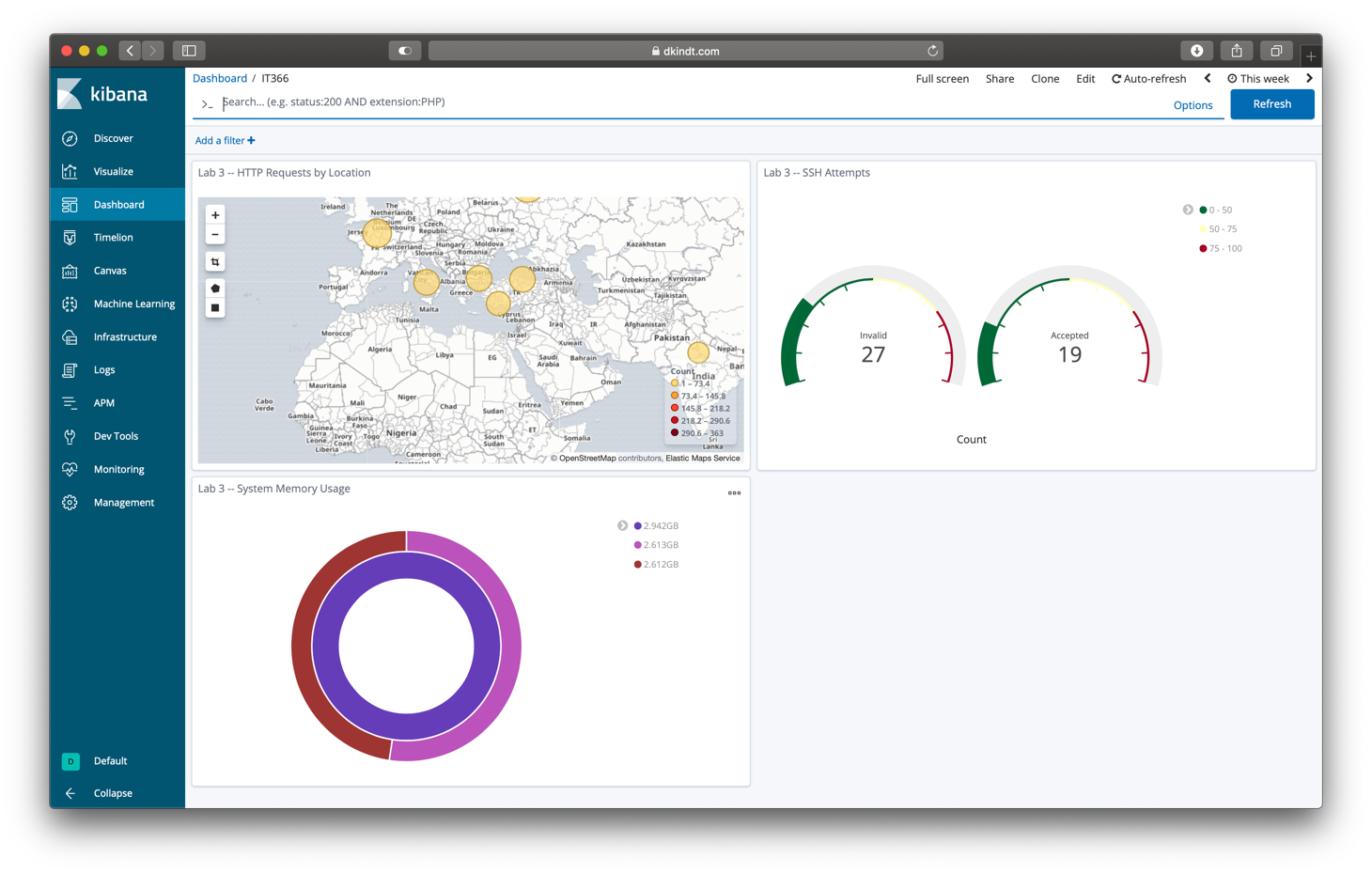
[Appendix 9](#_Toc1390898)

## Executive Summary

With businesses moving their operations into the *cloud*, it is increasingly important to monitor remote resources to ensure their businesses are able to run smoothly. Security information and event management systems (SIEMs) are an effective way to accomplish this goal. For the purposes of this lab, the Elasticsearch, Logstash, and Kibana (ELK) stack was installed on a Linux instance hosted by the Google Cloud Platform. The main objective was to create a dashboard in Kibana with three custom visualizations that would allow monitoring: SSH attempts, HTTP traffic, and system memory usage.

## Questions

1. What commands did you use to start and persistently run the components at reboot?
   1. `sudo systemctl *enable* <component-name>`
2. What are two other SIEMs?
   1. Splunk
   2. SolarWinds
3. What is the difference between Filebeat, Metricbeat, and Auditbeat?
   1. Metricbeat
      1. Collects various metrics and statistics from systems and services. This includes (but is not limited to): CPU, memory, Redis, Nginx.
   2. Filebeat
      1. Forward and centralize logs and files from all servers (sometimes thousands) to Kibana.
   3. Auditbeat
      1. Collects framework data from Linux as well as monitors the integrity of files on a given server and passes the information to elasticsearch.
4. Do you consider your stack secure? Why or why not? What could you improve?
   1. Yes, I do. I have enabled HTTPS, forwarded all HTTP requests, and added a login to my instance.
   2. User login credentials for the TAs to access Kibana at `dkindt.com` have been created:
      1. Username: `it366TA`
      2. Password: `MD8pQx$$@6C65uy6`
5. What was one thing you liked about the lab?
   1. Setting up Nginx, HTTPS, and my domain for the instance. It was also useful to be able to visual log data and I hope that we are able to come back to that aspect.
6. What was the hardest part of this lab? Explain.
   1. Building the visuals, because the documentation and tutorials seemed very dense and difficult to read. For example, getting my visual for HTTP requests to work with the region map.
7. Demonstrate that you have fulfilled the requirements.
   1. Dashboard with custom visualizations:



* 1. HTTPS with login:

A screenshot of a cell phone

Description automatically generated

## Conclusion

The goal to setup Elasticsearch, Logstash, and Kibana in order to better monitor was successful. It was very helpful to be able to view where HTTP traffic and SSH login attempts were originating from and to be able to build a dataset over time in order to continually secure operations. Being able to quickly view the breakdown of system memory used versus total, it allows us to allocate or release resources depending on needs. Overall, it was important to be able to quickly search and visualize system information and logs.

## References

<https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-ubuntu-18-04>

<https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-18-04>

<https://www.digitalocean.com/community/tutorials/how-to-use-kibana-dashboards-and-visualizations>

## Appendix

1. ELK Topology:

A screenshot of a cell phone

Description automatically generated

1. Beats configs:
   1. filebeat.yml:
2. *###################### Filebeat Configuration #########################  
     
   #=========================== Filebeat inputs =============================*filebeat.inputs:  
     
   *# Each - is an input. Most options can be set at the input level, so  
   # you can use different inputs for various configurations.  
   # Below are the input specific configurations.*- type: log  
     
    *# Change to true to enable this input configuration.* enabled: false  
     
    *# Paths that should be crawled and fetched. Glob based paths.* paths:  
    - /var/log/\*.log  
    - /var/log/nginx/access.log  
     
   *#============================= Filebeat modules ===============================*filebeat.config.modules:  
    *# Glob pattern for configuration loading* path: ${path.config}/modules.d/\*.yml  
     
    *# Set to true to enable config reloading* reload.enabled: false  
     
   *#==================== Elasticsearch template setting ==========================*setup.template.settings:  
    index.number\_of\_shards: 3  
    index.number\_of\_replicas: 0  
     
   *#================================ General =====================================  
     
   # The name of the shipper that publishes the network data. It can be used to group  
   # all the transactions sent by a single shipper in the web interface.*name: elk-01  
     
   *# The tags of the shipper are included in their own field with each  
   # transaction published.*tags: ["dkindt-ELK"]  
     
   *#============================== Kibana =====================================  
     
   # Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.  
   # This requires a Kibana endpoint configuration.*setup.kibana:  
     
    *# Kibana Host  
    # Scheme and port can be left out and will be set to the default (http and 5601)  
    # In case you specify and additional path, the scheme is required: http://localhost:5601/path  
    # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601* host: "localhost:5601"  
     
   *#================================ Outputs =====================================  
     
   #----------------------------- Logstash output --------------------------------*output.logstash:  
    *# The Logstash hosts* hosts: ["localhost:5044"]  
     
    *# Optional SSL. By default is off.  
    # List of root certificates for HTTPS server verifications  
    #ssl.certificate\_authorities: ["/etc/pki/root/ca.pem"]  
     
    # Certificate for SSL client authentication  
    #ssl.certificate: "/etc/pki/client/cert.pem"  
     
    # Client Certificate Key  
    #ssl.key: "/etc/pki/client/cert.key"  
     
   #================================ Processors =====================================  
     
   # Configure processors to enhance or manipulate events generated by the beat.*processors:  
    - add\_host\_metadata: ~  
    - add\_cloud\_metadata: ~
   1. metricbeat.yml:
3. metricbeat.modules:  
   - module: system  
    metricsets: ["diskio", "filesystem", "fsstat"]  
    diskio.include\_devices: ["sda", "sda1"]  
    filesystem.ignore\_types: [nfs, smbfs, autofs]  
     
   metricbeat.config.modules:  
    path: ${path.config}/modules.d/\*.yml  
    reload.enabled: false  
     
   setup.template.settings:  
    index.number\_of\_shards: 1  
    index.number\_of\_replicas: 0  
    index.codec: best\_compression  
   name:  
    elk-01  
   tags: ["elk-stack"]  
     
   setup.kibana:  
    host: "localhost:5601"  
     
   output.logstash:  
    hosts: ["localhost:5044"]  
     
   processors:  
    - add\_host\_metadata: ~  
    - add\_cloud\_metadata: ~
   1. auditbeat.yml:
4. auditbeat.modules:  
     
   - module: auditd  
    audit\_rules: |  
    -a always,exit -F arch=b32 -S all -F key=32bit-abi  
    -a always,exit -F arch=b64 -S execve,execveat -k exec  
    ## Identity changes.  
    -w /etc/group -p wa -k identity  
    -w /etc/passwd -p wa -k identity  
    -w /etc/gshadow -p wa -k identity  
    ## Unauthorized access attempts.  
    -a always,exit -F arch=b64 -S open,creat,truncate,ftruncate,openat,open\_by\_handle\_at -F exit=-EACCES -k access  
    -a always,exit -F arch=b64 -S open,creat,truncate,ftruncate,openat,open\_by\_handle\_at -F exit=-EPERM -k access  
   - module: file\_integrity  
    paths:  
    - /bin  
    - /usr/bin  
    - /sbin  
    - /usr/sbin  
    - /etc  
     
   *# REMOVE BELOW (Lines 26-31) if using Auditbeat 6.5.4. System module is only compatible with 6.6 onwards.*- module: system  
    datasets:  
    - host *# General host information, e.g. uptime, IPs* - process *# Started and stopped processes* - socket *# Opened and closed sockets* - user *# User information* state.period: 12h  
    user.detect\_password\_changes: true  
     
   setup.template.settings:  
    index.number\_of\_shards: 1  
    index.number\_of\_replicas: 0  
     
   *#tags: ["service-X", "web-tier"]*setup.kibana:  
    host: "localhost:5601"  
     
   output.logstash:  
    hosts: ["localhost:5044"]  
     
   processors:  
    - add\_host\_metadata: ~  
    - add\_cloud\_metadata: ~
5. Input configs
   1. 02-beats-input.conf:
6. input {  
    beats {  
    port => 5044  
    }  
   }
7. Filter configs:
   1. 10-syslog-filter.conf:
8. filter {  
    if [fileset][module] == "system" {  
    if [fileset][name] == "auth" {  
    grok {  
    match => { "message" => ["%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} sshd(?:\[%{POSINT:[system][auth][pid]}\])?: %{DATA:[system][auth][ssh][event]} %{DATA:[system][auth][ssh][method]} for (invalid user )?%{DATA:[system][auth][user]} from %{IPORHOST:[system][auth][ssh][ip]} port %{NUMBER:[system][auth][ssh][port]} ssh2(: %{GREEDYDATA:[system][auth][ssh][signature]})?",  
    "%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} sshd(?:\[%{POSINT:[system][auth][pid]}\])?: %{DATA:[system][auth][ssh][event]} user %{DATA:[system][auth][user]} from %{IPORHOST:[system][auth][ssh][ip]}",  
    "%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} sshd(?:\[%{POSINT:[system][auth][pid]}\])?: Did not receive identification string from %{IPORHOST:[system][auth][ssh][dropped\_ip]}",  
    "%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} sudo(?:\[%{POSINT:[system][auth][pid]}\])?: \s\*%{DATA:[system][auth][user]} :( %{DATA:[system][auth][sudo][error]} ;)? TTY=%{DATA:[system][auth][sudo][tty]} ; PWD=%{DATA:[system][auth][sudo][pwd]} ; USER=%{DATA:[system][auth][sudo][user]} ; COMMAND=%{GREEDYDATA:[system][auth][sudo][command]}",  
    "%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} groupadd(?:\[%{POSINT:[system][auth][pid]}\])?: new group: name=%{DATA:system.auth.groupadd.name}, GID=%{NUMBER:system.auth.groupadd.gid}",  
    "%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} useradd(?:\[%{POSINT:[system][auth][pid]}\])?: new user: name=%{DATA:[system][auth][user][add][name]}, UID=%{NUMBER:[system][auth][user][add][uid]}, GID=%{NUMBER:[system][auth][user][add][gid]}, home=%{DATA:[system][auth][user][add][home]}, shell=%{DATA:[system][auth][user][add][shell]}$",  
    "%{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} %{DATA:[system][auth][program]}(?:\[%{POSINT:[system][auth][pid]}\])?: %{GREEDYMULTILINE:[system][auth][message]}"] }  
    pattern\_definitions => {  
    "GREEDYMULTILINE"=> "(.|\n)\*"  
    }  
    remove\_field => "message"  
    }  
    date {  
    match => [ "[system][auth][timestamp]", "MMM d HH:mm:ss", "MMM dd HH:mm:ss" ]  
    }  
    geoip {  
    source => "[system][auth][ssh][ip]"  
    target => "[system][auth][ssh][geoip]"  
    }  
    }  
    else if [fileset][name] == "syslog" {  
    grok {  
    match => { "message" => ["%{SYSLOGTIMESTAMP:[system][syslog][timestamp]} %{SYSLOGHOST:[system][syslog][hostname]} %{DATA:[system][syslog][program]}(?:\[%{POSINT:[system][syslog][pid]}\])?: %{GREEDYMULTILINE:[system][syslog][message]}"] }  
    pattern\_definitions => { "GREEDYMULTILINE" => "(.|\n)\*" }  
    remove\_field => "message"  
    }  
    date {  
    match => [ "[system][syslog][timestamp]", "MMM d HH:mm:ss", "MMM dd HH:mm:ss" ]  
    }  
    }  
    }  
   }
   1. 11-nginx-filter.conf:
9. filter {  
    if [fileset][module] == "nginx" {  
    if [fileset][name] == "access" {  
    grok {  
    match => { "message" => ["%{IPORHOST:[nginx][access][remote\_ip]} - %{DATA:[nginx][access][user\_name]} \[%{HTTPDATE:[nginx][access][time]}\] \"%{WORD:[nginx][access][method]} %{DATA:[nginx][access][url]} HTTP/%{NUMBER:[nginx][access][http\_version]}\" %{NUMBER:[nginx][access][response\_code]} %{NUMBER:[nginx][access][body\_sent][bytes]} \"%{DATA:[nginx][access][referrer]}\" \"%{DATA:[nginx][access][agent]}\""] }  
    remove\_field => "message"  
    }  
    mutate {  
    add\_field => { "read\_timestamp" => "%{@timestamp}" }  
    }  
    date {  
    match => [ "[nginx][access][time]", "dd/MMM/YYYY:H:m:s Z" ]  
    remove\_field => "[nginx][access][time]"  
    }  
    useragent {  
    source => "[nginx][access][agent]"  
    target => "[nginx][access][user\_agent]"  
    remove\_field => "[nginx][access][agent]"  
    }  
    geoip {  
    source => "[nginx][access][remote\_ip]"  
    target => "[nginx][access][geoip]"  
    }  
    }  
    else if [fileset][name] == "error" {  
    grok {  
    match => { "message" => ["%{DATA:[nginx][error][time]} \[%{DATA:[nginx][error][level]}\] %{NUMBER:[nginx][error][pid]}#%{NUMBER:[nginx][error][tid]}: (\\*%{NUMBER:[nginx][error][connection\_id]} )?%{GREEDYDATA:[nginx][error][message]}"] }  
    remove\_field => "message"  
    }  
    mutate {  
    rename => { "@timestamp" => "read\_timestamp" }  
    }  
    date {  
    match => [ "[nginx][error][time]", "YYYY/MM/dd H:m:s" ]  
    remove\_field => "[nginx][error][time]"  
    }  
    }  
    }  
   }
10. Output configs:
    1. 30-elasticsearch.conf:
11. output {  
     elasticsearch {  
     hosts => ["localhost:9200"]  
     manage\_template => false  
     index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"  
     }  
    }
12. logstash.yml:
13. path.data: /var/lib/logstash  
    path.logs: /var/log/logstash
14. elasticsearch.yml:
15. cluster.name: it366-elk  
    node.name: node-1  
    path.data: /var/lib/elasticsearch  
    path.logs: /var/log/elasticsearch  
    network.host: 127.0.0.1  
    http.port: 9200
16. kibana.yml:
17. server.port: 5601  
    server.host: "0.0.0.0"  
    server.name: "dkindt.com"  
    elasticsearch.hosts: ["http://localhost:9200"]