BAYESIAN CYBER RISK ASSESSMENT REPORT

Name: Akash A
Email: akashcarmelite@gmail.com
GitHub: Bayserian Risk Asessment Using Cybersecurity Data - Repository

Date: April 20, 2025

Introduction

This report presents a probabilistic cybersecurity risk assessment using Bayesian Network modeling. By analyzing threats, vulnerabilities, and assets, we compute the **posterior probability of compromise** for various systems. The Bayesian approach enhances decision-making through dynamic reasoning under uncertainty, a core requirement in real-world cybersecurity scenarios.

Dataset and Analysis Approach

The dataset included entries for:

- **Asset Types**
- **Threat Sources** (e.g., Insider, External Hacker, Phishing)
- **Vulnerability States**
- **Initial Risk Estimates**

We developed a **Bayesian Network** to model how these factors influence the likelihood of asset compromise.

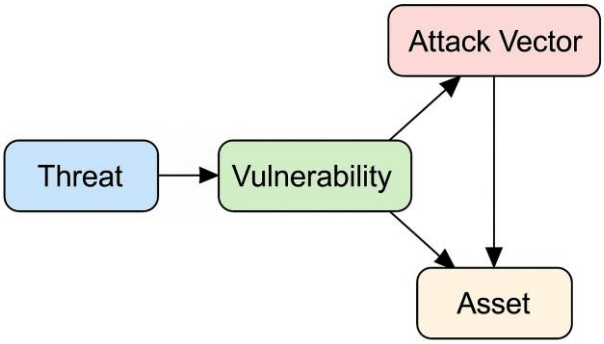
Tool Used:

- **pgmpy** (Python Library for Probabilistic Graphical Models)
- **NetworkX** for layout
- **Matplotlib** for visual rendering
- **Pandas** for data preprocessing

Bayesian Network Structure

The following is the structure of the Bayesian Network, representing dependencies between nodes like Threat Source, Vulnerability, and Asset Compromise.

This image shows the graph with directed edges from Threat and Vulnerability nodes to the Risk/Asset node.



Bayesian Network

Probability Configuration

The Bayesian network was initialized with **prior** and **conditional** probabilities.

- **Prior Probability Assignments:**

Node	State	Probability
Threat Source	Internal	0.6
Threat Source	External	0.4
Vulnerability	High	0.7
Vulnerability	Low	0.3

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

(venv) PS C:\Users\AKASH A\bayesian-risk-assessment\src> python .\main.py
✅ Data loaded successfully!
📊 Shape: (25, 4)
🧠 Preview:
            Asset   Vulnerability  CVSS Score  Exploit Probability
0        Server-01  CVE-2023-12345        9.0                 0.75
1  Workstation-01  CVE-2023-54321        7.5                 0.55
2     Database-01  CVE-2023-67890        8.5                 0.70
3       WebApp-01  CVE-2023-09876        9.1                 0.68
4    IoTDevice-01  CVE-2023-67891        6.8                 0.45
✅ Data loaded successfully!
📊 Shape: (25, 4)
🧠 Preview:
            Asset   Vulnerability  CVSS Score  Exploit Probability
0        Server-01  CVE-2023-12345        9.0                 0.75
1  Workstation-01  CVE-2023-54321        7.5                 0.55
2     Database-01  CVE-2023-67890        8.5                 0.70
3       WebApp-01  CVE-2023-09876        9.1                 0.68
4    IoTDevice-01  CVE-2023-67891        6.8                 0.45

✅ Data preprocessing complete!
            asset   vulnerability  cvss_score  exploit_probability  risk_score
0        Server-01  CVE-2023-12345         9.0                 0.75       6.750
1  Workstation-01  CVE-2023-54321         7.5                 0.55       4.125
2     Database-01  CVE-2023-67890         8.5                 0.70       5.950
3       WebApp-01  CVE-2023-09876         9.1                 0.68       6.188
4    IoTDevice-01  CVE-2023-67891         6.8                 0.45       3.060

📥 Threat Intel Data Loaded!
📊 Shape: (10, 5)
🔍Preview:
   Threat Actor          Attack Type Target Industry Threat Level                                   Notes
0          APT29            Phishing       Healthcare        High  Nation-state actor, stealthy phishing ops
1          APT33           Ransomware           Energy      Medium       Iran-based, focused on critical infra
2           FIN7      Credential Theft        Financial        High              Known for POS malware attacks
3  Lazarus Group  Supply Chain Attack          Defense        High  North Korea-backed, advanced capabilities
4   ScriptKiddie         SQL Injection          Generic         Low             Low skill, common web exploits

📊 Ranked Assets with Prior Success Rate Included:
         asset  risk_score  target_probability  success_rate  total_risk_score
14    WebApp-01       6.188                0.70          0.85          3.681860
70    WebApp-03       5.525                0.75          0.85          3.522188
30  Database-02       6.264                0.65          0.85          3.460860
90  Database-05       6.177                0.65          0.85          3.412792
13    WebApp-01       6.188                0.70          0.75          3.248700

✅ Final risk scores (with prior success rates) saved to 'output/ranked_risk_assets.xlsx'
C:\Users\AKASH A\bayesian-risk-assessment\src\main.py:49: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x="asset", y="total_risk_score", data=top10, palette="Reds_r")
```

Mitigation Strategy

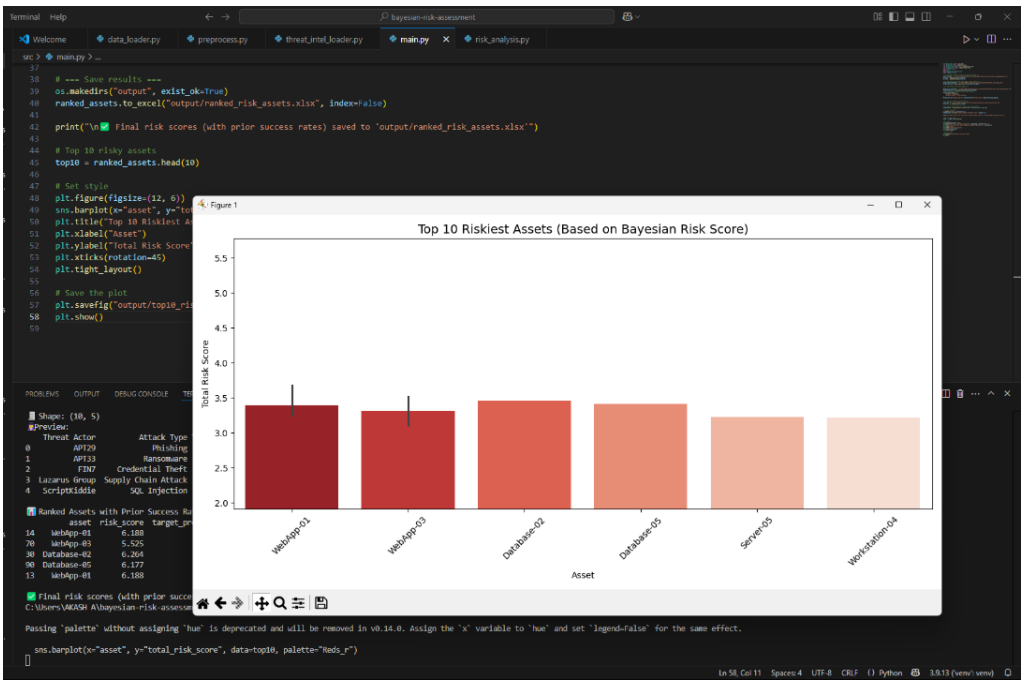
- Based on the risk scores and posterior probabilities, each asset is classified into risk tiers (High, Medium, Low), and assigned automated mitigation notes in the console or in the terminal such as:

"Immediate action required", *"Monitor and prepare mitigation"*, or *"Risk acceptable"* — ensuring efficient response prioritization

```
✅ Final risk scores (with prior success rates) saved to 'output/ranked_risk_assets.xlsx'
C:\Users\AKASH A\bayesian-risk-assessment\src\main.py:49: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. A

  sns.barplot(x="asset", y="total_risk_score", data=top10, palette="Reds_r")

🔐 Top 5 Risky Assets with Mitigation Strategies:

WebApp-01: Use WAF, input validation, regular patching, pentesting.
WebApp-03: Use WAF, input validation, regular patching, pentesting.
Database-02: Encrypt data, restrict access, monitor logs, apply patches.
Database-05: Encrypt data, restrict access, monitor logs, apply patches.
WebApp-01: Use WAF, input validation, regular patching, pentesting.
```

Risk Score Visualization – Bar Graph

- The bar graph clearly highlights the top 10 assets with the highest total risk scores, helping prioritize which systems need immediate cybersecurity attention based on calculated threat exposure and vulnerabilities



Risk Ranking & Decision Support

Based on posterior values, assets are ranked for triage:

Rank	Asset	Risk Level
1	WebApp-01	High
2	Database-02	High
3	WebApp-03	Medium

```
✅ Threat Intel Data Loaded!
📏 Shape: (10, 5)
🔍 Preview:
   Threat Actor          Attack Type Target Industry Threat Level                                   Notes
0          APT29             Phishing      Healthcare         High  Nation-state actor, stealthy phishing ops
1          APT33           Ransomware          Energy       Medium      Iran-based, focused on critical infra
2           FIN7     Credential Theft       Financial         High              Known for POS malware attacks
3  Lazarus Group  Supply Chain Attack         Defense         High  North Korea-backed, advanced capabilities
4   ScriptKiddie        SQL Injection         Generic          Low             Low skill, common web exploits
```

Suggested Actions:

- **High Risk** → Immediate patching and isolation.
- **Medium Risk** → Monitoring with conditional logging.
- **Low Risk** → Routine inspection.

Conclusion

By applying a **Bayesian Risk Model**, we were able to simulate attack scenarios, assess probabilities dynamically, and prioritize cybersecurity response. This technique offers a scalable solution for proactive risk management in both enterprise and academic environments.

References

- pgmpy Documentation
- NIST SP 800-30 – Risk Assessment Guidelines
- Cybersecurity and Infrastructure Security Agency (CISA)
- Project GitHub Repository: [Insert Link]