SECURING THE PERIMETER

7/10/2025

ANSHUL SHUKLA
CBS-0417

Project Scenario

Overview

XYZ is the premier cryptocurrency exchange. They transact over a billion trades everyday and are considered to be one of the most reliable and secure exchanges in the world. Due to their rapid growth, they've faced challenges in scaling their security posture.
The largest challenge they've faced is with their Perimeter Network Security being secure. The networking team was overburdened with the rapid growth and a majority of the network infrastructure was built insecurely.

Due to a lack of visibility and a lack of proper access control setup on the network, it was inevitable that a breach took place! XYZ was hit with a massive attack in which their network was breached and their internal servers were compromised resulting in over 500 Bitcoin being stolen!

Needing to get the bottom of this breach and resolving their current perimeter issues they've contracted you from SecureCorp, a world renowned cybersecurity consulting firm. Your job is to redesign their network architecture securely and set up a SIEM to monitor against future attacks.

Section 1:

Designing a secure Network Architecture

Identify Network Vulnerabilities

1. No Network Segmentation

All the servers (web servers, database servers and the file storage server) are placed together in a single subnet with no separation between public-facing and internal systems. This means there's no isolation between services that should be protected (like databases and storage) and those that are exposed to the internet (like the web servers). If an attacker manages to compromise one of the public-facing web servers, they could move laterally and directly access the databases or the file storage server, potentially stealing or deleting sensitive data.

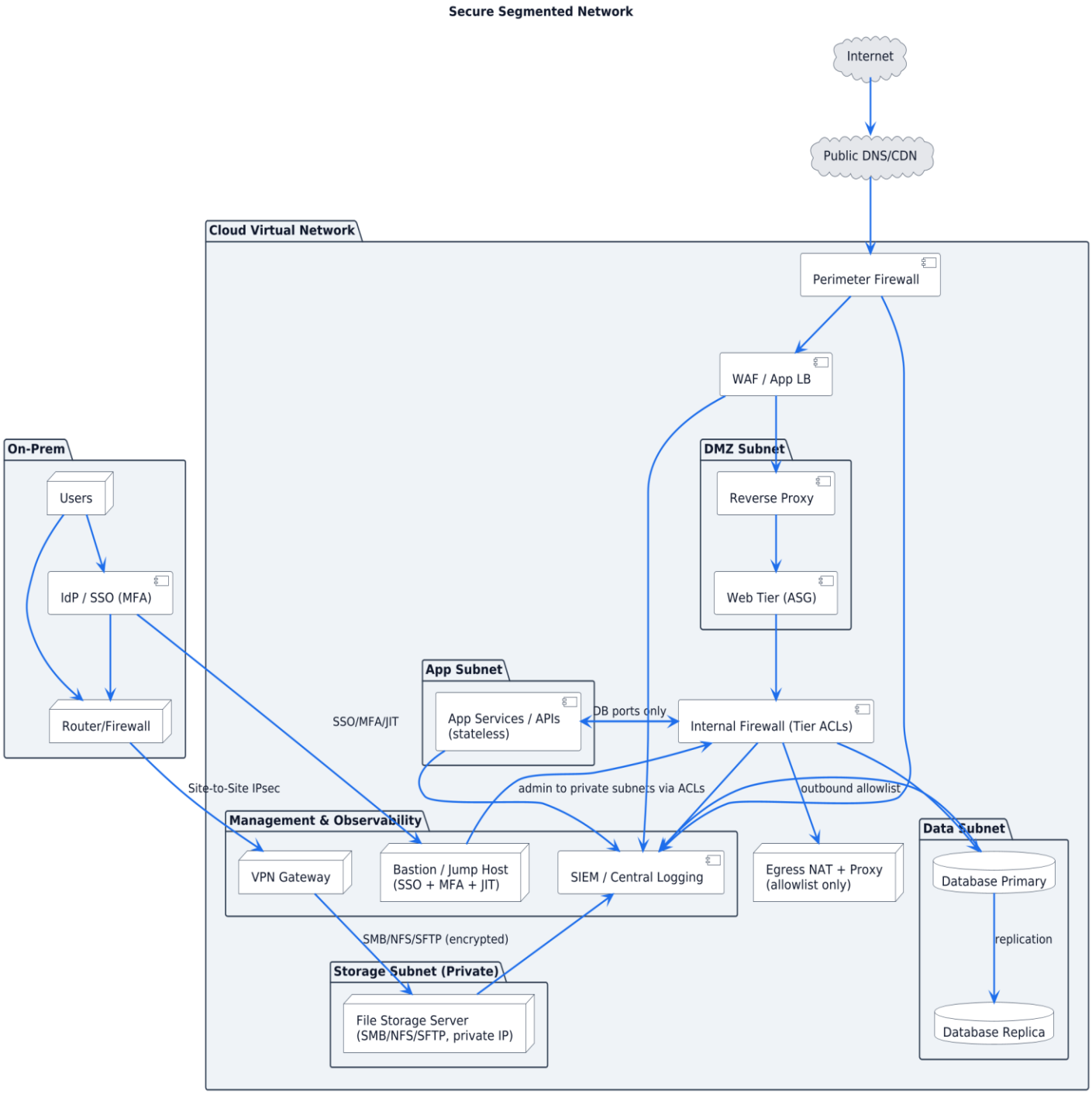
2. Lack of Firewalls and Traffic Filtering

The network design doesn't include any dedicated firewalls between critical components. All the servers are directly connected to the internet and there are no controls in place to filter or block suspicious traffic between the servers and the outside world.

3. No Secure, Encrypted Access to File Storage

The file storage server is just sitting in the same subnet as everything else and appears to be accessible from the internet, with no indication of any encryption or secure tunnel (like a VPN) between the on-premises network and the storage. Sensitive files could be exposed to attackers on the internet or intercepted during transit if encryption isn't enforced.

Network Redesign



Convince the Stakeholders

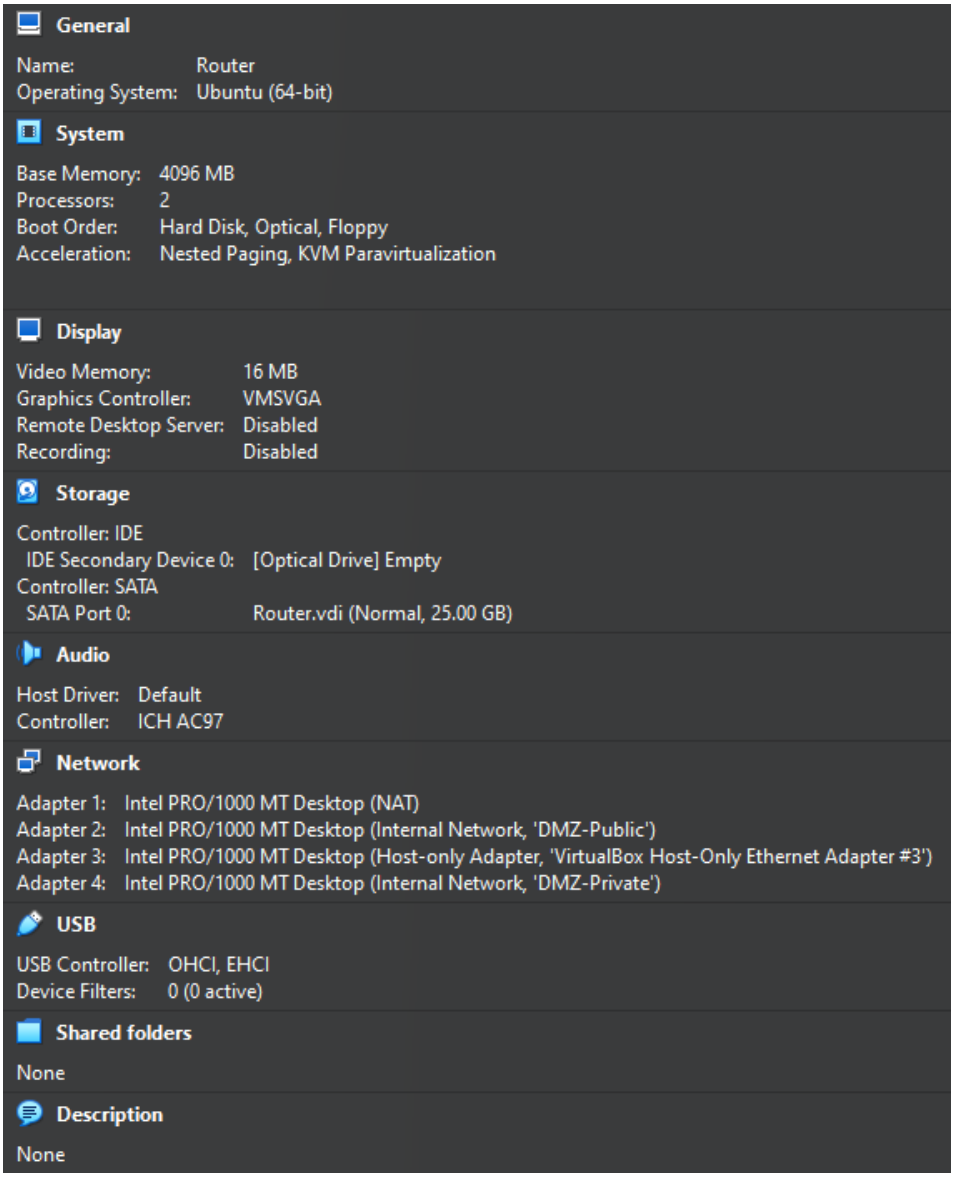
Why do we need to add firewalls to our network?
Firewalls are like security guards for our network they control what comes in and goes out. Without them anyone from the internet could try to access our servers directly, making it much easier for hackers to break in or attack us. By adding firewalls we can block unwanted traffic only allow necessary connections and quickly stop suspicious activity before it reaches sensitive data.
What is the benefit of having different areas in our network for web servers and database servers?
Separating web servers and database servers into different areas (or subnets) adds an extra layer of security. If someone manages to hack a web server, they can't just go straight to our database and steal everything. They'd run into another barrier and more security checks. This setup also makes it easier to control and monitor traffic so we can spot unusual behavior faster and protect our most valuable information.
What does a VPN do for our connection to the file storage server?
A VPN creates a secure encrypted tunnel between our on-premises network and the file storage server in the cloud. That means even if someone is watching the internet traffic they can't see or steal our files. Only trusted users with the right credentials can access the storage and everything sent over the VPN is protected from eavesdropping and tampering.

Section 2:

Building a secure Network Architecture in VirtualBox

Network Setup

1. Public DMZ: 192.168.10.0/24 – Web-Server
2. Private DMZ: 192.168.20.0/24 – ELK-Server



Router VM connected to both DMZ-Public and DMZ-Private adapters

Network Setup

```
routeradmin@Router:~$ hostname
ip -br a
ip r
sysctl net.ipv4.ip_forward
sudo iptables -L FORWARD -v -n     # or nft equivalent if using nftables
Router
lo               UNKNOWN        127.0.0.1/8 ::1/128
enp0s3           UP             10.0.2.15/24 metric 100 fd17:625c:f037:2:a00:27ff:fe2b:d419/64 fe80::a00:27ff:fe2
b:d419/64
enp0s8           UP             192.168.10.1/24 fe80::a00:27ff:fea9:d5df/64
enp0s9           UP             192.168.56.10/24 fe80::a00:27ff:fe03:e440/64
enp0s10          UP             192.168.20.1/24
default via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
10.0.2.2 dev enp0s3 proto dhcp scope link src 10.0.2.15 metric 100
10.0.2.3 dev enp0s3 proto dhcp scope link src 10.0.2.15 metric 100
192.168.10.0/24 dev enp0s8 proto kernel scope link src 192.168.10.1
192.168.20.0/24 dev enp0s10 proto kernel scope link src 192.168.20.1
192.168.56.0/24 dev enp0s9 proto kernel scope link src 192.168.56.10
net.ipv4.ip_forward = 1
[sudo] password for routeradmin:
Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
 1637 5817K ACCEPT     0    --  *      *       0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLIS
HED
   81  4884 ACCEPT     0    --  *      *       192.168.10.0/24      192.168.20.0/24
    0     0 ACCEPT     0    --  *      *       192.168.20.0/24      192.168.10.0/24
routeradmin@Router:~$
```

Router VM showing the connected IPs and Packet Forwarding

Section 3:
Continuous Monitoring with a SIEM

Understanding SIEM Benefits

1. Real Time Threat Detection
A SIEM system helps us spot security threats as they happen, instead of finding out after the damage is done. It collects and analyzes data from all parts of our network so we can quickly see if something suspicious is going on and respond right away.
2. Centralized Log Management
With a SIEM all our security logs and alerts are collected in one place. This makes it much easier to investigate incidents, find patterns and see exactly what happened during an attack without having to dig through different systems.
3. Helps Meet Compliance Requirements
Many regulations like GDPR or HIPAA require us to monitor our systems and keep detailed records of security events. A SIEM helps us do this automatically making it easier to prove that we're following the rules and protecting sensitive data.

Deploy SIEM Components in VirtualBox


```
elkadmin@ELK-Server:~$ systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/usr/lib/systemd/system/kibana.servi     Active: active (running) since Tue 2025-10-07 03:13:
       Docs: https://www.elastic.co
   Main PID: 11755 (node)
      Tasks: 11 (limit: 9436)
     Memory: 759.6M (peak: 1.0G)
        CPU: 2min 18.207s
     CGroup: /system.slice/kibana.service
             └─11755 /usr/share/kibana/bin/../node/glibc-
lines 1-10/10 (END)
^C
elkadmin@ELK-Server:~$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/usr/lib/systemd/system/elasticsearc
    Drop-In: /etc/systemd/system/elasticsearch.service.d
             └─limits.conf, override.conf
     Active: active (running) since Tue 2025-10-07 03:29:
       Docs: https://www.elastic.co
   Main PID: 14817 (java)
      Tasks: 102 (limit: 9436)
     Memory: 4.5G (peak: 4.6G)
        CPU: 3min 5.502s
     CGroup: /system.slice/elasticsearch.service
             ├─14817 /usr/share/elasticsearch/jdk/bin/jav
             ├─14876 /usr/share/elasticsearch/jdk/bin/jav
             └─14896 /usr/share/elasticsearch/modules/x-p

Oct 07 03:28:48 ELK-Server systemd[1]: Starting elasticse
Oct 07 03:29:14 ELK-Server systemd[1]: Started elasticsea
lines 1-17/17 (END)
```

Setup Monitoring

```
webadmin@Web-Server:/etc/filebeat/modules.d$ sudo systemctl status --no-pager filebeat
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
     Loaded: loaded (/usr/lib/systemd/system/filebeat.service; enabled; preset: enabled
)
     Active: active (running) since Mon 2025-10-06 23:19:31 UTC; 29s ago
       Docs: https://www.elastic.co/beats/filebeat
   Main PID: 4354 (filebeat)
      Tasks: 10 (limit: 2268)
     Memory: 107.1M (peak: 107.3M)
        CPU: 2.552s
     CGroup: /system.slice/filebeat.service
             └─4354 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/fil…

Oct 06 23:19:33 Web-Server filebeat[4354]: {"log.level":"info","@timestamp":"2025-10-0…
Oct 06 23:19:33 Web-Server filebeat[4354]: {"log.level":"info","@timestamp":"2025-10-0…
Oct 06 23:19:33 Web-Server filebeat[4354]: {"log.level":"info","@timestamp":"2025-10-0…
Oct 06 23:19:33 Web-Server filebeat[4354]: {"log.level":"info","@timestamp":"2025-10-0…
Oct 06 23:19:33 Web-Server filebeat[4354]: {"log.level":"info","@timestamp":"2025-10-0…
Oct 06 23:19:33 Web-Server filebeat[4354]: {"log.level":"info","@timestamp":"2025-10-0…
Oct 06 23:19:33 Web-Server filebeat[4354]: {"log.level":"info","@timestamp":"2025-10-0…
Oct 06 23:19:33 Web-Server filebeat[4354]: {"log.level":"info","@timestamp":"2025-10-0…
Oct 06 23:19:33 Web-Server filebeat[4354]: {"log.level":"info","@timestamp":"2025-10-0…
Oct 06 23:19:33 Web-Server filebeat[4354]: {"log.level":"info","@timestamp":"2025-10-0…
Hint: Some lines were ellipsized, use -l to show in full.
webadmin@Web-Server:/etc/filebeat/modules.d$
```

Filebeat service on the web server
[command: ‘systemctl status filebeat’]

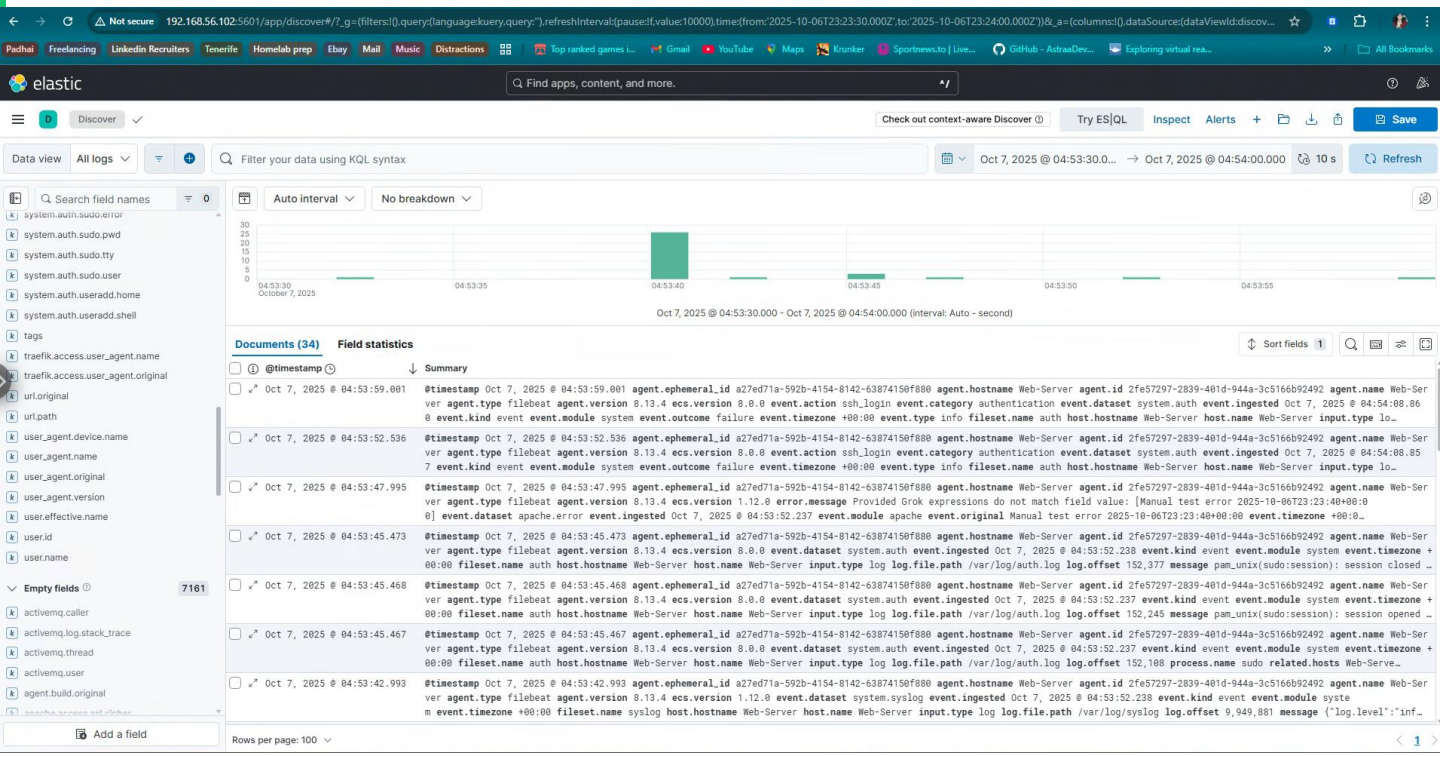
Setup Monitoring

```
elkadmin@ELK-Server:~$ systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/usr/lib/systemd/system/kibana.servi>     Active: active (running) since Tue 2025-10-07 03:13:>
       Docs: https://www.elastic.co
   Main PID: 11755 (node)
      Tasks: 11 (limit: 9436)
     Memory: 759.6M (peak: 1.0G)
        CPU: 2min 18.207s
     CGroup: /system.slice/kibana.service
             └─11755 /usr/share/kibana/bin/../node/glibc->lines 1-10/10 (END)
^C
elkadmin@ELK-Server:~$ system ctl status elasticsearch
Command 'system' not found, did you mean:
  command 'systemd' from deb systemd (255.4-1ubuntu8.10)
  command 'system3' from deb simh (3.8.1-6.1)
Try: sudo apt install <deb name>
elkadmin@ELK-Server:~$
```

Elastic search and Kibana running on ELK-Server

Project Information Slide

Setup Monitoring



Kibana receives logs from the Filebeat host

Section 4:
Zero Trust

Zero Trust Comparison

1. Consideration of All Resources

Zero Trust Approach: Treats every device, software, and system as a potential security risk, even if it's inside the network. It assumes nothing should be trusted by default every connection and device has to prove it's safe.	**Traditional Approach:** Usually trust everything that's already inside the network perimeter. Once you're "in," you're trusted, and there's less focus on what devices are actually doing.

Benefits of Zero Trust: By checking every device and system all the time Zero Trust reduces the chances that a hacker can move around undetected if they break in. It's much harder for threats to spread.

2. Secured Communication

Zero Trust Approach: All communication, no matter if it's within the company network or going outside is encrypted and secured. Data is always protected in transit.	**Traditional Approach:** Only connections that go out to the internet are encrypted, while internal network traffic might not be secured.

Benefits of Zero Trust: Even if someone taps into the internal network, they can't read or steal sensitive data because everything is encrypted. This stops a lot of common attacks dead in their tracks.

Zero Trust Comparison

3. Continuous Monitoring	
Zero Trust Approach: The system always monitors and checks the security status of users, devices, and software.	**Traditional Approach:** Usually checks users and devices just when they first connect. After that, it mostly assumes things are safe.
Benefits of Zero Trust: Zero Trust can spot suspicious activity right away not just at the beginning. This means we can catch threats faster and stop attacks before they do damage.	

Zero Trust Model

Device Agent & Gateway Model

I chose the Device Agent and Gateway model for XYZ because it gives us a stronger way to protect the network, especially with how people are working these days. This model works by having a security agent installed on every device, like laptops or phones, that wants to connect to our network. Every connection also has to pass through a secure gateway. The gateway checks that the device is safe and follows our security rules before letting it access anything sensitive.

What I like about this model is that even if someone accidentally brings a risky device or if a device gets infected, it can’t get into the network unless it passes all the checks at the gateway. This stops a lot of attacks before they start. It also means we can see what devices are trying to connect at all times, so we can spot weird or suspicious behavior and react quickly.