Implementation and Evaluation of Zero Trust Architecture (ZTA) for Network Security

Student - Vinay Reddy Bulagakula
Project Supervisor - Kyle Martin

Aims & Objectives

Aim:
To design, implement, and evaluate a Zero Trust Architecture (ZTA) framework to boost the security of a small business network by employing contemporary security methods and tools.

Objectives:
Analyse the existing models of ZTA, such as NIST 800-207 with the help of literature review.
come up with a ZTA based framework with Multi-Factor Authentication (MFA) and Role-Based Access Control (RBAC).
Apply micro-segmentation and network monitoring by Wireshark and SIEM systems.
Make a policy management application in C#, HTML, CSS, and MySQL on the web interface.
Automate alerting and the enforcement of security enforcement policy with Python scripts.
Test the capabilities of the ZTA system in comparison with the conventional security models using simulated test situations.

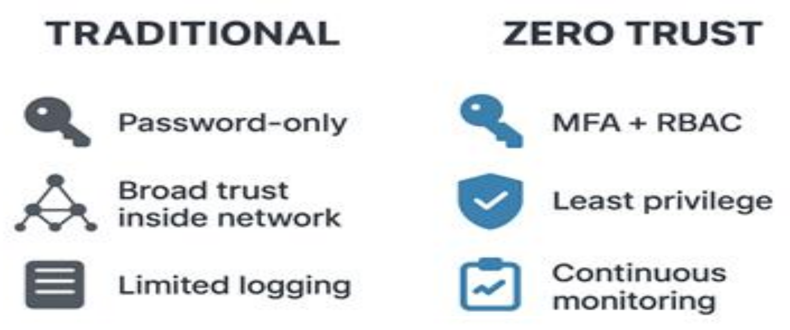


Fig.1 Security Type in Networks

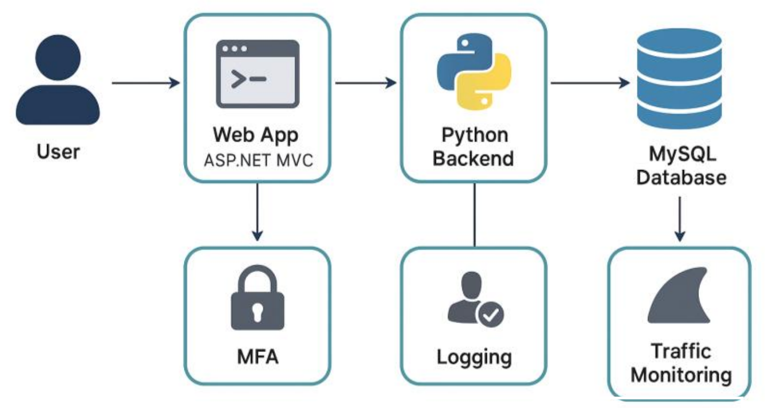


Fig.2 Logging and Monitoring

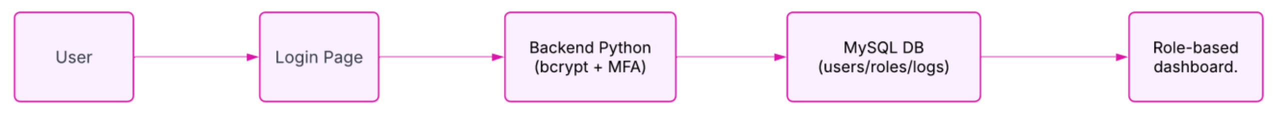


Fig.3 Login Process

Methods

To illustrate **Zero Trust Architecture (ZTA)** practically, I have made a proof-of-concept system based on open-source and a hybrid backend architecture.

System Setup :

- I tested the system on Windows 10 operating system using Python (in the backend), ASP.NET Core MVC (user interface), and MySQL (database).
- Authentication, access control and logging were done by Python scripts, and the web application offered an easy one to use interface.

Authentication & MFA :

- Password hash was done with crypt hash
- I enabled Time-based One-Time Passwords (TOTP) Google Authenticator, where to log in the password was needed, and an OTP.
- A QR code was created at sign-up to facilitate the set up of an MFA.

Role Based Access Control (RBAC)

- I developed three roles Admin, Analyst, Viewer.
- Admins could add their own policies, but Analysts viewed logs and view only users were permitted to see only what they were allowed to see.

Logging & Monitoring

- Any attempted log-ins, OTP results or access requests were stored in MySQL with IP and timestamp.
- To perform the validation, I deployed Wireshark to monitor and analyse the traffic on the network to ensure that the expected Zero Trust patterns were observed upon requests and responses.

Testing & Evaluation :

- I tested the simulated logins (proper, incorrect, expired OTPs, unauthorized access).
- The system was able to block unauthorized access and capture all events.
- As compared to the traditional models, it was identified that my prototype was superior in terms of credential theft and lateral movement.

Ongoing Work

Enhancing MFA integration:
Better QR-code on boarding, testing times on OTP expiry using multiple devices.

Increase RBAC enforcements:
Role enhancements and restrictions validation with simulated insiders.

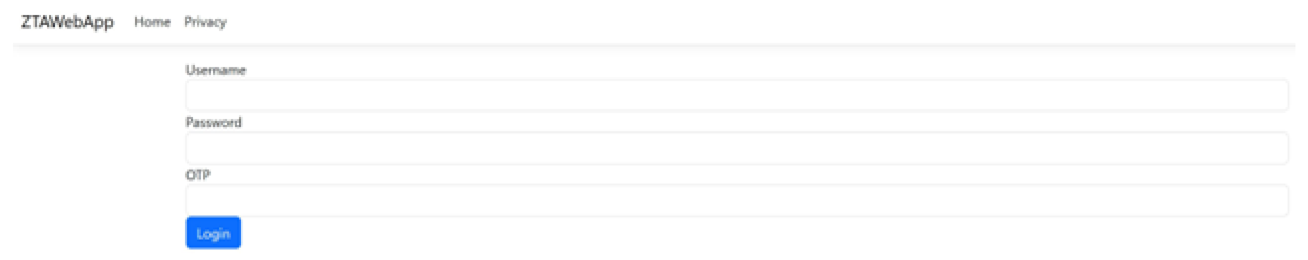
Log analysis:
Writing requests to the MySQL log to detect the failures, malicious IPs and anomalies in access.

Traffic surveillance:
Leveraging Wireshark to detect anomalous patterns (e.g., repetitive Failed logins, occurrences of bursts).

Automation scripts:
Expanding Python scripts to generate texts of alerts based on log events, and connecting with lightweight SIEM products

Discussions

The implementation shows that ZTA principles can be applied effectively at a prototype level using open-source tools. MFA reduced the risk of password-only attacks, while RBAC restricted lateral movement within the system. Logs enhanced traceability, and traffic analysis added another defensive layer. However, integration between Python and ASP.NET introduced complexity, and scaling to enterprise-level would require containerisation or cloud-native services.



Conclusion and Future Work

Conclusions: The system demonstrated practical enforcement of ZTA principles, combining authentication, authorization, and monitoring. The layered defense approach increased system security. Prototype confirms improved resilience against credential Theft.
Future Work: Integration with SIEM platforms, real-time anomaly detection using ML, and deployment in cloud environments.

Acknowledgements

I would like to thank my Supervisor, **Kyle Martin** for his support.
This work was inspired and guided by the **Zero Trust Architecture Framework** as defined in the
NIST Special Publication 800-207 (Rose et al., 2020
https://doi.org/10.6028/NIST.SP.800-207

SCHOOL OF COMPUTING, ENGINEERING & TECHNOLOGY

MSc IT with Cyber Security