

Full-Stack Secure Network & Application Infrastructure with SIEM Monitoring



By: Jared Lowe

F25 Senior Capstone

August 13th - December 5th

TOC

1. Project Overview.....pg. 3
2. Project Purposepg. 4
3. Objectives.....pg. 4
4. Scope of Work.....pg. 5
5. Costs and Risks.....pg. 6
6. Issues.....pg. 6
7. Lessons Learned.....pg. 7
8. Implementation.....pg 8-67
9. Conclusions & Findings.....pg. 68

1. Project Overview

This final report documents my final status for the semester-long project. The overall goal of this project is to design and implement a secure, enterprise-like IT environment that mirrors the operational complexity and challenges of a small business network. The project includes deploying servers, databases, and a web application, surrounded by layers of network security, monitoring, and logging, to simulate what a modern IT or cybersecurity professional might encounter in the field. Over the course of the build, each system was configured, secured, and integrated into the SIEM for real-time monitoring and analysis. After I will evaluate the findings that I found and do/state what I would do to keep it a secure enterprise network.

This project allows me to integrate several disciplines: networking, systems administration, cybersecurity, and web development into a single, complex, and unified environment. By the end of the semester, I have learned and built a functioning network system capable of not only hosting services securely but also detecting and monitoring potential security events through a Security Information and Event Management (SIEM) system.

This report documents every component of the environment, including system configurations, firewall routing, application deployment, and SIEM integration. Each screenshot included contains a clear description and explains the impact on the overall project.

This report focuses on the entirety of the project, which includes:

- Setting up the virtual infrastructure and network topology
- Deploying and configuring the pfSense firewall
- Installing and hardening Linux and Windows virtual machines
- Building and securing a PostgreSQL database
- Developing and hosting a Ruby on Rails web application
- Enabling basic application-level logging for future SIEM monitoring

2. Project Purpose

The purpose is to simulate a real-world enterprise IT infrastructure that aims to represent how a modern organization might create a combination of web applications, databases, and firewalls under the supervision of a centralized monitoring system:

- Securely segmented — isolating internal networks from external threats
- Application-driven — featuring a working, database-backed web app
- Monitoring-ready — structured to integrate with a SIEM for visibility and alerting

By completing this build, I'm gaining not only technical experience but also the ability to think like both a system administrator, SOC analyst, and a cybersecurity analyst.

3. Objectives

1. Implemented network segmentation, routing, and security policies using pfSense.
2. Deployed and configured a Linux-based web application and PostgreSQL database.
3. Built a functional monitoring environment capable of SIEM-level logging, analysis, and alerting.
4. Simulated a realistic enterprise structure by integrating servers, workstations, and network appliances across multiple isolated VLANs and firewall zones.
5. Developed hands-on proficiency in system administration, networking, Linux hardening, and secure configuration practices.
6. Applied cybersecurity principles through controlled offensive testing and log correlation.
7. Documented the entire implementation process with clear, structured, and professional-level detail.

4. Scope of Work

This project focused on building the infrastructure, including network segmentation, firewall architecture, virtual machines, database systems, logging pipelines, and security monitoring. After validating system connectivity and log ingestion, the environment was expanded to include Pi-hole, Filebeat agents, and staged cyberattacks to test SIEM visibility.

The table below outlines each major task and its completion status:

Steps	Time	Completed
VMWare virtual networks + machines	Week 1	Yes
pfSense firewall setup	Week 2	Yes
Linux & Windows VM setup	Week 3	Yes
PostgreSQL database creation	Week 3-5	Yes
Ruby on Rails app deployment	Week 5-7	Yes
Siem configuration and alerting	Week 7-8	yes
PiHole	Week 8-9	Yes
Cyber attacks and logging	Week 9-13	Yes
Constructing stronger networking after attacks	Week 12-14	Yes
Documentation and Final	Week 14	Yes

5. Costs and Risks

I have not and will not buy any software used or code to reach this point in my project. I have been loaned an SSD by Dr.Wimmer for my project. All resources and software were found for free online.

The overall risk exposure is low. Most technical risks are mitigated through documentation, version control, and the use of VMware snapshots. Since I also used backups everyday I did not run into any problems.

6. Issues

Throughout the project, I faced several challenges, primarily related to network connectivity and firewall configuration. Early on, devices were unable to communicate properly because pfSense rules or interfaces were misconfigured. Troubleshooting these issues involved trial, error, and learning, mostly in understanding how VMware's virtual networks communicate with pfSense.

I received helpful guidance from Dr. Wimmer, who walked me through several key troubleshooting steps and explained underlying networking principles. While these setbacks delayed progress slightly, they significantly deepened my understanding of both networking fundamentals and pfSense's architecture.

Now, all systems communicate cleanly, firewall rules are properly segmented, and the environment runs smoothly. These early difficulties turned into valuable learning experiences that strengthened my troubleshooting skills and confidence with networked environments.

With Wimmers help I got the systems online and communicated and the rest of the process went smoothly as I kept regular backups and snapshots.

7. Lessons Learned

This project provided significant hands-on experience in both technical execution and structured project management. Key lessons learned include:

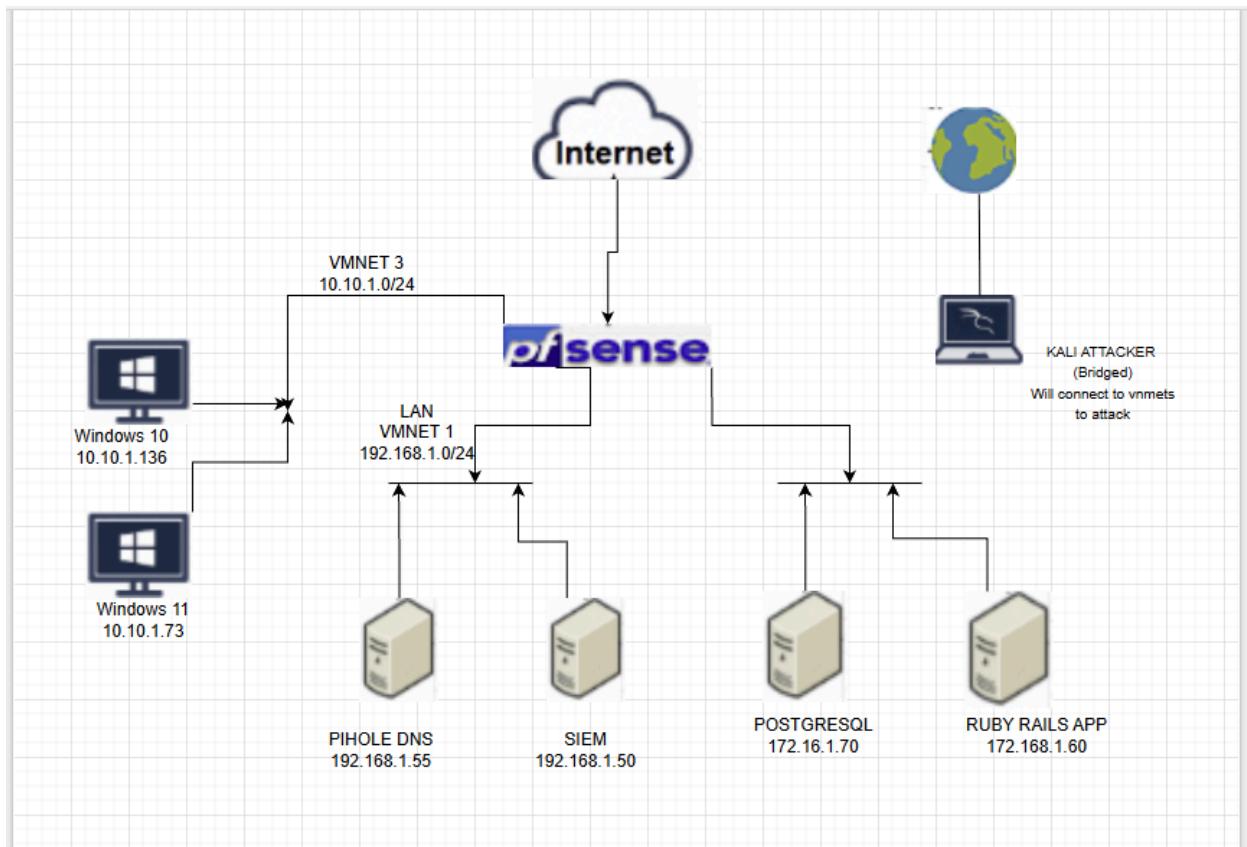
- Documentation is essential. Maintaining detailed notes on IP addresses, passwords, keys, firewall rules, and system configurations prevented confusion and helped quickly resolve issues especially during pfSense and filebeat troubleshooting.
- Planning network segmentation early avoids later problems. The importance of designing VLANs, interfaces, and firewall policies before deployment became clear once systems began interacting across different networks. Static IPs for pfsense worked faster.
- Infrastructure tasks are interconnected. Fixing one issue often required understanding several layers: routing, DNS, firewalls, OS configuration, and log forwarding mirroring real enterprise environments. Doing one task meant checking the other machines and pfsense.
- Cybersecurity visibility matters. The SIEM component reinforced how logs, alerts, and detections form the first line of defense for SOC analysts.
- Hands-on testing builds stronger understanding than theory in class. Running attacks from Kali and observing their traces in Elasticsearch provided real-world insight into how intrusion attempts appear from a defender's perspective.

Overall, this project significantly enhanced my technical confidence and prepared me for real-world roles in networking, systems administration, and cybersecurity operations.

8. Implementation

1. Network Architecture and Topology

2. VMnet	3. Subnet	4. Purpose
5. VMnet8	6. NAT / Bridged	7. Internet access (WAN)
8. VMnet1	9. 192.168.10.0/24	10. Internal LAN & Management
11. VMnet3	12. 10.10.1.0/24	13. Workers sim
14. VMnet4	15. 172.16.1.0/24	16. Application services



2. Created a VMware virtual network:

Here I set up vmware workstation pro to utilize all vms and use the virtual network editor.

← VMware Workstation Pro (For Windows) 17.6.4

Primary Downloads Open Source

Search: Version: 17.6.4 Release ID: 533272 Language: English

I agree to the [Terms and Conditions](#)

Product Download

File Name	Release Date	Last Updated	SHA2	MD5
VMware Workstation Pro for Windows VMware-workstation-full-17.6.4-24832109.exe(405.72 MB) Build Number: 24832109	Jul 15, 2025	Jul 09, 2025 b94ba9	10fe3a36f525d88aa133118ab3b5a16b18da88d4aa11b14d74e4164b3f	b387e0a655798ba356d9a7331d98851a

Here is where I used the network editor to configure our network map to distribute different networks to each vm.

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet1	Host-only	-	Connected	-	192.168.1.0
VMnet3	Host-only	-	Connected	-	10.10.1.0
VMnet4	Host-only	-	Connected	Enabled	172.16.1.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.238.0

Add Network... Remove Network... Rename Network...

VMnet Information

Bridged (connect VMs directly to the external network)
Bridged to:

NAT (shared host's IP address with VMs)

Host-only (connect VMs internally in a private network)

Connect a host virtual adapter to this network
Host virtual adapter name: VMware Network Adapter VMnet8

Use local DHCP service to distribute IP address to VMs

Subnet IP: Subnet mask:

Administrator privileges are required to modify the network configuration.

Restore Defaults Import... Export... OK Cancel Apply Help

3. Created the remaining VMs and attached them to the correct VMnets. Also, power on VMs and perform initial OS installs/updates on each VM where i set the static ips that correlated with the range.

4.pfsense→ attached all VMNETSs to it

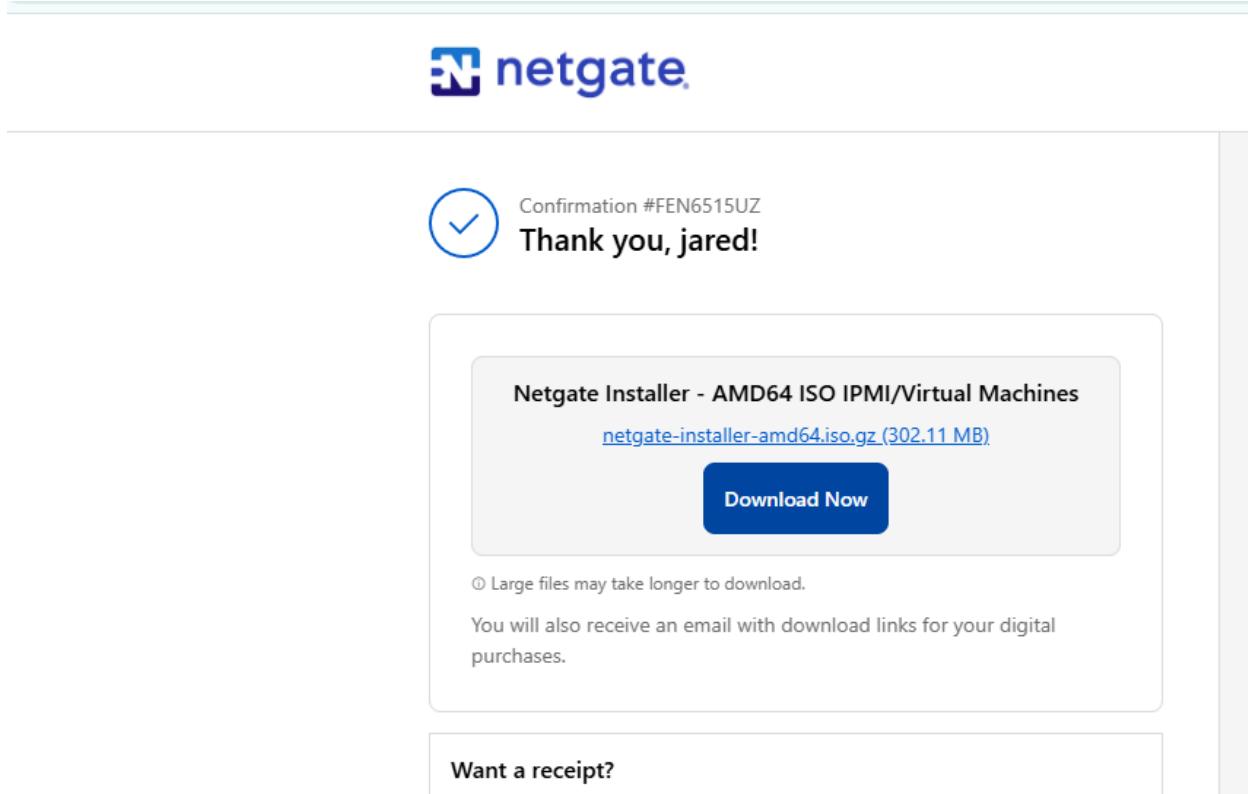


figure 4.1

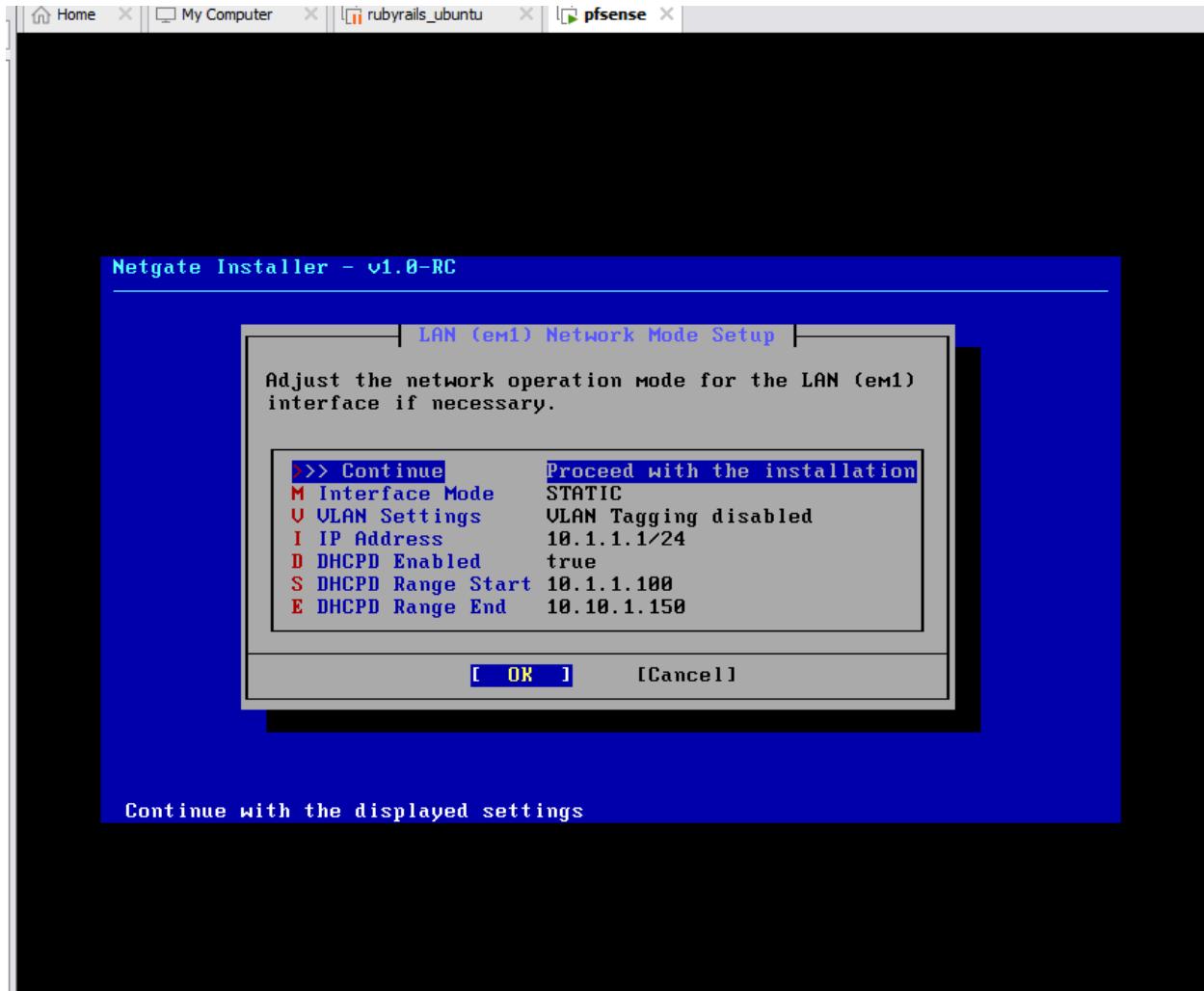


figure 4.2

```

pfSense 2.8.1-RELEASE amd64 20250909-1629
Bootup complete

FreeBSD/amd64 (pfSense.home.arpa) (ttyv0)

VMware Virtual Machine - Netgate Device ID: 3d404f301c950cd0d19c

*** Welcome to pfSense 2.8.1-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0 -> v4/DHCP4: 192.168.102.129/24
LAN (lan)      -> em1 -> v4: 192.168.1.10/24
DATA (opt1)    -> em2 -> v4: 172.16.1.10/24
WORKERS (opt2) -> em3 -> v4: 10.10.1.10/24

```

figure 4.3

5.ubuntu-ruby-rails → vmnet4

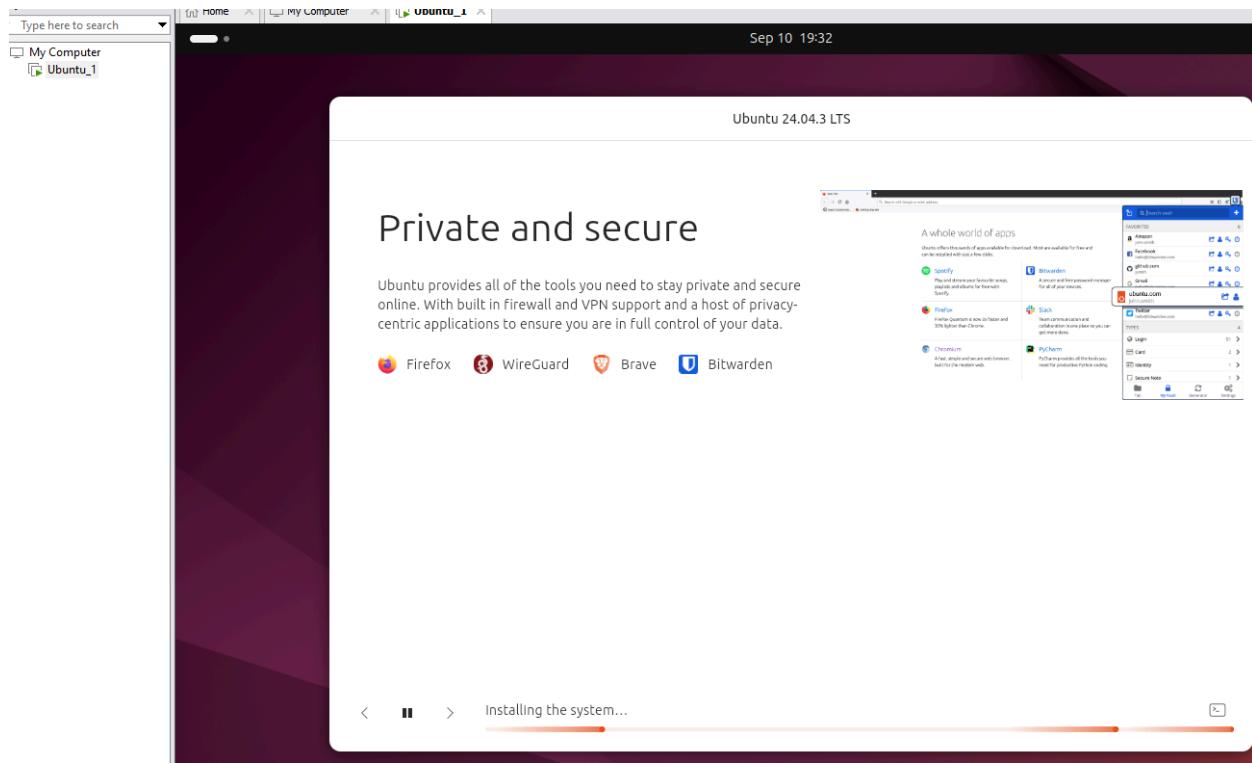


figure 5.1

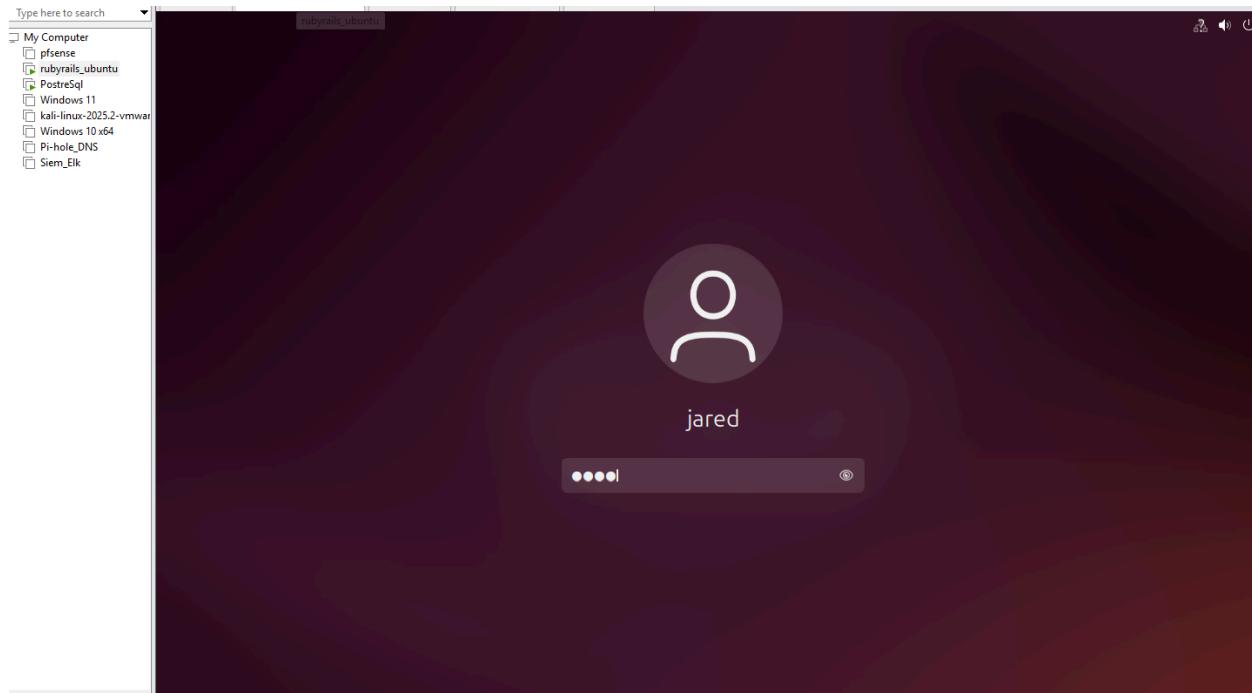


figure 5.2

6. ubuntu-postgresql → vmnet4

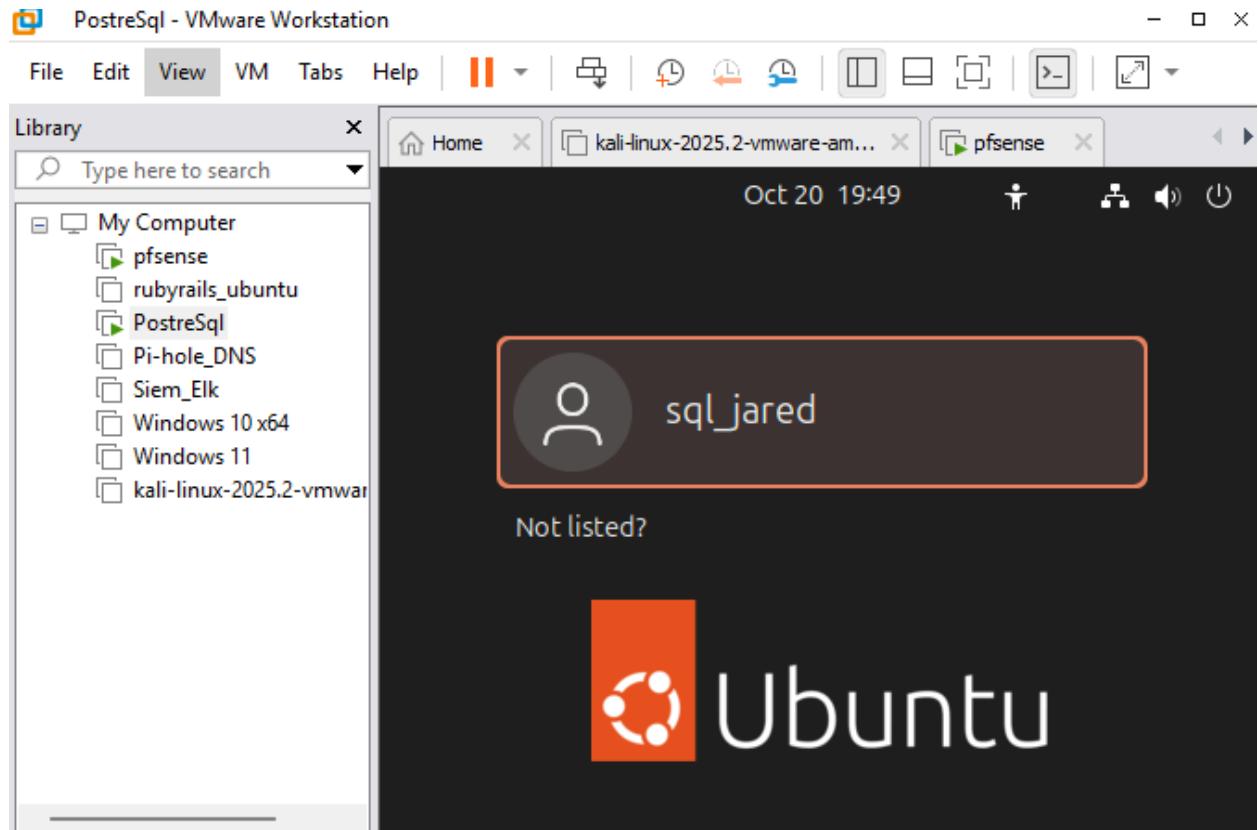


figure 6.1

7.ubuntu-siem-elk →vmnet1

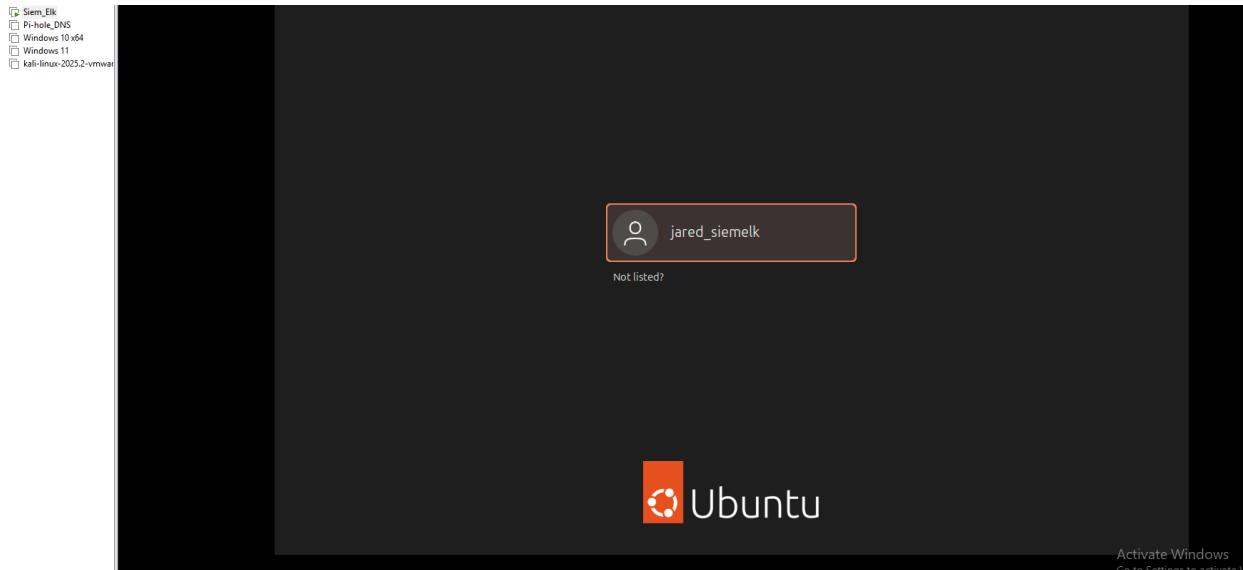


figure 7.1

8. Windows 11 → vmnet3

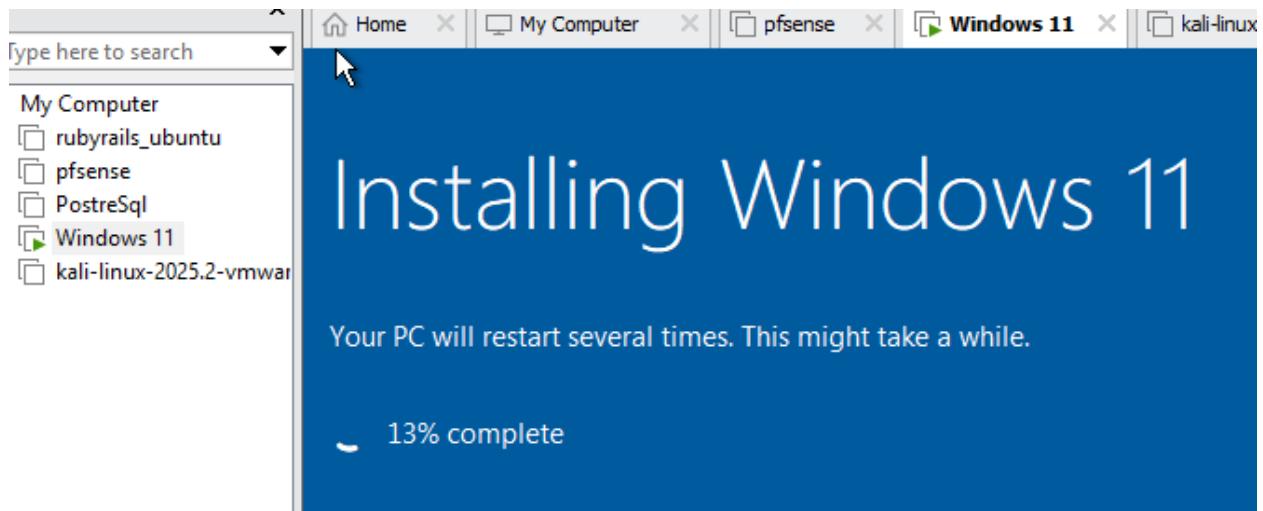


figure 8.1

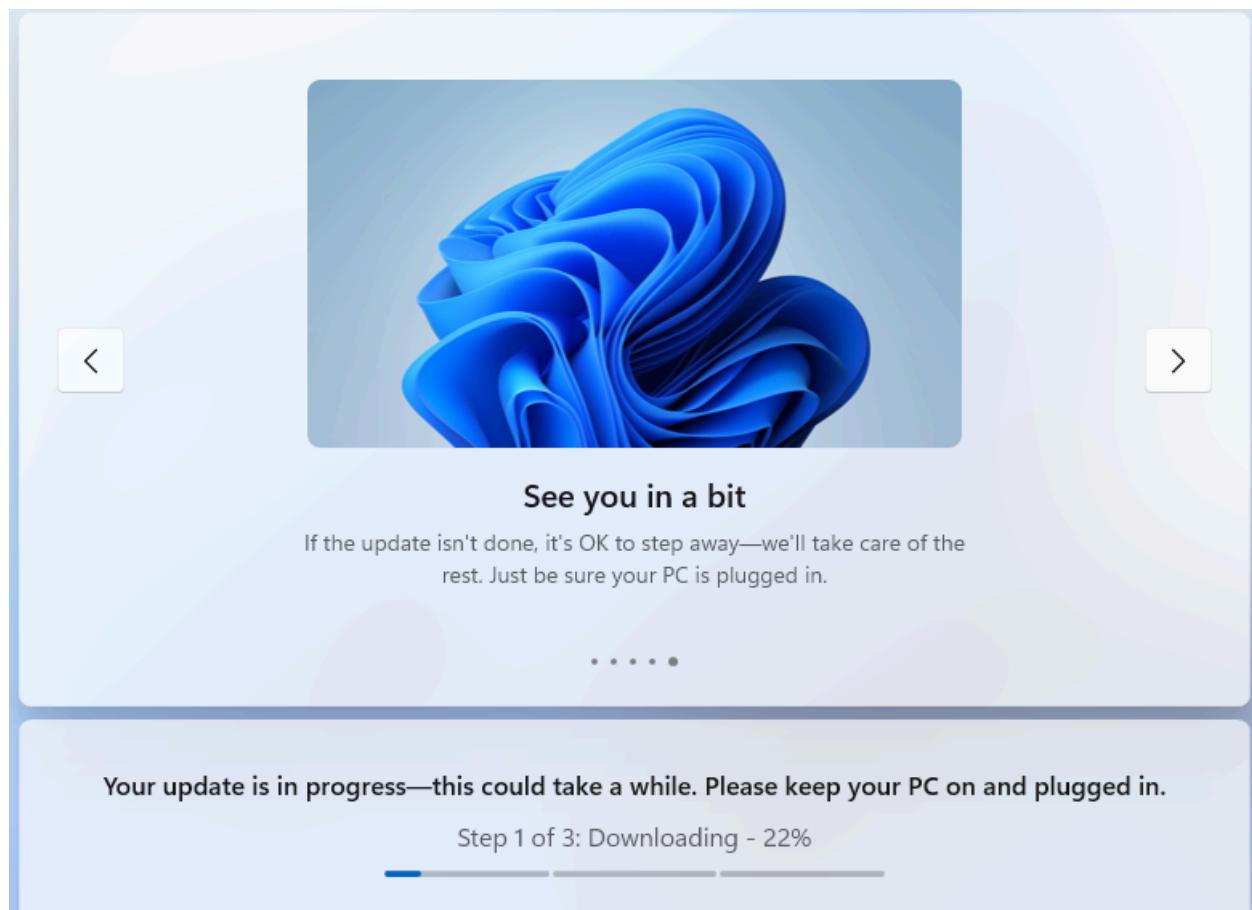


figure 8.2

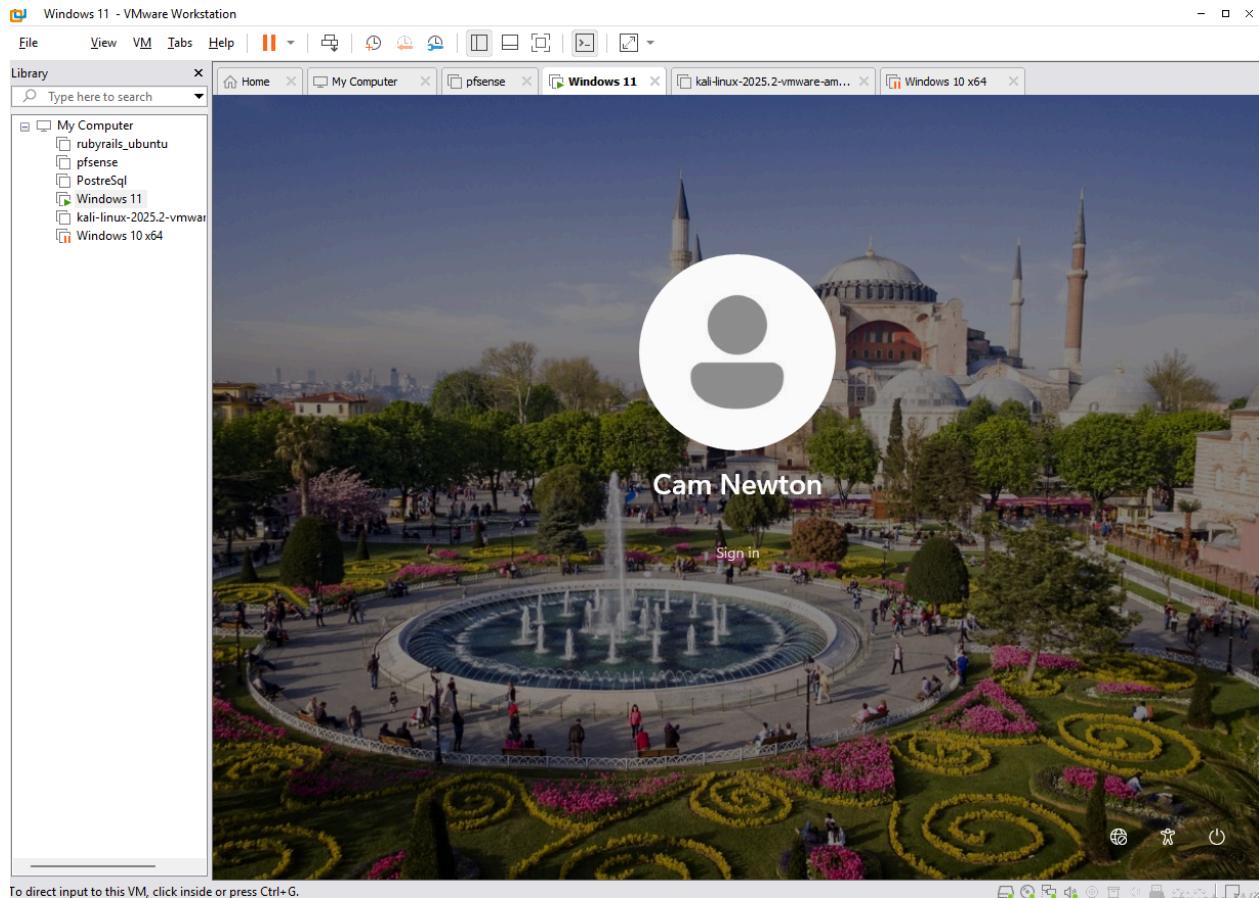


figure 8.3

9.windows10 → vmnet3

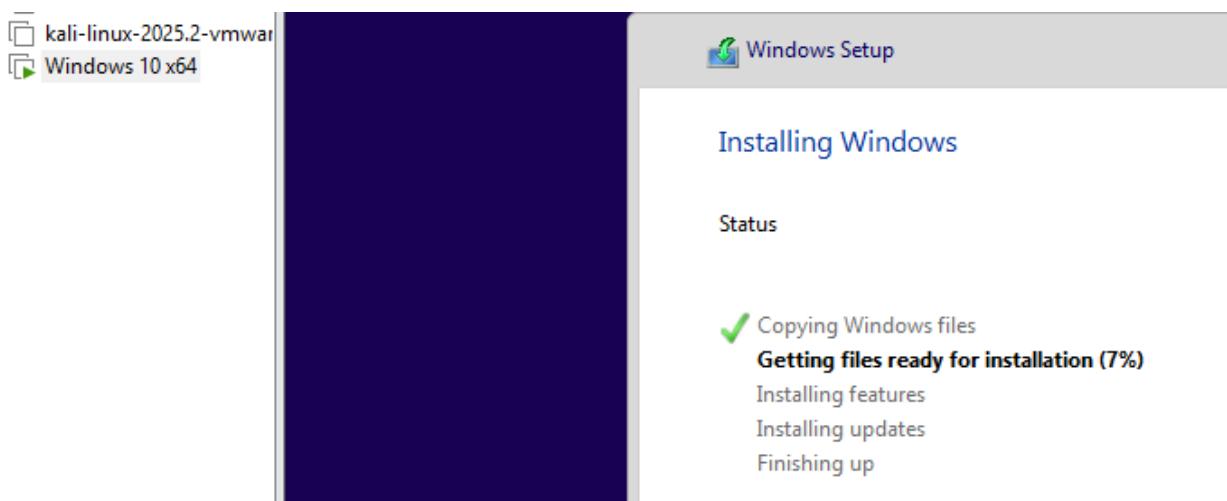


figure 9.1

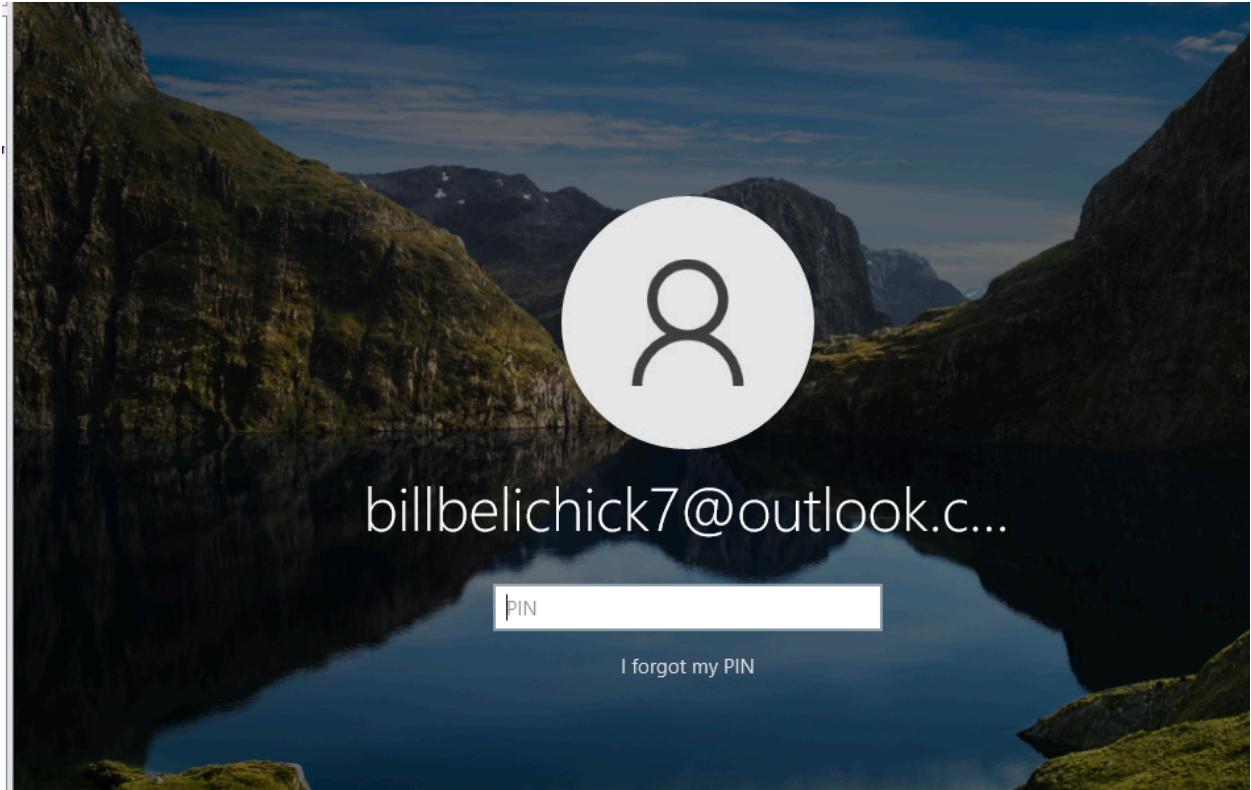


figure 9.2

10. kali-linux-attacker →bridged to outside(will connect to other vmnets after set up

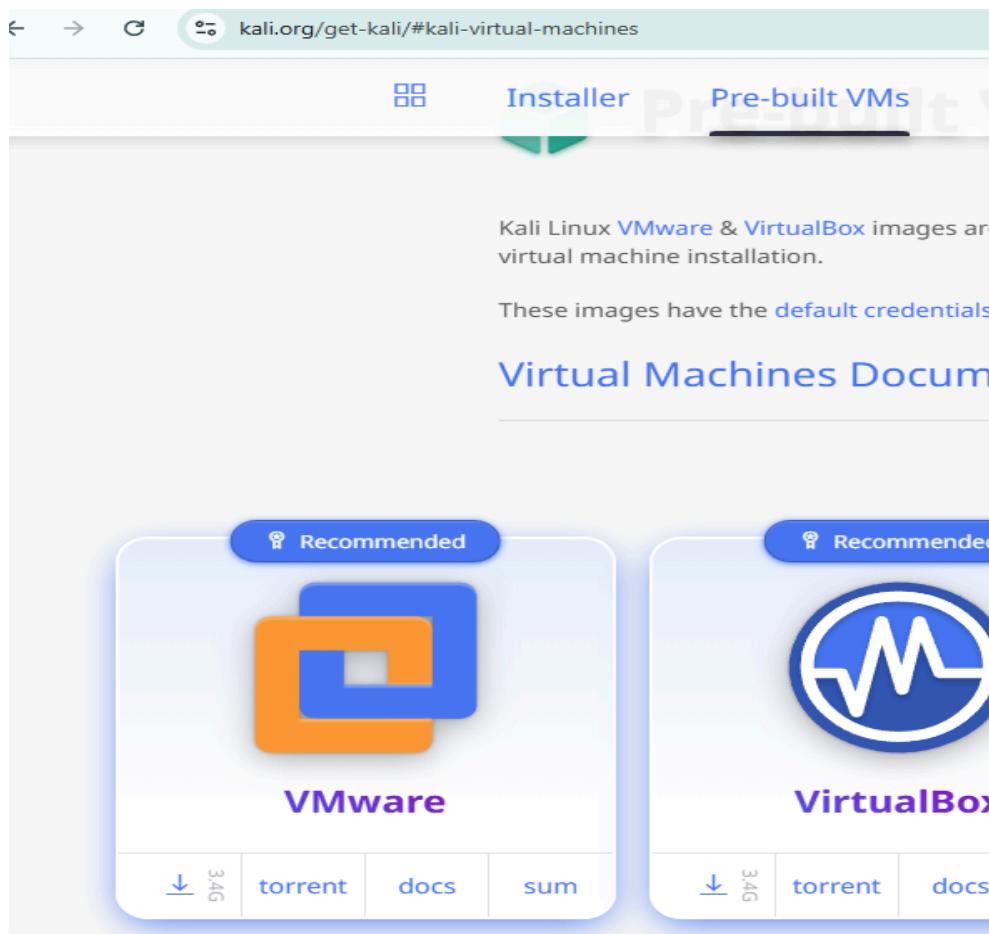


figure 10.1



figure 10.2

11. Boot pfSense, assign interfaces, and verify pfSense WebGUI access

Interfaces		
WAN	1000baseT <full-duplex>	192.168.102.129
LAN	1000baseT <full-duplex>	192.168.1.10
DATA	1000baseT <full-duplex>	172.16.1.10
WORKERS	1000baseT <full-duplex>	10.10.1.10

figure 11.1

12. Configure pfSense firewall rules with first being the NAT outbound rules so it can access internet:

Mappings										
	Interface	Source	Source Port	Destination	Destination Port	NAT Address	NAT Port	Static Port	Description	Actions
<input type="checkbox"/>	WAN	172.16.1.0/24	*	*	*	WAN address	*		NAT 4 Workers	
<input type="checkbox"/>	WAN	10.10.1.0/24	*	*	*	WAN address	*		NAT 4 Apps/labs	
<input type="checkbox"/>	WAN	192.168.1.0/24	*	*	*	WAN address	*		NAT 4 LAN	

Figure 12.1

13. Validate IP addressing and reachability: each VM can ping its gateway and pfSense LAN IP with firewall access. Doing this with firewall rules.

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
X 0/342 B	*	RFC 1918 networks	*	*	*	*	*		Block private networks	
X 0/0 B	*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks	
<input checked="" type="checkbox"/> 0/0 B	IPv4 *	*	*	*	*	*	none			

Figure 13.1

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓ 3/1.10 MIB	*	*	*	LAN Address	80	*	*		Anti-Lockout Rule	
✓ 3/94.62 MIB	IPv4 TCP	LAN subnets	*	*	*	*	none		allow all traffic	
✓ 16/256 KIB	IPv4 *	LAN subnets	*	*	*	*	none		Default allow LAN to any rule	
✓ 0/0 B	IPv6 *	LAN subnets	*	*	*	*	none		Default allow LAN IPv6 to any rule	

, figure 13.2

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/> 0/0 B	IPv4 *	DATA address	*	*	*	*	none		allow all traffic	

, figure 13.3

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/> 0/0 B	IPv4 TCP	WORKERS subnets	*	*	*	*	none		allow all traffic	

, figure 13.4

14. Install PostgreSQL on ubuntu-postgresql and create the application database and user. With these commands:

Running sudo apt update

```
sudo apt install -y build-essential libssl-dev zlib1g-dev libreadline-dev \
```

libyaml-dev libxml2-dev libxslt1-dev libcurl4-openssl-dev libffi-dev libncurses5-dev
 libgdbm-dev libdb-dev uuid-dev

Here, I enabled and started my PostgreSQL.

```
sql-jared@sql-jared-VMware-Virtual-Platform:~$ sudo systemctl enable postgresql
[sudo] password for sql-jared:
Synchronizing state of postgresql.service with SysV service script with /usr/lib
/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable postgresql
sql-jared@sql-jared-VMware-Virtual-Platform:~$ sudo systemctl start postgresql
sql-jared@sql-jared-VMware-Virtual-Platform:~$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; presen>
   Active: active (exited) since Tue 2025-10-21 19:05:23 EDT; 3min 45s ago
     Main PID: 1973 (code=exited, status=0/SUCCESS)
       CPU: 1ms

Oct 21 19:05:23 sql-jared-VMware-Virtual-Platform systemd[1]: Starting postgre>
Oct 21 19:05:23 sql-jared-VMware-Virtual-Platform systemd[1]: Finished postgre>
lines 1-8/8 (END)
```

figure 14.1

15. Now I entered my psql and set my databases, roles

`CREATE DATABASE myapp_development;`

`CREATE USER myapp_user WITH PASSWORD 'pass';`

`ALTER ROLE myapp_user SET client_encoding TO 'utf8';`

`ALTER ROLE myapp_user SET default_transaction_isolation TO 'read committed';`

`ALTER ROLE myapp_user SET timezone TO 'UTC';`

`GRANT ALL PRIVILEGES ON DATABASE myapp_development TO myapp_user;`

`GRANT ALL ON SCHEMA public TO testuser;`

`ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL ON TABLES TO testuser;`

```
postgres@sql-jared-VMware-Virtual-Platform:~$ psql
psql (16.10 (Ubuntu 16.10-0ubuntu0.24.04.1))
Type "help" for help.

postgres=# \du
              List of roles
   Role name   |          Attributes
-----+-----
    jared      | Create DB
myapp_user  |
    postgres   | Superuser, Create role, Create DB, Replication, Bypass RLS
   testuser   | Create DB

postgres=#
```

figure 15.1

16. Install Ruby, Rails, and app prerequisites on ubuntu-ruby-rails; deploying the Rails app.

```
jared@jared-VMware-Virtual-Platform:~$ ruby -v
ruby 3.2.2 (2023-03-30 revision e51014f9c0) [x86_64-linux]
jared@jared-VMware-Virtual-Platform:~$ bundler -v
Bundler version 2.4.10
jared@jared-VMware-Virtual-Platform:~$
```

figure 16.1

17. Configure Rails database.yml to point to ubuntu-postgresql and run migrations. I also tested to see if my Ruby Rails application was working.

cd ~testapp

Nano config/database.yml

```
GNU nano 7.2                                         conf
default: &default
  adapter: postgresql
  encoding: unicode
  pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>
  username: testuser
  password: pass
  host: 172.16.1.70
  port: 5432

  timeout: 5000

development:
<<: *default
database: testdb
```

figure 17.1

```
test:
<<: *default
database: testdb_test

# Store production database in the storage/ directory, which by default
# is mounted as a persistent Docker volume in config/deploy.yml.
production:
  primary:
    <<: *default
    database: testdb_test
  cache:
    <<: *default
    database: storage/production_cache.sqlite3
    migrations_paths: db/cache_migrate
  queue:
    <<: *default
    database: storage/production_queue.sqlite3
    migrations_paths: db/queue_migrate
  cable:
    <<: *default
    database: storage/production_cable.sqlite3
    migrations_paths: db/cable_migrate
```

figure 17.2

```
jared@jared-VMware-Virtual-Platform:~/testapp$ cd ~/testapp
jared@jared-VMware-Virtual-Platform:~/testapp$ rails db:create
Database 'testdb' already exists
Database 'testdb_test' already exists
jared@jared-VMware-Virtual-Platform:~/testapp$ rails db:migrate
jared@jared-VMware-Virtual-Platform:~/testapp$ rails server -b 172.16.1.60
=> Booting Puma
=> Rails 8.0.3 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 7.0.4 ("Romantic Warrior")
* Ruby version: ruby 3.2.2 (2023-03-30 revision e51014f9c0) [x86_64-linux]
* Min threads: 3
* Max threads: 3
* Environment: development
* PID: 3003
* Listening on http://172.16.1.60:3000
Use Ctrl-C to stop
Started GET "/" for 172.16.1.60 at 2025-10-23 11:32:23 -0400
Cannot render console from 172.16.1.60! Allowed networks: 127.0.0.0/127.255.255.255, ::1
ActiveRecord::SchemaMigration Load (0.8ms)  SELECT "schema_migrations"."version" FROM "schema_migrations" ORDER BY "schema_migrations"."version" ASC /*application='Testapp'*/
Processing by Rails::WelcomeController#index as HTML
  Rendering /home/jared/.rbenv/versions/3.2.2/lib/ruby/gems/3.2.0/gems/railties-8.0.3/lib/rails/templates/rails/welcome/index.html.erb
    Rendered /home/jared/.rbenv/versions/3.2.2/lib/ruby/gems/3.2.0/gems/railties-8.0.3/lib/rails/templates/rails/welcome/index.html.erb (Duration: 0.6ms | GC: 0.0ms)
Completed 200 OK in 33ms (Views: 4.3ms | ActiveRecord: 0.0ms (0 queries, 0 cached) | GC: 0.8ms
)
```

figure 17.3

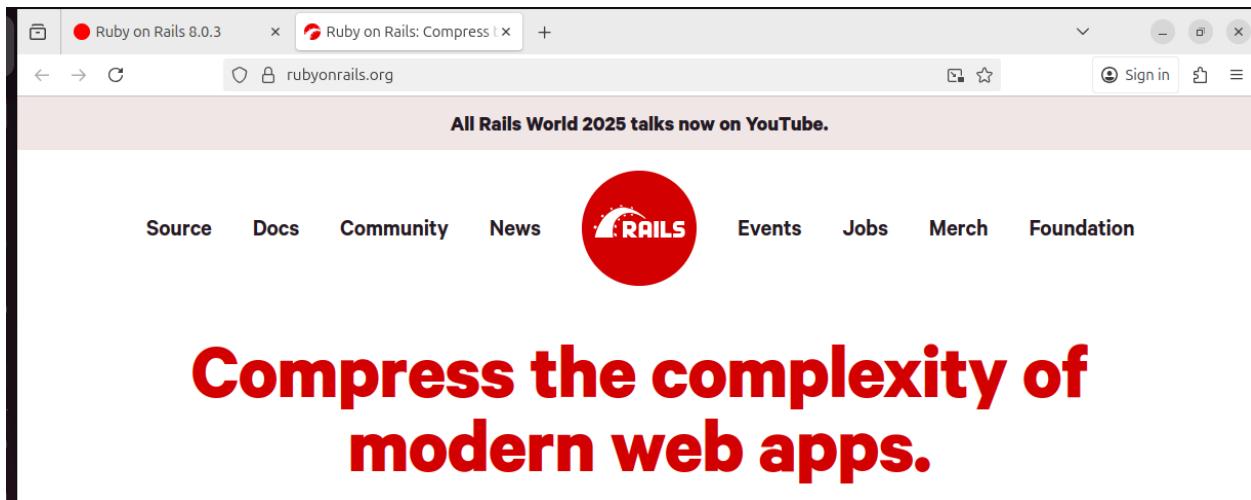


figure 17.4

18. Here, I integrated more of my PostgreSQL and Rails by setting up the logs to show on the Rails app. This gives it an interactive and cleaner look to the app and also see the entry logs.

```
nano log_entries_controller.rb
```

```
class LogEntriesController < ApplicationController
```

```
  def index
```

```
    @logs = LogEntry.all  end end
```

```
cd ..../views
```

```
mkdir log_entries
```

```
cd log_entries
```

```
nano index.html.erb
```

```
<h1>System Logs</h1><ul>  <% @logs.each do |log| %> <li><%= log.level %>: <%=>  
log.message %></li> <% end %></ul>
```

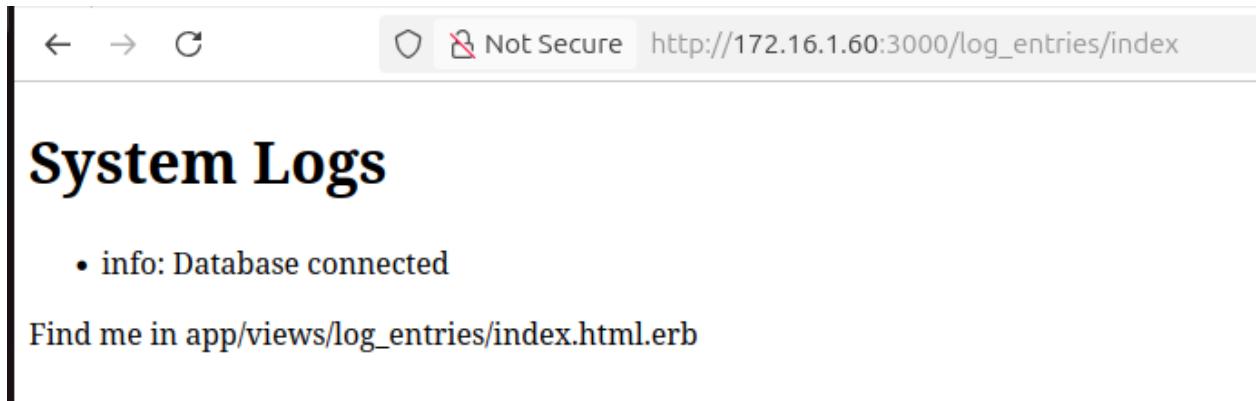


figure 18.1

```

Started GET "/log_entries/index" for 172.16.1.60 at 2025-10-23
12:18:38 -0400
Processing by LogEntriesController#index as HTML
  Rendering layout layouts/application.html.erb
  Rendering log_entries/index.html.erb within layouts/application
    [1m[36mLogEntry Load (2.2ms)[0m] [1m[34mSELECT
      "log_entries".* FROM "log_entries" /
    *action='index',application='Testapp',controller='log_entries'*/[0m
    [0m
    ↳ app/views/log_entries/index.html.erb:3
      Rendered log_entries/index.html.erb within layouts/application
      (Duration: 4.7ms | GC: 0.0ms)

```

figure 18.2

Made it look nicer

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>JARED'S Secure IT Dashboard</title>
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<style>
```

```
body { background-color:#0f172a; color:#f1f5f9; font-family: 'Segoe UI', sans-serif;
text-align:center; margin:0; padding:0; }
```

```
h1 { font-size:2.5rem; color:#38bdf8; margin-top:40px; }
```

```
.card { background:#1e293b; border-radius:10px; padding:25px; margin:20px auto;
width:80%; max-width:800px; box-shadow:0 0 20px rgba(0,0,0,0.4); }
```

```
.status { color:#10b981; font-weight:bold; }

.ipbox { font-family:monospace; color:#facc15; }

a { color:#38bdf8; text-decoration:none; }

a:hover { text-decoration:underline; }

</style>

</head>

<body>

<h1>Jared's IT Project Environment</h1>

<div class="card">

  <p><b>Ruby on Rails App:</b> <span class="ipbox">172.16.1.60</span> — <span
  class="status">Running</span></p>

  <p><b>PostgreSQL Database:</b> <span class="ipbox">172.16.1.70</span> — <span
  class="status">Connected</span></p>

  <p><b>SIEM / ELK Stack:</b> <span class="ipbox">192.168.1.50</span> — <span
  class="status">Monitoring</span></p>

</div>

<div class="card">

  <h2>System Logs</h2>
```

```

<p>View and verify database logs in real time:</p>

<a href="/log_entries">Open Logs Viewer →</a>

</div>

<footer style="margin-top:40px;font-size:0.9rem;color:#64748b;">

    &copy; 2025 Born Rich Operations (BRO) | Built by Jared Lowe

</footer>

</body>

</html>

```

The screenshot shows a web application running at <http://172.16.1.60>. The page has a dark blue background. At the top, it says "Welcome#index" and "Find me in app/views/welcome/index.html.erb". Below that is a section titled "Jared's IT Project Environment". This section contains three status cards:

- Ruby on Rails App:** 172.16.1.60 — Running
- PostgreSQL Database:** 172.16.1.70 — Connected
- SIEM / ELK Stack:** 192.168.1.50 — Monitoring

Below this is a "System Logs" section with the sub-instruction "View and verify database logs in real time:" followed by a link "Open Logs Viewer →". At the bottom of the page is a footer with the text "© 2025 Born Rich Operations (BRO) | Built by Jared Lowe" and a "Activate Windows" watermark.

```
<!DOCTYPE html>
```

```
<html>

<head>

<title>System Logs</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

    body { background-color:#0f172a; color:#f1f5f9; font-family: 'Segoe UI', sans-serif;
text-align:center; margin:0; padding:0; }

    h1 { color:#38bdf8; margin-top:30px; }

    table { width:90%; margin:30px auto; border-collapse:collapse; background:#1e293b;
border-radius:10px; overflow:hidden; }

    th, td { padding:12px 15px; border-bottom:1px solid #334155; }

    th { background:#334155; color:#38bdf8; }

    tr:hover { background:#1e40af; }

    .timestamp { color:#facc15; font-family:monospace; }

    a { color:#38bdf8; text-decoration:none; }

    a:hover { text-decoration:underline; }

</style>

</head>
```

```
<body>

<h1>System Logs</h1>

<table>

  <tr>

    <th>ID</th>

    <th>Level</th>

    <th>Message</th>

    <th>Created At</th>

  </tr>

<% @logs.each do |log| %>

  <tr>

    <td><%= log.id %></td>

    <td><%= log.level %></td>

    <td><%= log.message %></td>

    <td class="timestamp"><%= log.created_at.strftime("%Y-%m-%d %H:%M:%S") if log.created_at %></td>

  </tr>

<% end %>
```

</table>

<p>← Back to Dashboard</p>

</body>

</html>

ID	LEVEL	MESSAGE	CREATED AT
3	info	User viewed System Logs page	2025-10-30 02:06:22
2	info	User viewed System Logs page	2025-10-30 02:04:25
1	info	User viewed System Logs page	2025-10-30 01:58:40

```

GNU nano 7.2                               /home/jared/testapp/app/controllers/application_controller.rb
class ApplicationController < ActionController::Base
  # Only allow modern browsers supporting webp images, web push, badges, import maps, CSS nesting, and CSS :has.
  allow_browser_versions: :modern
end
def record_log(message, level = 'info')
  LogEntry.create(message: message, level: level)
rescue => e
  Rails.logger.error "Failed to record log: #{e.message}"
end

```

```

GNU nano 7.2                               /home/jared/testapp/app/controllers/log_entries_controller.rb
class LogEntriesController < ApplicationController
  def index
    record_log("User viewed System Logs page", "info")
    @logs = LogEntry.all.order(created_at: :desc)
  end

  def api
    render json: LogEntry.all
  end
end

```

I also added it to be readable in JSON for my future Siem tool(api), figure 18.1.1

19. Install Puma and nginx and create a systemd service for the Rails app.

```
jared@jared-VMware-Virtual-Platform:~/testapp$ bundle exec puma -v
Puma starting in single mode...
* Puma version: 7.0.4 ("Romantic Warrior")
* Ruby version: ruby 3.2.2 (2023-03-30 revision e51014f9c0) [x86_64-linux]
* Min threads: 3
* Max threads: 3
* Environment: development
* PID: 4510
DEPRECATION WARNING: The config.web_console.whitelisted_ips is deprecated and will be ignored in future release of web_console. Please use config.web_console.allowed_ips instead. (called from <top (required)> at /home/jared/testapp/config/environment.rb:5)
* Listening on http://0.0.0.0:3000
Use Ctrl-C to stop
```

```
jared@jared-VMware-Virtual-Platform: $ curl -kI http://172.16.1.60/
HTTP/1.1 200 OK
Server: nginx/1.24.0 (Ubuntu)
Date: Thu, 06 Nov 2025 00:15:00 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 0
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 0
X-Content-Type-Options: nosniff
X-Permitted-Cross-Domain-Policies: none
Referrer-Policy: strict-origin-when-cross-origin
Link: </assets/application-8b441ae0.css>; rel=preload; as=style; nopush
Vary: Accept
ETag: W/"7a19728ffa76036f4cc9c6431081209c"
Cache-Control: max-age=0, private, must-revalidate
Set-Cookie: _testapp_session=pRQzpVe8mMek%2FgqoJK1HE0uOX9CtnXYECL%2BBy5H6wOxy8rzMmutpprR%2B0sglGgEvv%2BZoMVCsfJZskdIKhv2ZrZtGSVwE%2FyuV7n8tEvjkkb30gUfLRku%2Btk4iRXg0Be50MzqjLPs19hLa4NR6L0d1ounK4GeBrxL0jK68TvG5cV9JBb9Yjbbq10Su0rq9J%2B1QgPDr8ce%2BbFrY9PnUrk2B2Wku%2FglR5CDNIkUfqZiPC6GtSekGY%2BiwCwuMKuhIOnLDwqrGMzvSue5%2BARX1bYfhgr7SMV11Mn--kLh2Bjx2xEsV0Tq2--0FaZl3m10Vm5NtkkEF0Tw%3D%3D; path=/; secure; httponly; samesite=lax
X-Request-ID: 0f1d3e0-602e-46db-aa02-50edb48804c
X-Runtime: 0.351311
Strict-Transport-Security: max-age=63072000; includeSubDomains
```

Activate Windows
 Go to Settings to activate Windows.

figure 19.1

20. Add Elastic repository & installed Elasticsearch

```
jared-siemelk@jared-siemelk-Virtual-Platform:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:2 https://artifacts.elastic.co/packages/8.x/apt stable InRelease [3,248 B]
Get:3 https://artifacts.elastic.co/packages/8.x/apt stable/main amd64 Packages [93.8 kB]
Hit:4 http://us.archive.ubuntu.com/ubuntu noble InRelease
Hit:5 http://us.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:6 http://us.archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 97.1 kB in 1s (89.9 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
jared-siemelk@jared-siemelk-Virtual-Platform:~$ sudo apt install elasticsearch -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 666 MB of archives.
After this operation, 1,285 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/8.x/apt stable/main amd64 elasticsearch amd64 8.19.6 [666 MB]
27% [elasticsearch 226 MB/666 MB 34%] 18.4 MB/s 23s
```

figure 20.1

`sudo apt update`

`sudo apt install wget curl gnupg -y`

Imports a GPG key and saved it:

```
curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o
/usr/share/keyrings/elasticsearch-keyring.gpg
```

Adds a repository:

```
echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg]
https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee
/etc/apt/sources.list.d/elastic-8.x.list
```

`sudo apt install elasticsearch -y`

Then I configure it:

Edit /etc/elasticsearch/elasticsearch.yml:

network.host: 192.168.1.50

http.port: 9200

discovery.type: single-node

Then enable & start:

sudo systemctl daemon-reload

sudo systemctl enable elasticsearch

sudo systemctl start elasticsearch

Test:

curl http://192.168.1.50:9200

```

Password for the [elastic] user successfully reset.
New value: VjMX=0=ZAeuMW4JXSyc
jared-siemelk@jared-siemelk-VMware-Virtual-Platform:~$ ^C
jared-siemelk@jared-siemelk-VMware-Virtual-Platform:~$ curl -u elastic https://192.168.1.50:9200 -k
Enter host password for user 'elastic':
{
  "name" : "jared-siemelk-VMware-Virtual-Platform",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "t-cYdFg3RcertYol87-ehg",
  "version" : {
    "number" : "8.19.6",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "d2c42d91a1eb9e14b1a37c4d87eb2533ec859e2b",
    "build_date" : "2025-10-21T22:05:27.062491219Z",
    "build_snapshot" : false,
    "lucene_version" : "9.12.2",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
jared-siemelk@jared-siemelk-VMware-Virtual-Platform:~$ █

```

figure 20.2

```

jared-siemelk@jared-siemelk-VMware-Virtual-Platform:~$ sudo systemctl status elasticsearch --no-pager
sudo systemctl status kibana --no-pager
● elasticsearch.service - Elasticsearch
  Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; preset: enabled)
  Drop-In: /etc/systemd/system/elasticsearch.service.d
    └─override.conf
  Active: active (running) since Mon 2025-11-03 17:45:53 EST; 54min ago
    Docs: https://www.elastic.co
  Main PID: 1604 (java)
    Tasks: 97 (limit: 9374)
   Memory: 2.7G (peak: 2.8G)
      CPU: 3min 30.970s
  CGroup: /system.slice/elasticsearch.service
          ├─1604 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=server -Dcli.script...
          ├─2272 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.ne...
          ├─2303 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller
Nov 03 17:44:30 jared-siemelk-VMware-Virtual-Platform systemd[1]: Starting elasticsearch.service - Elasticsearch...
Nov 03 17:44:30 jared-siemelk-VMware-Virtual-Platform systemd-entrypoint[1604]: warning: ignoring JAVA_HOME=/usr/sh... JDK
Nov 03 17:45:53 jared-siemelk-VMware-Virtual-Platform systemd[1]: Started elasticsearch.service - Elasticsearch.

```

figure 20.3

```

jared-siemelk@jared-siemelk-VMware-Virtual-Platform:~$ sudo nano /etc/kibana.yml
jared-siemelk@jared-siemelk-VMware-Virtual-Platform:~$ sudo nano /etc/kibana/kibana.yml
jared-siemelk@jared-siemelk-VMware-Virtual-Platform:~$ sudo systemctl enable kibana
Created symlink /etc/systemd/system/multi-user.target.wants/kibana.service → /usr/lib/systemd/system/kibana.service.
jared-siemelk@jared-siemelk-VMware-Virtual-Platform:~$ sudo systemctl start kibana
jared-siemelk@jared-siemelk-VMware-Virtual-Platform:~$ sudo systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/usr/lib/systemd/system/kibana.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-10-28 21:13:12 EDT; 6s ago
       Docs: https://www.elastic.co
      Main PID: 13098 (node)
        Tasks: 11 (limit: 4543)
       Memory: 289.6M (peak: 291.2M)
          CPU: 7.042s
         CGroup: /system.slice/kibana.service
                   └─13098 /usr/share/kibana/bin/../node/glibc-217/bin/node /usr/share/kibana/bin/../src/cli/dist

Oct 28 21:13:12 jared-siemelk-VMware-Virtual-Platform systemd[1]: Started kibana.service - Kibana.
Oct 28 21:13:12 jared-siemelk-VMware-Virtual-Platform kibana[13098]: Kibana is currently running with legacy OpenSSL pr...
Oct 28 21:13:13 jared-siemelk-VMware-Virtual-Platform kibana[13098]: {"log.level":"info","@timestamp":...
Oct 28 21:13:13 jared-siemelk-VMware-Virtual-Platform kibana[13098]: Native global console methods have been overridden...
Oct 28 21:13:15 jared-siemelk-VMware-Virtual-Platform kibana[13098]: [2025-10-28T21:13:15.334-04:00][INFO ][root] Kibana...
Oct 28 21:13:15 jared-siemelk-VMware-Virtual-Platform kibana[13098]: [2025-10-28T21:13:15.367-04:00][INFO ][node] Kibana...
lines 1-17/17 (END)

```

figure 20.4

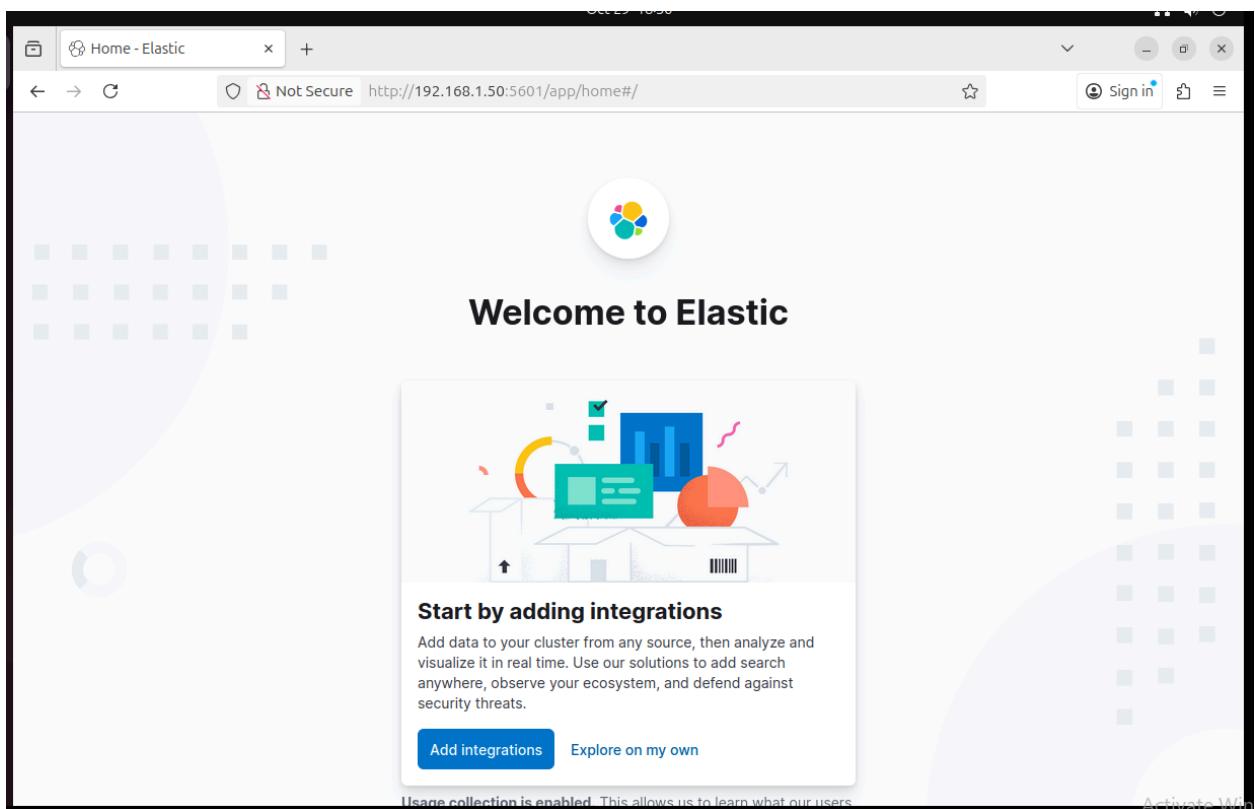
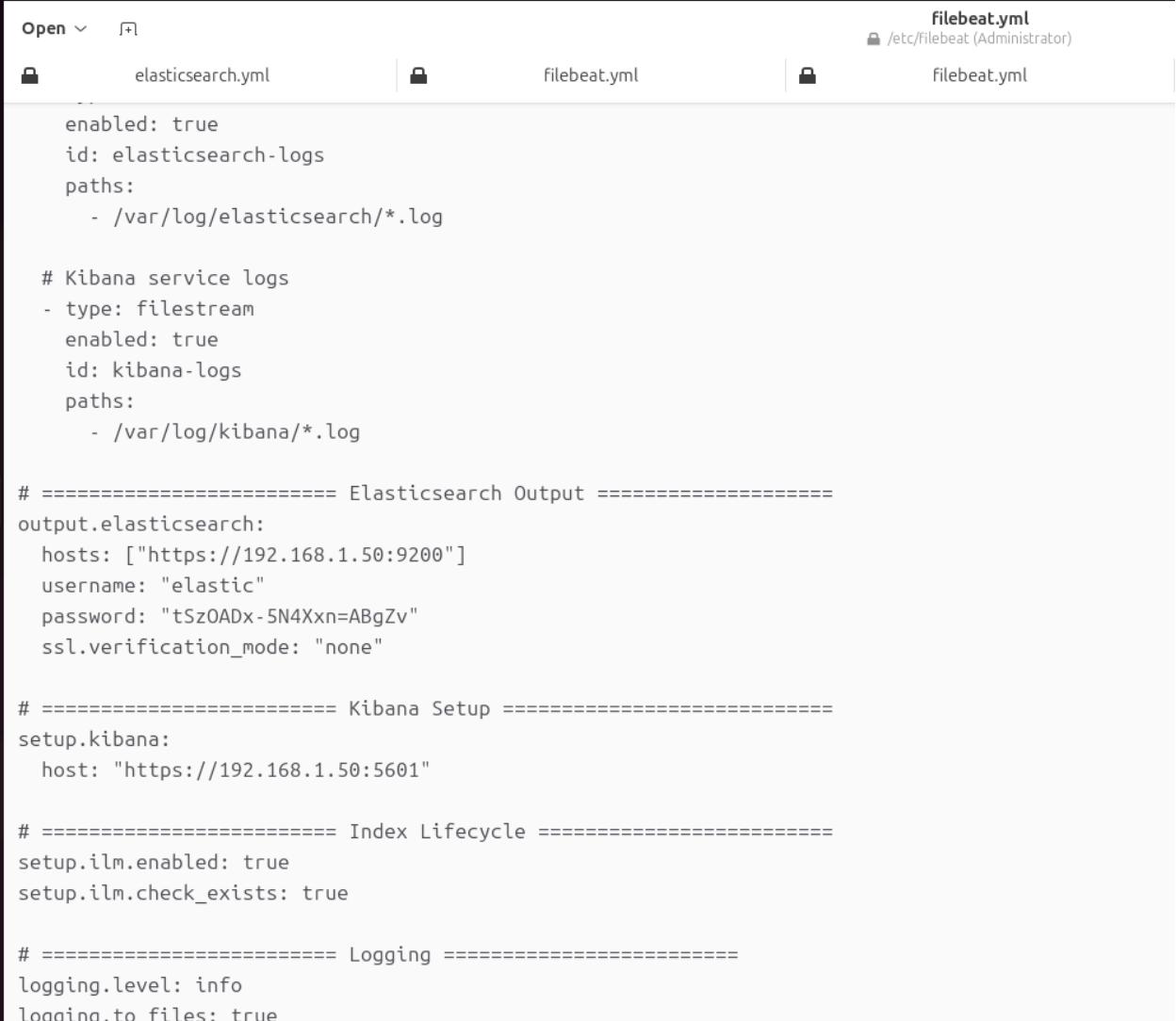


figure 20.5

21. Now since we got Elastic on and working I set all vm to use filebeat to forward logs to elastic so we can create alerts. Filebeat will read PostgreSQL log files and forward them securely to

Elasticsearch.



```
Open ▾ + elasticsearch.yml filebeat.yml filebeat.yml
filebeat.yml
/etc/filebeat (Administrator)

enabled: true
id: elasticsearch-logs
paths:
- /var/log/elasticsearch/*.log

# Kibana service logs
- type: filestream
  enabled: true
  id: kibana-logs
  paths:
    - /var/log/kibana/*.log

# ===== Elasticsearch Output =====
output.elasticsearch:
  hosts: ["https://192.168.1.50:9200"]
  username: "elastic"
  password: "tSz0ADx-5N4Xxn=ABgZv"
  ssl.verification_mode: "none"

# ===== Kibana Setup =====
setup.kibana:
  host: "https://192.168.1.50:5601"

# ===== Index Lifecycle =====
setup.ilm.enabled: true
setup.ilm.check_exists: true

# ===== Logging =====
logging.level: info
logging.to_files: true
```

figure 21.1

Here we have the filebeats.yml for ruby rails to show that it is working and sending.

```
GNU nano 7.2                               /etc/filebeat/filebeat.yml

# ===== Filebeat inputs =====
filebeat.inputs:
- type: filestream
  id: rails_app_log
  enabled: true
  paths:
    - /home/jared/testapp/log/production.log
    - /home/jared/testapp/log/development.log
  parsers:
    - multiline:
        type: pattern
        pattern: '^\\[[A-Z][a-z]{2}\\] .+\\$|^Started .+|^Processing by .+|^Completed .+'
        negate: true
        match: after
  fields:
    app: ruby_on_rails
    env: production
  fields_under_root: true

- type: filestream
  id: puma_log
  enabled: true
  paths:
    - /home/jared/testapp/log/puma.stdout.log
    - /home/jared/testapp/log/puma.stderr.log
  fields:
    app: puma
    env: production
  fields_under_root: true

# Exclude lines. A list of regular expressions to match. It drops the lines that are
# matching any regular expression from the list.
# Line filtering happens after the parsers pipeline. If you would like to filter lines
```

figure 21.2

Permissions was a big issue so here on each machine for simplicity:

sudo chown root:filebeat /etc/filebeat/filebeat.yml

sudo chmod 640 /etc/filebeat/filebeat.yml

sudo chown -R filebeat:filebeat /var/log/filebeat

Enable detailed SQL activity logging for ingestion.

```

GNU nano 7.2
max_connections = 100
unix_socket_directories = '/var/run/postgresql'

#-----#
# SSL SETTINGS
#-----#
ssl = off
ssl_cert_file = '/etc/ssl/certs/ssl-cert-snakeoil.pem'
ssl_key_file = '/etc/ssl/private/ssl-cert-snakeoil.key'

#-----#
# MEMORY AND PERFORMANCE
#-----#
shared_buffers = 128MB
dynamic_shared_memory_type = posix

#-----#
# WRITE-AHEAD LOG
#-----#
max_wal_size = 1GB
min_wal_size = 80MB

#-----#
# LOGGING AND REPORTING - REQUIRED FOR FILEBEAT
#-----#
logging_collector = on
log_directory = '/var/log/postgresql'
log_filename = 'postgresql-16-main.log'
log_statement = 'all'
log_connections = on
log_disconnections = on
log_line_prefix = '%m [%p] %u@%d '

```

figure 21.2

After Filebeat starts sending logs:

```

sudo filebeat setup --dashboards \
-E setup.kibana.host=http://192.168.1.50:5601 \
-E output.elasticsearch.username=elastic \
-E output.elasticsearch.password='tSzOADx-5N4Xxn=ABgZv'

```

Also for simplicity I added all pfSense logs to be forwarded to Kibana where we can read them there.

Remote Logging Options

Enable Remote Logging Send log messages to remote syslog server

Source Address

This option will allow the logging daemon to bind to a single IP address, rather than all IP addresses. If a single IP is picked, remote syslog servers must all be of that IP type. To mix IPv4 and IPv6 remote syslog servers, bind to all interfaces.

NOTE: If an IP address cannot be located on the chosen interface, the daemon will bind to all addresses.

IP Protocol

This option is only used when a non-default address is chosen as the source above. This option only expresses a preference; if an IP address of the selected type is not found on the chosen interface, the other type will be tried.

Remote log servers

Remote Syslog Contents

- Everything
- System Events
- Firewall Events
- DNS Events (Resolver/unbound, Forwarder/dnsmasq, filterdns)
- DHCP Events (DHCP Daemon, DHCP Relay, DHCP Client)
- PPP Events (PPPoE WAN Client, L2TP WAN Client, PPTP WAN Client)
- General Authentication Events
- Captive Portal Events
- VPN Events (IPsec, OpenVPN, L2TP, PPPoE Server)
- Gateway Monitor Events
- Routing Daemon Events (RADVD, UPnP, RIP, OSPF, BGP)
- Network Time Protocol Events (NTP Daemon, NTP Client)
- Wireless Events (hostapd)

Syslog sends UDP datagrams to port 514 on the specified remote syslog server, unless another port is specified. Be sure to set syslogd on the remote server to accept syslog messages from pfSense.

figure 21.3

22. Now that I have all vms to send logs with filebeat I now will take them with kibana and elastic to show on their site to have alerts and read the logs.

This loads prebuilt dashboards such as; which is excellent for this project:

[Filebeat PostgreSQL] Database Logs Overview

[Filebeat PostgreSQL] Query Duration Overview ECS

The screenshot shows the 'Dashboards' section of the Elastic Stack interface. At the top, there's a search bar and filters for 'Recently updated', 'Tags', and 'Created by'. A blue button on the right says 'Create dashboard'. Below this, there are two tabs: 'All' (selected) and 'Starred'. A table lists various dashboards with columns for name, last updated time, and actions (edit, delete). The dashboards listed include:

- [Filebeat Suricata] Events Overview
- [Filebeat Suricata] Alert Overview
- [Filebeat Netflow] Top-N Flows
- [Filebeat Netflow] Traffic Analysis
- [Filebeat Netflow] Overview
- [Filebeat Netflow] Geo Location
- [Filebeat Netflow] Flow records
- [Filebeat Netflow] Flow Exporters
- [Filebeat Netflow] Conversation Partners
- [Filebeat Netflow] Autonomous Systems
- [Filebeat icinga] Main Log ECS

At the bottom right, there's a message: 'Activate Windows Go to Settings to activate Windows'.

figure 22.1

We will use the vms credentials such as apps and ips to filter machine activity that I would like to see which will help for alerting.

connection authorized: user=postgres

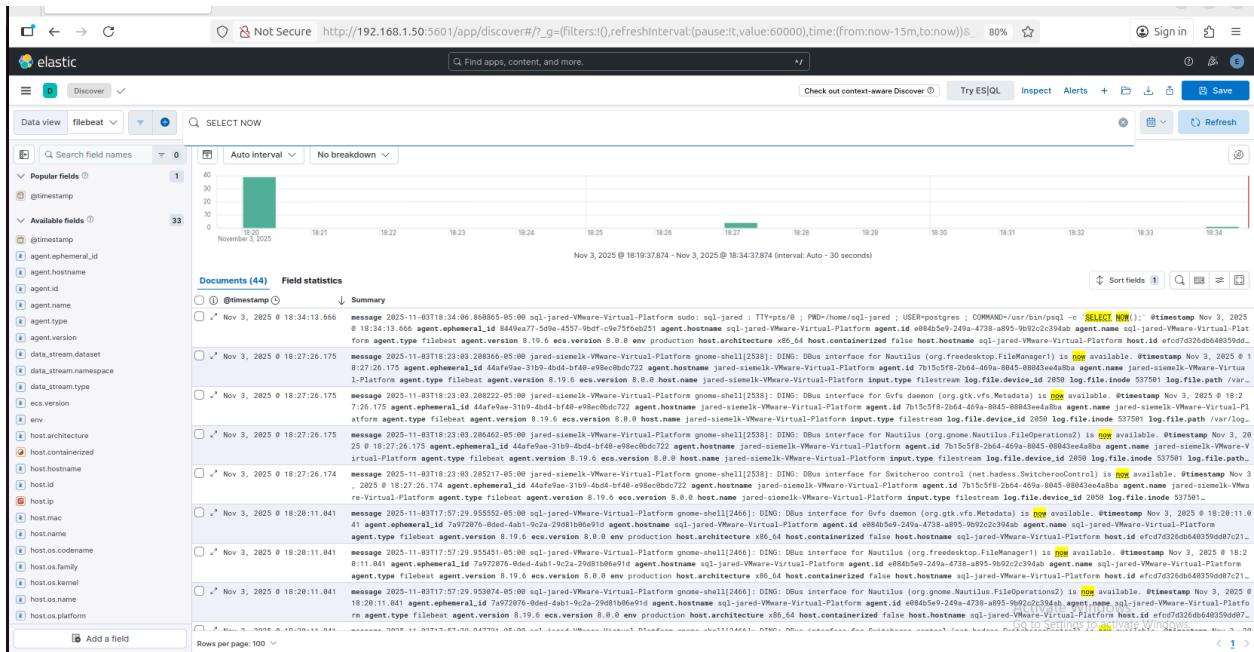


figure 22.2

Example log entries observed in Kibana making sure my ruby appears and right host machine:

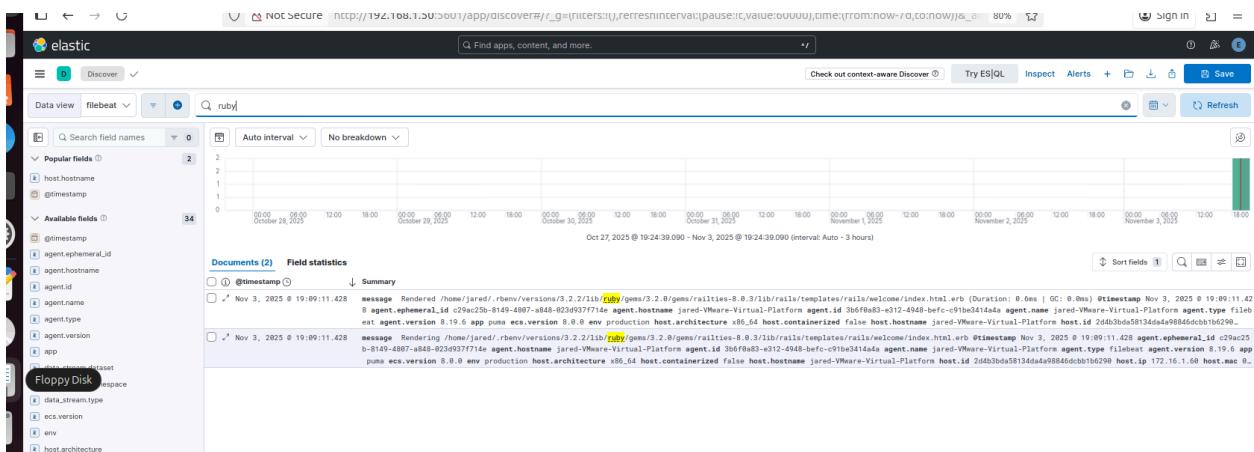


figure 22.3

Now that we have them all working now we set up dashboards in kibana that where we are able to see any errors, logs, and activity on our virtual machines(database and site)

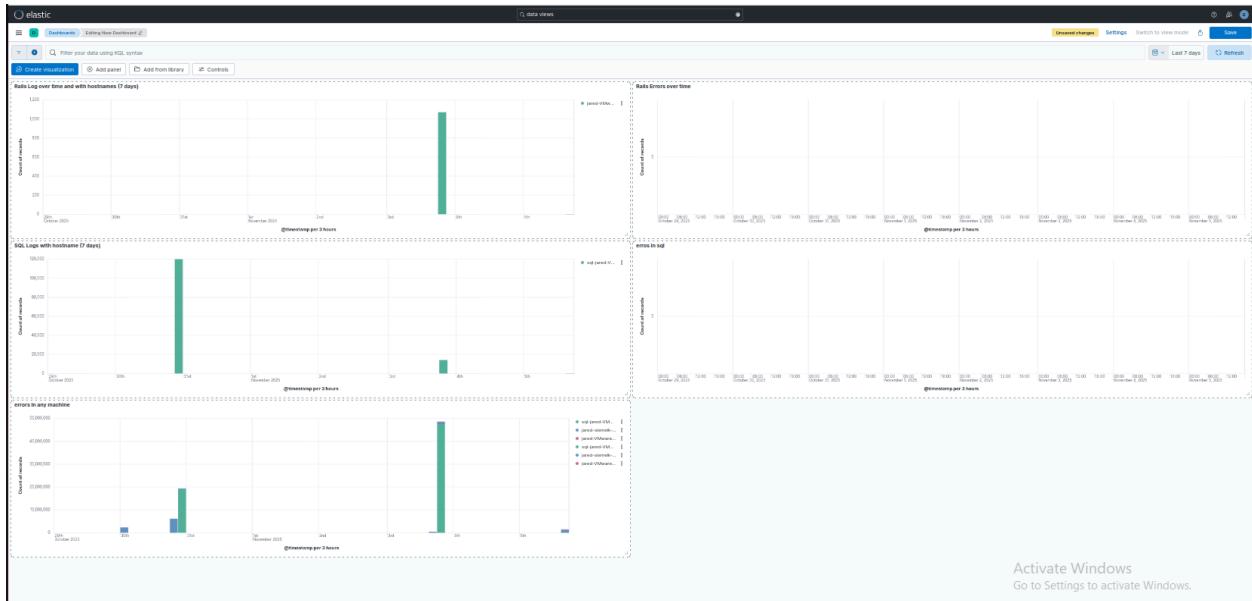


figure 22.4

Here I tested with logging into the db in rails on the site multiple items and then i sent a test db to the test database in psql and got activity!

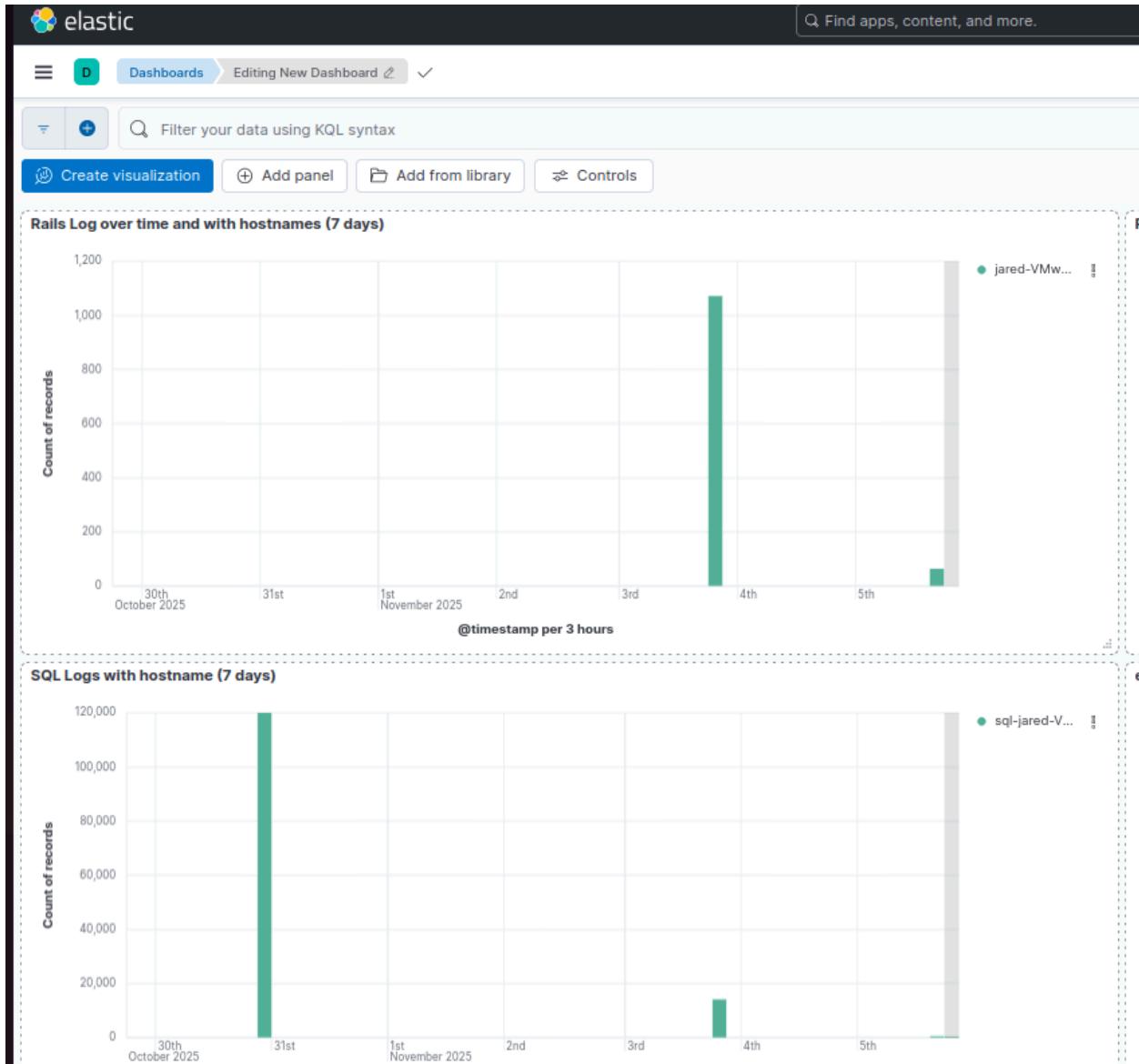


figure 22.5

23. Testing the rules and alerts for the machines to make sure logs are coming in on time and working.

With our first rule/alert any ruby rails that gives out a bad http response code will get alerted:

The screenshot shows the 'Rule definition' section of the Elasticsearch interface. It includes fields for defining a KQL or Lucene query, setting a data view, defining a query, specifying a group, threshold, and time window, setting the number of documents to send, and adding fields to alert details. A 'Test query' button is present, along with a message indicating 0 matches found in the last 7 days. Activation and settings options are also visible.

Elasticsearch query Alert when matches are found during the latest query run. [View documentation](#)

KQL or Lucene
Use KQL or Lucene to define a text-based query.

Select a data view
DATA VIEW filebeat

Define your query
app: nginx OR app: puma AND http.response.status_code >= 500

Set the group, threshold, and time window
WHEN count()
OVER all documents
IS ABOVE 1000
FOR THE LAST 7 days

Set the number of documents to send
SIZE 100

Exclude matches from previous runs

Add more fields to alert details
container.id host.hostname host.id host.name kubernetes.pod.uid

Test query **Copy query**

Query matched 0 documents in the last 7d.

Activate Go to Setting

figure 23.1

```
Nov 05 19:18:50 jared-VMware-Virtual-Platform systemd[1]: Started filebeat.service - Filebeat sends log files to Logstash
jared@jared-VMware-Virtual-Platform:~$ for i in {1..15}; do curl -sk https://172.16.1.60/doesnotexist >/dev/null; done
```

figure 23.1.1

The screenshot shows the Elasticsearch Alerting interface. At the top, it says "ALERT – Web Ruby Spike". Below that, it indicates the Type is "Elasticsearch query" and the API key owner is "elastic".

Rule details:

- Rule is:** Enabled
- Last response:** Succeeded (5 minutes ago)
- Notify when alerts generated:** Enabled

Definition:

- Rule type:** Elasticsearch query
- Description:** Alerts for Web Ruby Spike
- Runs every:** 5 minutes
- Actions:** No actions defined

At the bottom, there are tabs for "Alerts" (which is selected) and "History".

figure 23.1.2

When postgresql gets a bad login testing alerts and rules

The screenshot shows the Elasticsearch Data View interface for defining an alert rule. The title is "DATA VIEW filebeat".

Define your query:

```
message: postgres OR message: postgresql AND message: FATAL
```

Set the group, threshold, and time window:

```
WHEN count() OVER all documents IS ABOVE 1000 FOR THE LAST 5 days
```

Set the number of documents to send:

```
SIZE 100
```

Exclude matches from previous runs

Add more fields to alert details:

- container.id
- host.hostname
- host.id
- host.name
- kubernetes.pod.uid

Buttons at the bottom:

- ▶ Test query
- 📋 Copy query

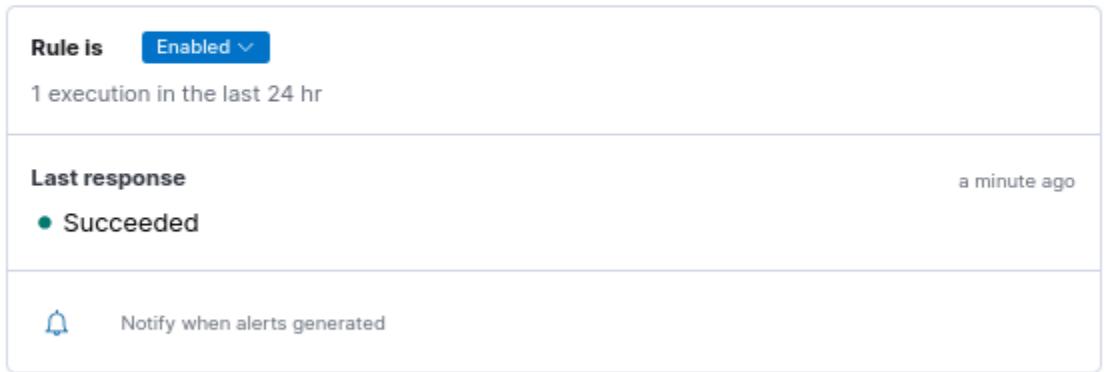
figure 23.2

```
sql-jared@sql-jared-VMware-Virtual-Platform:~$ PGPASSWORD=badpass psql -h 172.16.1.70 -U wrong -d postgres -c "SELECT 1;" || true  
psql: error: connection to server at "172.16.1.70", port 5432 failed: FATAL:  password authentication failed for user "wrong"  
sql-jared@sql-jared-VMware-Virtual-Platform:~$ █
```

figure 23.2.1

ALERT – Postgres Failed Logins

Type [Elasticsearch query](#) API key owner **elastic**



[Alerts](#) [History](#)

figure 23.2.2

24. Now I will add the pihole DNS and configure it for pfsense to use.

```
curl -sSL https://install.pi-hole.net | bash
```

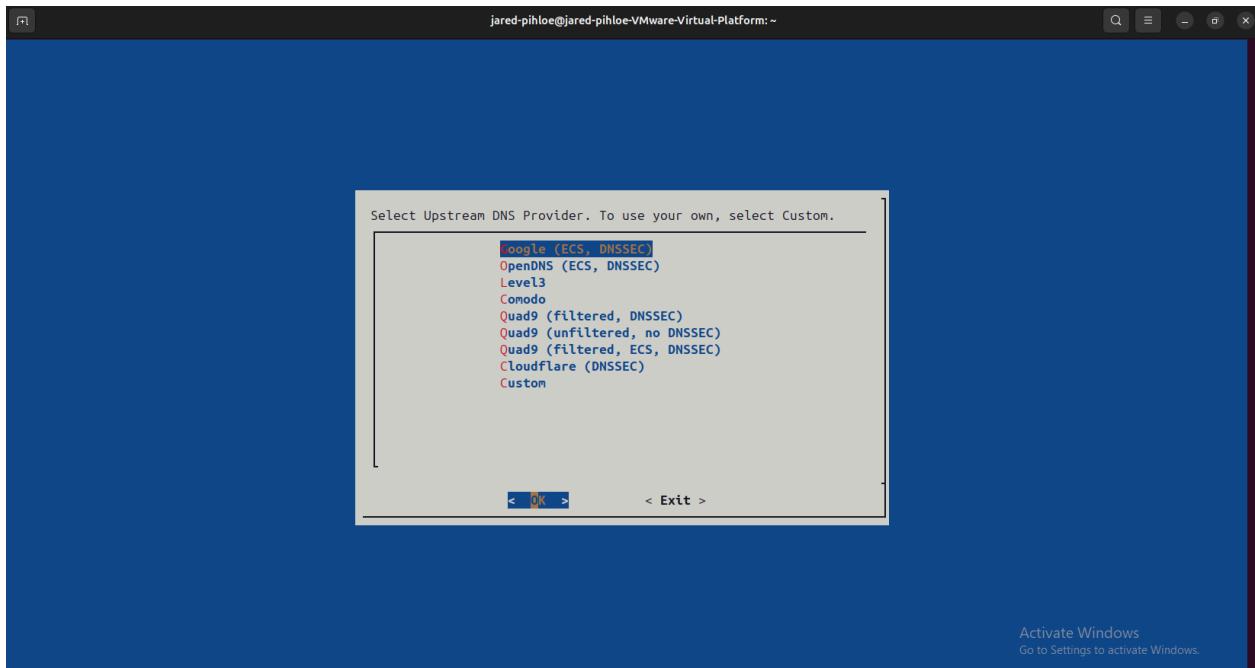


figure 24.1

Here I finished the installation and now we can move to adding it to pfSense.

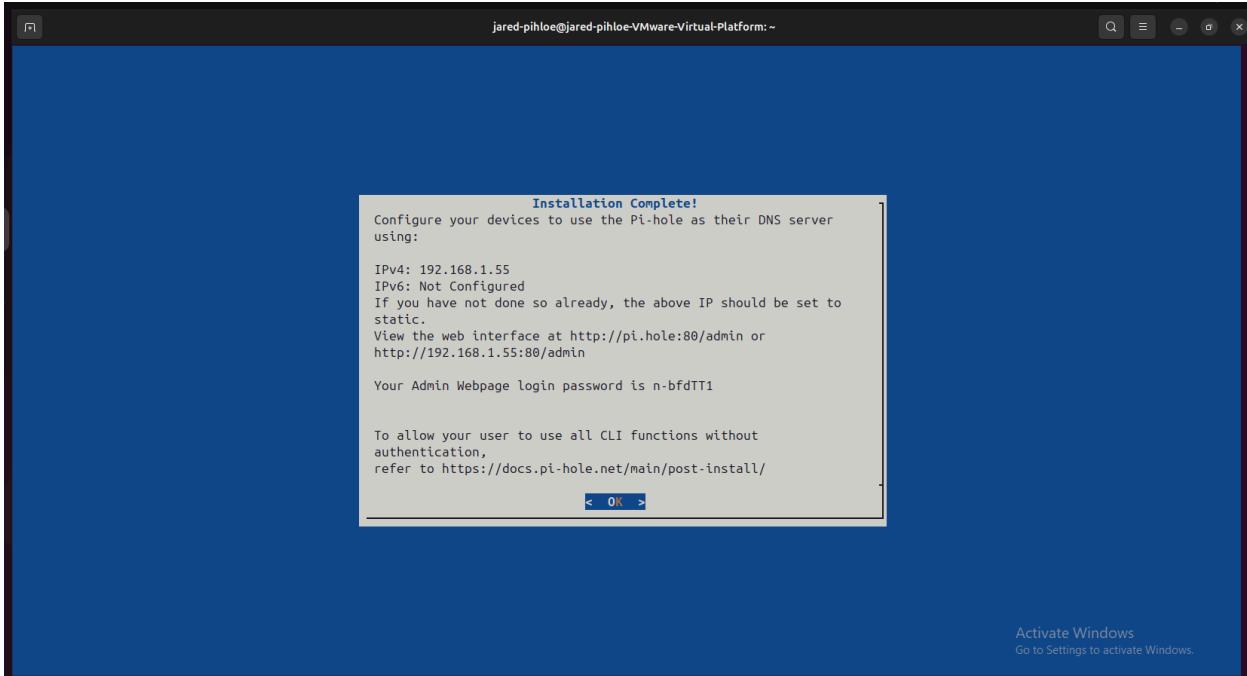


figure 24.2

Here I set it the DNS on the LAN to the ip for the pihole(I did it for each vnet/interface)

The screenshot shows a configuration interface for a DHCP server. The main sections include:

- Enable:** A checked checkbox labeled "Enable DHCP server on LAN interface".
- Deny Unknown Clients:** A dropdown menu set to "Allow all clients". A note explains that this allows any DHCP client to get an IP address from this scope.
- Ignore Client Identifiers:** An unchecked checkbox for "Do not record a unique identifier (UID) in client lease data if present in the client DHCP request". A note states that this is useful for dual-boot clients sharing the same MAC address.
- DNS Registration:** Set to "Track server". A note says it overrides the default DNS registration policy.
- Early DNS Registration:** Set to "Track server". A note says it overrides the default early DNS registration policy.
- Primary Address Pool:**
 - Subnet:** 192.168.1.0/24
 - Subnet Range:** 192.168.1.1 - 192.168.1.254
 - Address Pool Range:** From 192.168.1.50 To 192.168.1.100. A note states that this range must not overlap with other pools.
 - Additional Pools:** A green button labeled "+ Add Address Pool". A note says it's for additional pools outside the main range.
- Server Options:**
 - WINS Servers:** A field containing "WINS Server 1".
 - DNS Servers:** A field containing "192.168.1.55". Below it are fields for "DNS Server 2", "DNS Server 3", and "DNS Server 4".
- Other DHCP Options:**
 - Gateway:** A field containing "192.168.1.10". A note says it specifies an alternate gateway if the default is incorrect.

figure 24.2

25.Now I can start using the kali machine to attack the network.

I allowed the kali to be connected and take all vmnets so I can attack machines all at once and use this here to see the ips that are attacking using Kibana.

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:ec:d5:0e brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.51/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
            valid_lft 5321sec preferred_lft 5321sec
        inet6 fe80::28c3:1cff:841e:63e6/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:ec:d5:22 brd ff:ff:ff:ff:ff:ff
        inet 172.16.1.128/24 brd 172.16.1.255 scope global dynamic noprefixroute eth1
            valid_lft 1689sec preferred_lft 1689sec
        inet6 fe80::a30c:c8f:4800:c199/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:ec:d5:18 brd ff:ff:ff:ff:ff:ff
        inet 10.10.1.50/24 brd 10.10.1.255 scope global dynamic noprefixroute eth2
            valid_lft 5321sec preferred_lft 5321sec
        inet6 fe80::cd6d:48b4:cbb1:8d11/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
(kali㉿kali)-[~]
$
```

figure 25.1

I then installed metasploit and gobuster where i will use them to attack.


```
(kali㉿kali)-[~]
$ gobuster dir \
-u http://172.16.1.60:3000 \
-w /usr/share/wordlists/dirb/common.txt

Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

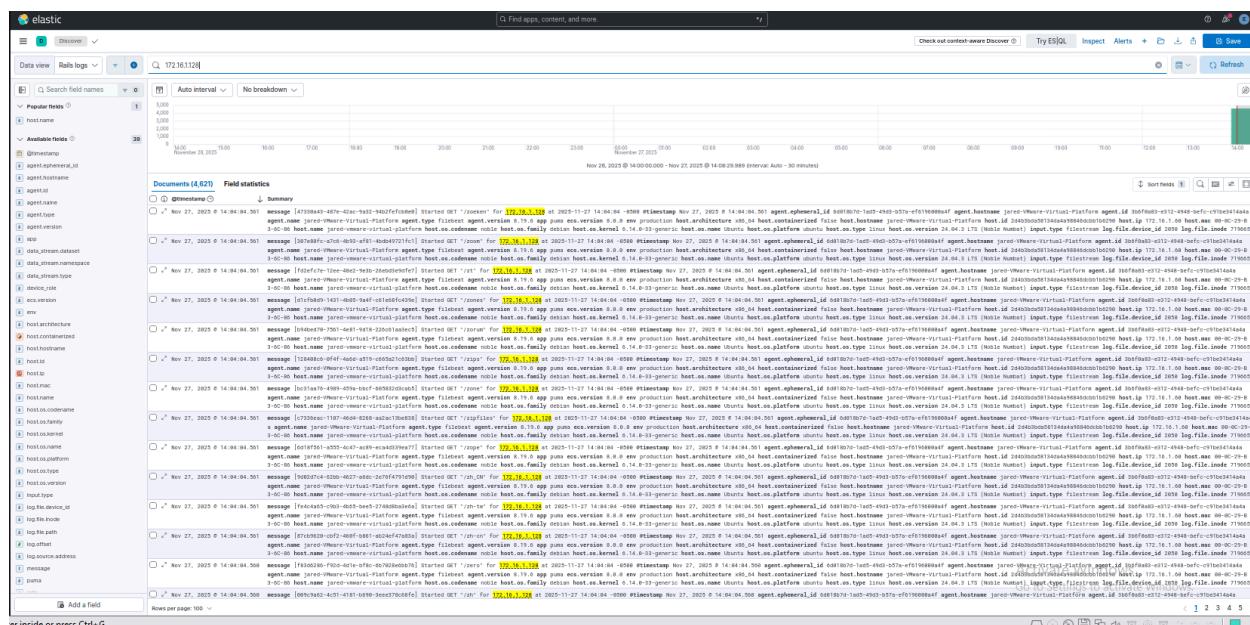
[+] Url:          http://172.16.1.60:3000
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.8
[+] Timeout:      10s

Starting gobuster in directory enumeration mode

/404                           (Status: 200) [Size: 4836]
/500                           (Status: 200) [Size: 7918]
/400                           (Status: 200) [Size: 6685]
/robots.txt                    (Status: 200) [Size: 99]
/up                            (Status: 200) [Size: 73]
Progress: 4613 / 4613 (100.00%)

Finished
```

Here I use gobuster scan on the machine to discover several accessibilities.

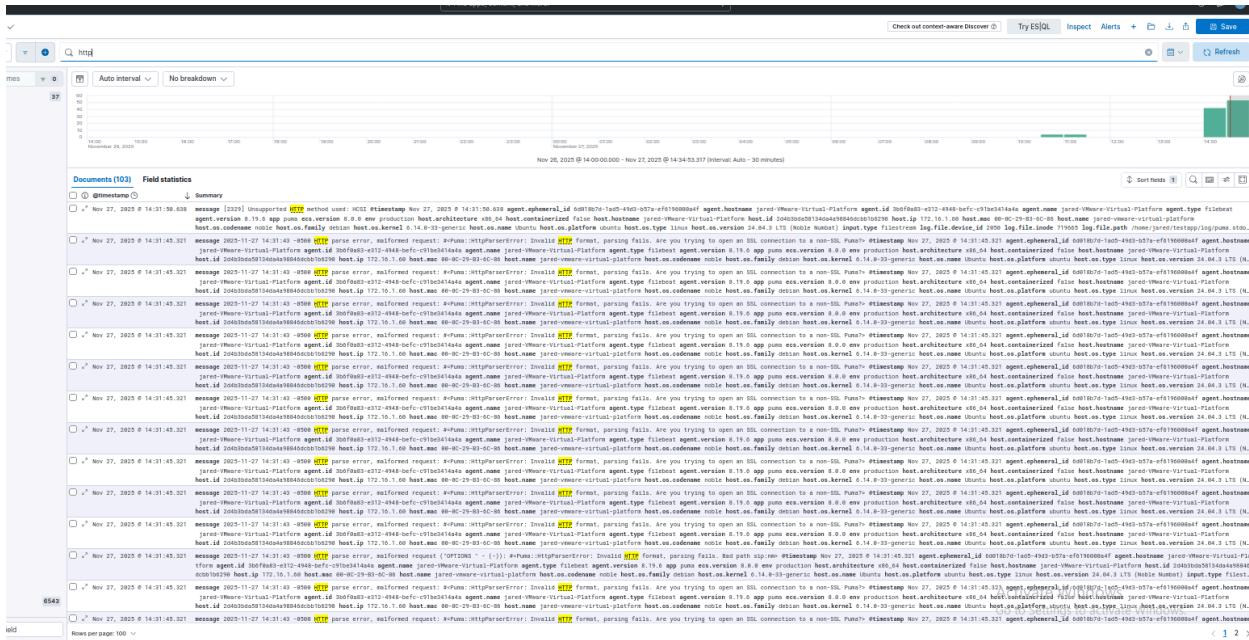


Alot of /get that kali used for this test and we see this on kibana.

```
(kali㉿kali)-[~]
└─$ msfconsole -q -x "use auxiliary/scanner/http/http_version; set RHOSTS 172.16.1.60; set RPORT 3000; run; exit"
RHOSTS ⇒ 172.16.1.60
RPORT ⇒ 3000
[+] 172.16.1.60:3000
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

(kali㉿kali)-[~]
```

Used the metasploit command to attack our ruby rails site that I made.



using HTTP search we saw all the attacks.

27. Postgresql Attacks

```
(kali㉿kali)-[~]
$ sudo nmap -sS -sV -p 5432 172.16.1.70
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-27 14:36 EST
Nmap scan report for 172.16.1.70
Host is up (0.00074s latency).

PORT      STATE SERVICE      VERSION
5432/tcp  open  postgresql PostgreSQL DB 9.6.0 or later
MAC Address: 00:0C:29:99:94:59 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.90 seconds

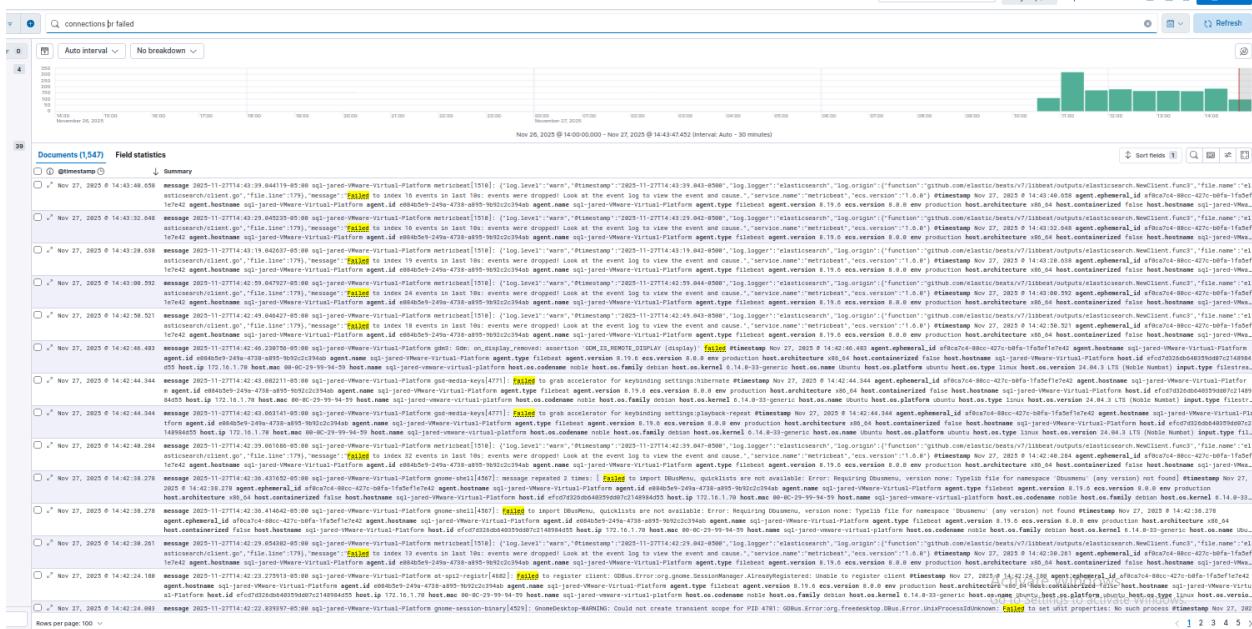
(kali㉿kali)-[~]
```

using nmap to find vulnerabilities.



```
(kali㉿kali)-[~]
$ for i in {1..50}; do PGPASSWORD=wrongpass psql -h 172.16.1.70 -U postgres -c "SELECT 1;" 2>/dev/null & done
[2] 101359
[3] 101360
[4] 101361
[5] 101362
[6] 101363
[7] 101364
[8] 101365
[9] 101366
[10] 101367
[11] 101368
[12] 101369
[13] 101370
[14] 101371
[15] 101372
[16] 101373
[17] 101374
[18] 101375
[19] 101376
[20] 101377
[21] 101378
[22] 101379
[23] 101380
[24] 101381
[25] 101382
[26] 101383
[27] 101384
[28] 101385
[29] 101386
[30] 101387
[31] 101388
```

Using an attack to send bad login attempts into postgres.



We saw those failures here in the logs

28. Win 10 and 11 Attacks

```
(kali㉿kali)-[~]
$ sudo nmap -sS -sv 10.10.1.73 10.10.1.136
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-27 14:47 EST
Nmap scan report for 10.10.1.73
Host is up (0.0018s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds?
5357/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: 00:0C:29:D1:AA:D6 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 10.10.1.136
Host is up (0.0020s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds?
MAC Address: 00:0C:29:BD:43:86 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

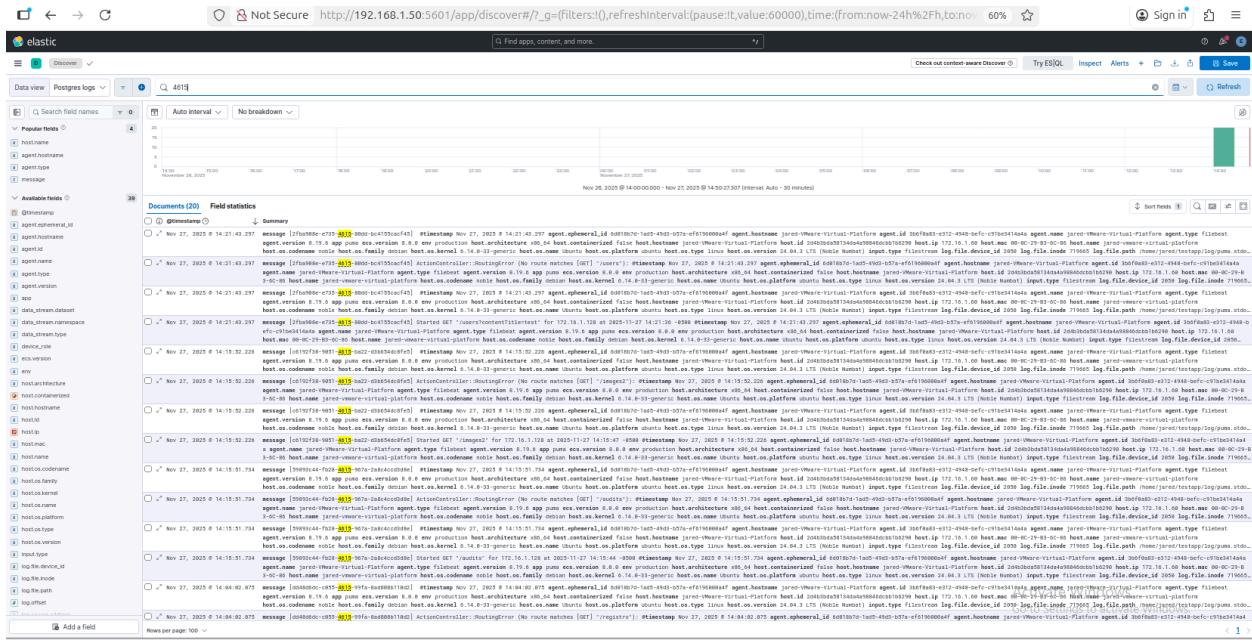
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 2 IP addresses (2 hosts up) scanned in 23.91 seconds
```

Nmap scan to see vulnerabilities in the vmnet 3 machines.

```
(kali㉿kali)-[~]
$ smbclient -L \\10.10.1.73\\ -N
do_connect: Connection to 10.10.1.73 -N failed (Error NT_STATUS_UNSUCCESSFUL)

(kali㉿kali)-[~]
$ smbclient -L \\10.10.1.136\\ -N
do_connect: Connection to 10.10.1.136 -N failed (Error NT_STATUS_UNSUCCESSFUL)
```

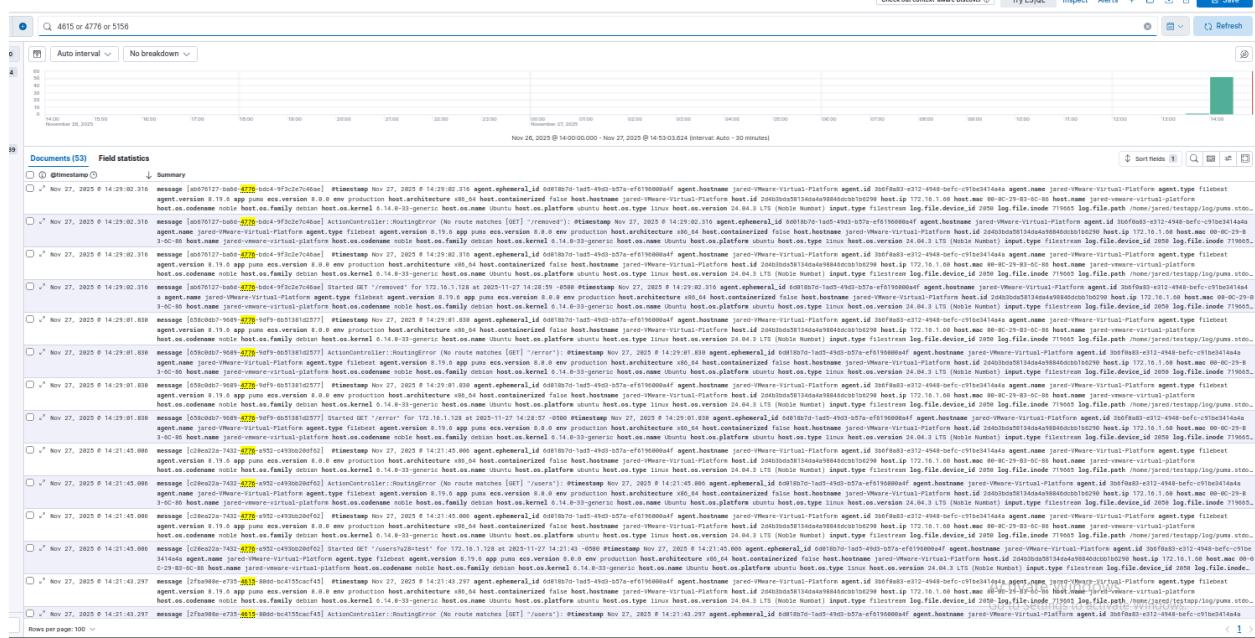
we used a command to try a SMB enumeration attempts that were failed due to windows firewall(good)



Here we see those failed attempts, it uses port 445

We use these attacks on the windows machines that tries to get as much info as it can from the windows firewall and system.

enum4linux -a 10.10.1.73 and enum4linux -a 10.10.1.136



here we saw those in the attempts on ports 445, 4625, 5108, and 139

```
(kali㉿kali)-[~]
$ hydra -l administrator -P /usr/share/wordlists/rockyou.txt rdp://10.10.1.73

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organization

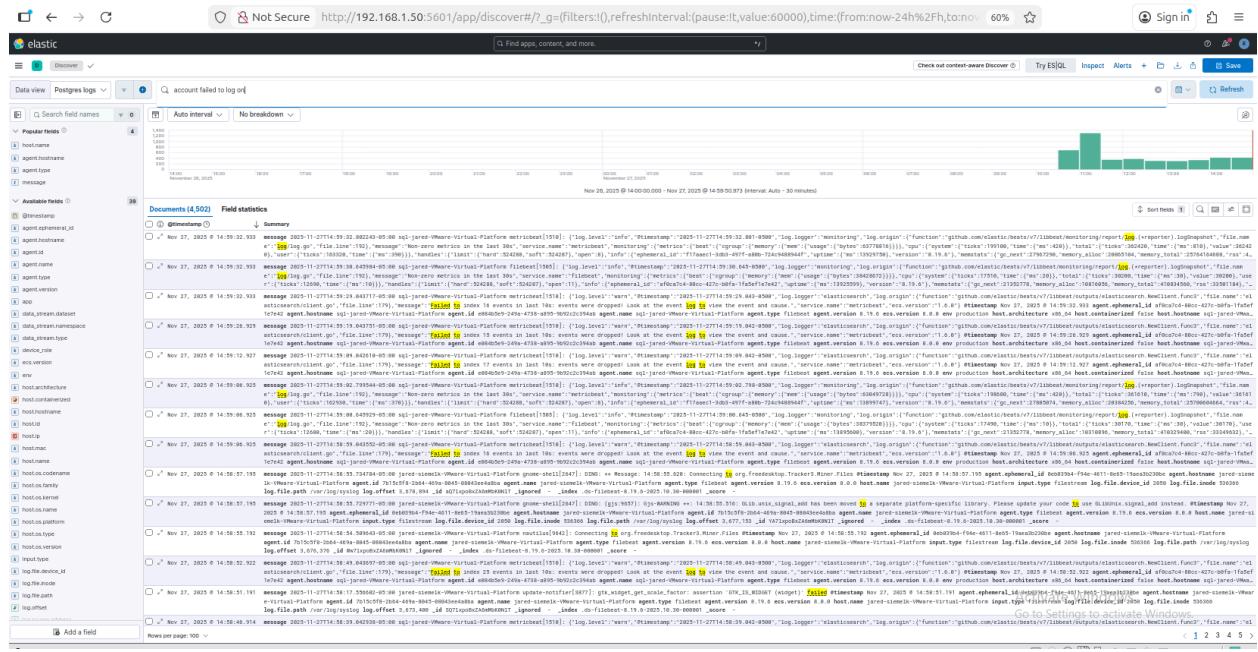
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-27 14:57:40
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parallel connection
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking rdp://10.10.1.73:3389/
[ERROR] freerdp: The connection failed to establish.
[ERROR] all children were disabled due too many connection errors
0 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-27 14:58:14

(kali㉿kali)-[~]
$ hydra -l administrator -P /usr/share/wordlists/rockyou.txt rdp://10.10.1.136
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organization

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-27 14:58:28
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parallel connection
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking rdp://10.10.1.136:3389/
[ERROR] freerdp: The connection failed to establish.
[ERROR] all children were disabled due too many connection errors
0 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-27 14:59:01
```

Hydra attempted to brute-force the Windows RDP login service (port 3389) using:

Username: administrator and Password list: rockyou.txt



Here we saw those attempts on the port search.

```
File System

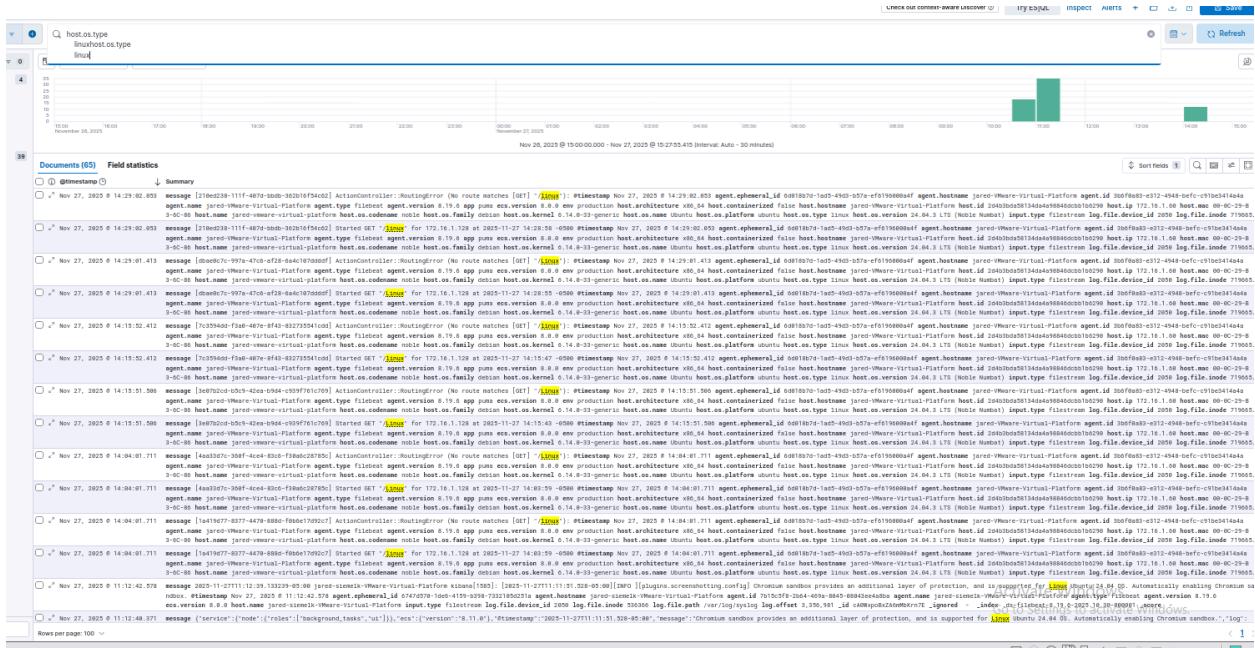
    . . .
    . . .
    . . .
    ( 3 C ) / \ |__| Metasploit!
;@'. __*,..` \|__|_
'....."/

[ metasploit v6.4.99-dev ]
+ -- =[ 2,572 exploits - 1,317 auxiliary - 1,683 payloads ]
+ -- =[ 432 post - 49 encoders - 13 nops - 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(scanner/smb/smb_version) > set RHOSTS 10.10.1.73
RHOSTS => 10.10.1.73
msf auxiliary(scanner/smb/smb_version) > run
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.23/lib/recog/fingerprint/regexp_fac
[*] 10.10.1.73:445 - SMB Detected (versions:2, 3) (preferred dialect:SMB 3.1.1) (compression capabil
b0b7bf204b5b}) (authentication domain:DESKTOP-B83AQEE)
[+] 10.10.1.73:445 - Host is running Version 10.0.19041 (likely Windows 10 version 2004)
[*] 10.10.1.73 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/smb/smb_version) > use auxiliary/scanner/smb/smb_enumshares
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf auxiliary(scanner/smb/smb_enumshares) > set RHOSTS 10.10.1.73
RHOSTS => 10.10.1.73
msf auxiliary(scanner/smb/smb_enumshares) > run
[-] 10.10.1.73:139 - Login Failed: Unable to negotiate SMB1 with the remote host: Not a valid SMB packet
[-] 10.10.1.73:139 - Login Failed: (0xc0000022) STATUS_ACCESS_DENIED: {Access Denied} A process has request
[*] 10.10.1.73: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/smb/smb_enumshares) > set RHOSTS 10.10.1.136
RHOSTS => 10.10.1.136
msf auxiliary(scanner/smb/smb_enumshares) > run
```

Using multiple Metasploit auxiliary modules against the two Windows machines. Attempted enumeration of SMB shares which fails



here i search for any linux asking for information.

29. Pi-hole attacks

```
[*] exec: sudo nmap -sS -sV -p- 192.168.1.50 192.168.1.55

[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-27 15:29 EST
Nmap scan report for 192.168.1.50
Host is up (0.00060s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
5601/tcp  open  http        Elasticsearch Kibana (serverName: jared-siemelk-VMware-Virtual-Platform)
9200/tcp  open  ssl/http    Elasticsearch REST API 7.0 or later (Shield plugin; realm: security)
9300/tcp  open  ssl/vrace?
MAC Address: 00:0C:29:14:B1:A9 (VMware)

Nmap scan report for 192.168.1.55
Host is up (0.00064s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
53/tcp    open  domain      dnsmasq 2.92test21 (pi-hole)
80/tcp    open  webdav
443/tcp   open  ssl/webdav

[+] exec: sudo nmap -sS -sV -p- 192.168.1.50 192.168.1.55
```

nmap on vmnet 1 to see vulnerabilities.

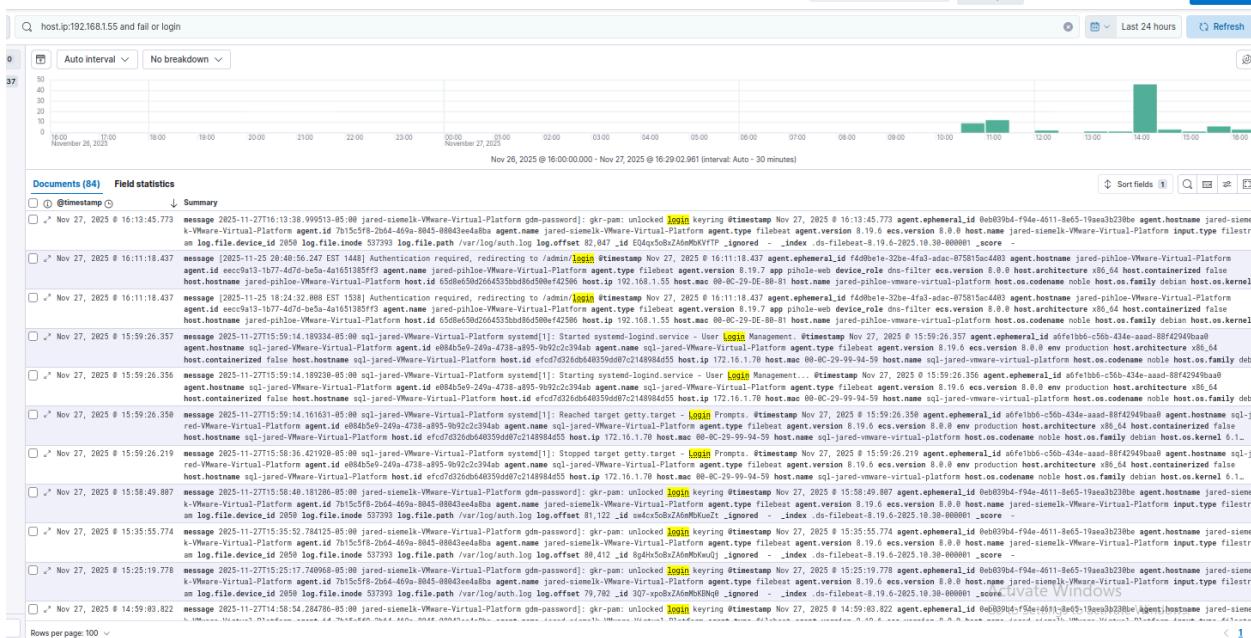
```
(kali㉿kali)-[~]
$ ffuf -u http://192.168.1.55/FUZZ -w /usr/share/wordlists/seclists/Discovery/Web-Content/common.txt
v2.1.0-dev

:: Method : GET
:: URL   : http://192.168.1.55/FUZZ
:: Wordlist : FUZZ: /usr/share/wordlists/seclists/Discovery/Web-Content/common.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout : 10
:: Threads : 40
:: Matcher : Response status: 200-299,301,302,307,401,403,405,500

:: Progress: [4746/4746] :: Job [1/1] :: 566 req/sec :: Duration: [0:00:09] :: Errors: 0 ::

(kali㉿kali)-[~]
$ 
```

using FFUF which is a fast file brute attacker to discover hidden content where I generated content showing it tried.



In kibana we saw all the failed attempts on the machine.

30. ELK attack

```
(kali㉿kali)-[~]
$ ffuf -u https://192.168.1.50:5601/FUZZ -w /usr/share/wordlists/dirb/common.txt -k

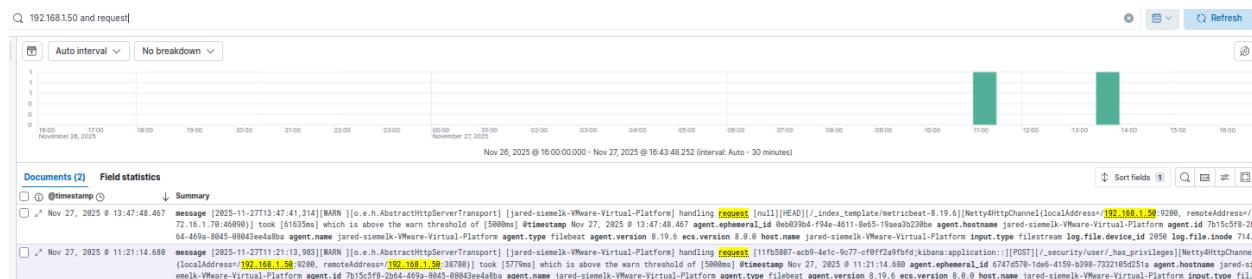
v2.1.0-dev

:: Method : GET
:: URL : https://192.168.1.50:5601/FUZZ
:: Wordlist : FUZZ: /usr/share/wordlists/dirb/common.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout : 10
:: Threads : 40
:: Matcher : Response status: 200-299,301,302,307,401,403,405,500

:: Progress: [4614/4614] :: Job [1/1] :: 497 req/sec :: Duration: [0:00:07] :: Errors: 4614 ::

(kali㉿kali)-[~]
```

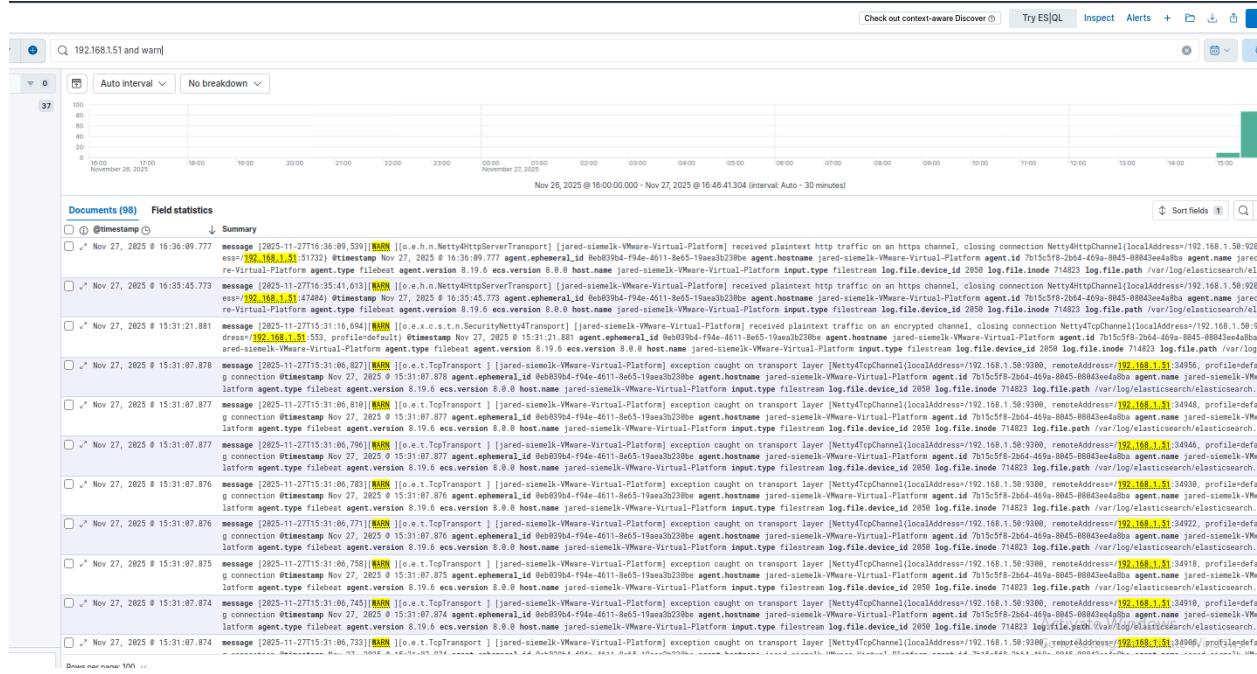
Another FUFF attack on the ELK machine.



Here we saw all the requests on the machine.

```
(kali㉿kali)-[~]
└─$ for i in {1..20}; do curl -k -u bad:bad https://192.168.1.50:9200/ >/dev/null; done
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
100  459  100  459    0     0 13741      0 --::-- --::-- --::-- 13909
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
100  459  100  459    0     0 32239      0 --::-- --::-- --::-- 32785
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
100  459  100  459    0     0 29972      0 --::-- --::-- --::-- 30600
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
100  459  100  459    0     0 30587      0 --::-- --::-- --::-- 32785
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
100  459  100  459    0     0 26421      0 --::-- --::-- --::-- 27000
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
100  459  100  459    0     0 30679      0 --::-- --::-- --::-- 32785
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
100  459  100  459    0     0 30490      0 --::-- --::-- --::-- 30600
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
100  459  100  459    0     0 30988      0 --::-- --::-- --::-- 32785
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
100  459  100  459    0     0 30616      0 --::-- --::-- --::-- 32785
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
100  459  100  459    0     0 30886      0 --::-- --::-- --::-- 32785
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
100  459  100  459    0     0 32990      0 --::-- --::-- --::-- 35307
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
100  459  100  459    0     0 30231      0 --::-- --::-- --::-- 30600
```

I used a repeating credential stuffing(login) attack to the ELK machine

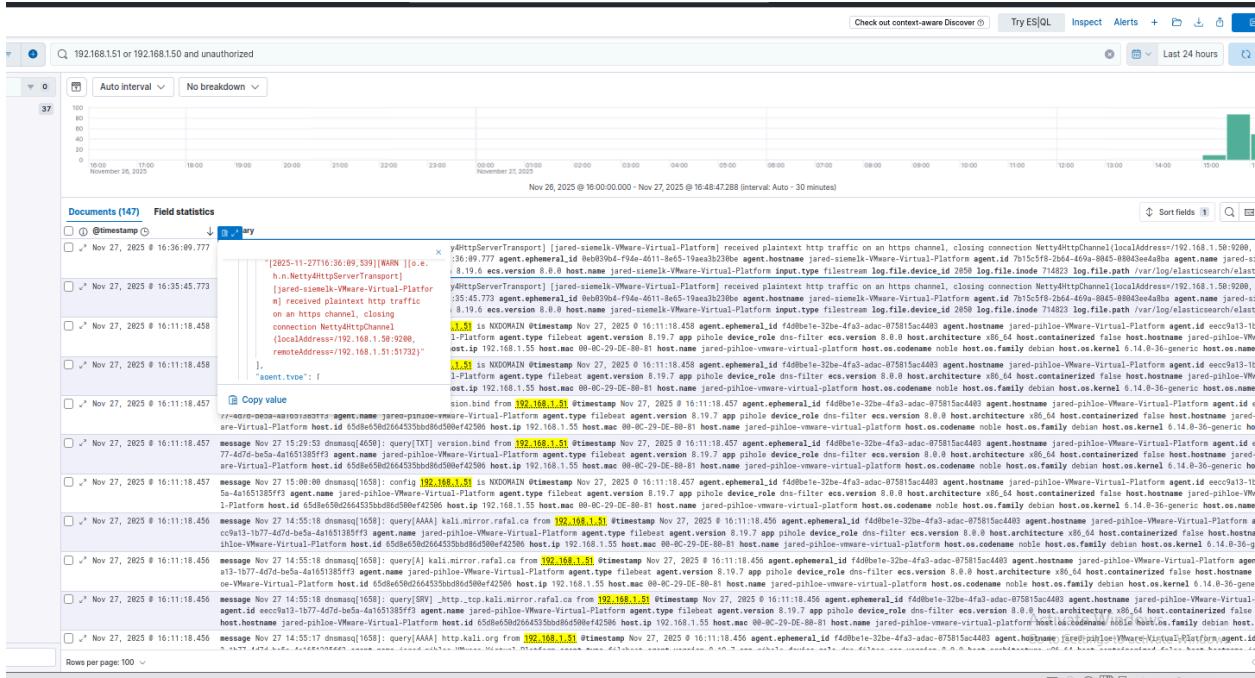


I got to see all these attacks and there were so many logs here.

```
(kali㉿kali)-[~]
$ curl -k https://192.168.1.50:9200/_cat/indices?v
{"error":{"root_cause":[{"type":"security_exception","reason":"missing authentication credentials for REST request [/ _cat/indices?v]","Bearer realm=\\"security\\","ApiKey"]}], "type":"security_exception","reason":"missing authentication credentials for REST request [/ _cat/indices?v]","Bearer realm=\\"security\\","ApiKey"]}, "status":401}

(kali㉿kali)-[~]
$
```

I used an attack to simulate an attacker trying to enumerate the cluster's indices without valid credentials.



We saw those attacks and even went further to see what they stated when it got them.

9. Conclusions and Findings

Working through the attacks and monitoring everything inside the SIEM taught me how predictable most attacker behavior actually is once logs are seen. Tools like Gobuster, FFUF, Nmap, and SMB scanners created obvious patterns in Kibana. Huge spikes of repeated requests that stood out immediately. Even authentication attacks were easy to spot, because every failed login, bad password, or rejected connection sent tons of logs. Seeing this in real time helped me understand why SIEMs are so important. I also realized that once you know what “normal” traffic looks like in your environment, spotting something unusual becomes a lot easier.

On the blue team/defensive side, the project helped me see several improvements that could make the network stronger. pfSense already separated different parts of the network, but enforcing stricter “deny” rules in pfSense and limiting traffic between only the subnets would make it even harder for an attacker to move around and view things. Adding Suricata would give the firewall the ability not only to filter traffic but to recognize things like port scans or brute force attempts as they happen.

Finally, Kibana gave me the ability to visualize attacks, but refining the dashboards and enabling automated alerts would take the setup to the next level. Tracking authentication failures, unusual network activity, DNS filtering events, and spikes in traffic would allow defenders to react quickly instead of relying on manual monitoring. Overall, this project showed me how important visibility, segmentation, and hardening are to any network. The attacks helped me see how an attacker thinks and the SIEM helped me understand how a defender can stay one step ahead.