Abstract

This report examines the firm's exposure to cyber intrusions by categorizing attack vectors, comparing internal and external breaches, and assessing the limitations of traditional defenses. Modern threats such as zero-day exploits, fileless malware, and insider misuse often evade conventional tools, making proactive strategies essential. To address this, a predictive machine learning model is introduced to anticipate intrusion types, supported by performance evaluation and visual analytics. The approach provides actionable insights for prioritizing risks and improving overall cybersecurity resilience.

Introduction

The growing reliance on digital infrastructure has significantly increased organizational vulnerability to cyberattacks. Intrusions today stem from diverse sources including network exploitation, application flaws, social engineering, and insider threats. While external breaches often exploit perimeter weaknesses, internal threats are harder to detect due to trusted access. Traditional defenses like firewalls and antivirus software are no longer sufficient against evolving tactics such as advanced persistent threats and encrypted attacks. To strengthen defenses, organizations must adopt predictive and intelligence-driven approaches. This study combines intrusion mapping, breach analysis, and a machine learning–based predictive model to identify likely threats, highlight key risk indicators, and guide proactive cybersecurity strategies.

QUESTION 9

You are part of a cybersecurity consulting group, asked to help a firm understand its exposure to different kinds of digital intrusions.

a) Begin by mapping out and categorizing all possible types of intrusions based on their attack vectors. (**Remember, Understand**)

b) Present a comparison between internal and external breaches using real-world or theoretical cases. Given the variety of sophisticated attacks today, you're also asked to explain why traditional defenses often fail to detect certain modern intrusions.

(Analyze, Evaluate)

c) Design—and ideally implement—a predictive model for the firm, helping it to anticipate which types of intrusions are most likely to affect its operation. (**Create)**

Cybersecurity Intrusions: Mapping, Categorization, and Comparative Analysis

(a) Mapping and Categorizing Intrusions Based on Attack Vectors

Digital intrusions can be classified according to the **attack vectors**—the paths or means by which a threat actor gains unauthorized access to systems, networks, or data. Below is a structured mapping of intrusion categories:

1. Network-Based Intrusions

- **Attack Vectors**: Exploiting vulnerabilities in communication protocols, unsecured ports, or misconfigured firewalls.
- **Examples**:
    - Distributed Denial of Service (DDoS) attacks
    - Man-in-the-Middle (MitM) interception
    - DNS spoofing/poisoning

2. Application-Level Intrusions

- **Attack Vectors**: Targeting software flaws, insecure coding, or weak application security.
- **Examples**:

- SQL Injection
- Cross-Site Scripting (XSS)
- Remote Code Execution (RCE)

3. Endpoint-Based Intrusions

- **Attack Vectors**: Compromising individual devices such as laptops, desktops, or IoT devices.
- **Examples**:
  - Malware infections (Trojan, Worms, Ransomware)
  - Keyloggers
  - Exploiting outdated or unpatched operating systems

4. Social Engineering Intrusions

- **Attack Vectors**: Manipulating human behavior to bypass technical defenses.
- **Examples**:
  - Phishing and Spear-Phishing
  - Pretexting or impersonation
  - Baiting (USB drops, malicious downloads)

5. Insider Threats

- **Attack Vectors**: Malicious or negligent employees, contractors, or partners.
- **Examples**:
  - Data theft by disgruntled employees
  - Accidental leaks via weak password practices
  - Misuse of privileged access

6. Supply Chain and Third-Party Intrusions

- **Attack Vectors**: Exploiting vulnerabilities in external vendors, partners, or software supply chains.
- **Examples**:
  - SolarWinds supply chain attack
  - Compromised SaaS providers
  - Tampered hardware or firmware

7. Physical Intrusions

- **Attack Vectors**: Direct access to hardware and storage devices.
- **Examples**:
  - Theft of servers or laptops
  - Installation of rogue USB devices
  - Unauthorized physical entry into data centers

(b) Comparison of Internal vs. External Breaches

1. External Breaches

- **Definition**: Unauthorized access originating from outside the organization's perimeter.
- **Characteristics**:
    - Usually opportunistic, financially motivated, or state-sponsored.
    - Target firewalls, web applications, exposed APIs, or phishing users.
- **Real-World Example**:
    - **Equifax Breach (2017)** – Attackers exploited a web application vulnerability in Apache Struts, exposing sensitive financial data of 147 million individuals.

2. Internal Breaches

- **Definition**: Breaches initiated from within the organization by employees, contractors, or insiders with access.
- **Characteristics**:
    - Can be intentional (malicious) or accidental (negligent).
    - Often harder to detect because insiders already have some level of trust and access.
- **Real-World Example**:
    - **Edward Snowden Case (2013)** – A system administrator at the NSA leaked classified data, demonstrating how insiders can bypass security measures.

Comparison Table

Aspect	External Breaches	Internal Breaches
Source	Hackers, cybercriminals, nation-states	Employees, contractors, insiders
Attack Vector	Exploiting software/network flaws, phishing	Abuse of credentials, accidental leaks
Motivation	Financial gain, espionage, disruption	Revenge, ideology, negligence
Difficulty to Detect	Moderate (perimeter defenses exist)	High (trusted access bypasses defenses)
Examples	Equifax (2017), Colonial Pipeline (2021)	Snowden leaks (2013), Capital One insider (2019)

Why Traditional Defenses Fail Against Modern Intrusions

Traditional defenses such as firewalls, signature-based antivirus, and simple intrusion detection systems (IDS) are increasingly ineffective against advanced threats due to several reasons:

1. **Polymorphic and Fileless Malware**
    - Attackers use malware that constantly changes signatures or operates in memory, bypassing traditional antivirus detection.
2. **Zero-Day Exploits**
    - Newly discovered vulnerabilities (unknown to vendors) allow attackers to infiltrate before patches are available.

3. **Encrypted Traffic**
    - With most web traffic encrypted (HTTPS), attackers can hide malicious payloads within encrypted streams, making inspection harder.
4. **Insider Threat Complexity**
    - Traditional defenses focus on external threats, but insiders already have legitimate access, making anomalies difficult to flag.
5. **Supply Chain Blind Spots**
    - Organizations trust third-party software and hardware providers, but compromised updates can inject malicious code (e.g., SolarWinds).
6. **Advanced Persistent Threats (APTs)**
    - Highly sophisticated, long-term intrusions that use stealthy techniques (low-and-slow attacks) evade detection for months or years.

(c) Designing a Predictive Model for Anticipating Intrusions

1. Objective

The goal is to build a predictive system that can estimate the likelihood of different intrusion types (e.g., phishing, malware, insider threat, DDoS) based on historical patterns, organizational risk factors, and real-time security event data.

---

2. Model Design Framework

Inputs (Features)

- **Network activity**: unusual traffic spikes, port scanning attempts, failed login attempts.
- **System logs**: error messages, unauthorized access attempts, privilege escalation events.
- **User behavior**: irregular working hours, large file transfers, unusual login locations.
- **Threat intelligence**: indicators of compromise (IoCs), known malware signatures, dark web chatter.
- **Business context**: industry sector, critical assets (finance vs. healthcare = different risks).

Outputs (Predictions)

- Probability distribution over intrusion categories:
    - 62% likelihood of phishing
    - 20% likelihood of insider misuse
    - 20% likelihood of ransomware attack
    - 15% likelihood of DDoS attempt

---

3. Proposed Approach

1. **Data Collection**
    - Gather historical incident data from SIEM (Security Information and Event Management) tools.
    - Use public datasets (e.g., **NSL-KDD**, **CICIDS2017**, **UNSW-NB15**) for training.

2. **Feature Engineering**
    - Extract statistical features: connection duration, failed logins, payload size.
    - Behavioral features: deviation from normal user/device activity.
3. **Model Selection**
    - **Machine Learning Options**:
        - Random Forest → robust against imbalanced data.
        - Gradient Boosting (XGBoost/LightGBM) → good for tabular intrusion data.
    - **Deep Learning Option**:
        - LSTM (Recurrent Neural Networks) for detecting sequential attack patterns in logs.
4. **Deployment Design**
    - Integrate with SIEM or SOC (Security Operations Center).
    - Real-time scoring of events → prioritized alerts for analysts.
    - Periodic retraining with new threat intelligence.

---

4. Python Implementation (Prototype Example)

```
# ---------- Predictive Intrusion Detection Model with Visualizations ----------
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
import warnings

# Optional: Suppress FutureWarnings (just in case)
warnings.filterwarnings("ignore", category=FutureWarning)
warnings.filterwarnings("ignore", category=UserWarning)

# ---------- Step 1: Load dataset ----------
df = pd.read_csv("intrusion_dataset.csv")

# Separate features and labels
X = df.drop("intrusion_type", axis=1)
y = df["intrusion_type"]
```

```
# ---------- Step 2: Train-Test Split ----------
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y
)

# ---------- Step 3: Train Model ----------
model = RandomForestClassifier(n_estimators=200, random_state=42)
model.fit(X_train, y_train)

# ---------- Step 4: Predictions ----------
y_pred = model.predict(X_test)

# ---------- Step 5: Evaluation ----------
print("=== Classification Report ===")
print(classification_report(y_test, y_pred))

# Confusion Matrix Visualization
plt.figure(figsize=(8,6))
cm = confusion_matrix(y_test, y_pred, labels=model.classes_)
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=model.classes_, yticklabels=model.classes_)
plt.title("Confusion Matrix: Intrusion Detection")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

# ---------- Step 6: Intrusion Distribution ----------
plt.figure(figsize=(6,4))
sns.countplot(y, order=y.value_counts().index, palette="Set2")
plt.title("Distribution of Intrusion Types in Dataset")
plt.xlabel("Count")
plt.ylabel("Intrusion Type")
plt.show()
```

```
# ---------- Step 7: Feature Importance ----------
feature_importance = pd.Series(model.feature_importances_,
index=X.columns).sort_values(ascending=False)


plt.figure(figsize=(7,4))
sns.barplot(x=feature_importance.values, y=feature_importance.index, palette="viridis")
plt.title("Feature Importance for Predicting Intrusions")
plt.xlabel("Importance Score")
plt.ylabel("Feature")
plt.show()

# ---------- Step 8: Example Prediction with Probabilities ----------
# Always use DataFrame with correct feature names (prevents warnings)
new_event = pd.DataFrame([{
    "failed_logins": 2,
    "packet_size": 1500,
    "connection_duration": 3,
    "privilege_escalation": 0,
    "traffic_anomaly_score": 0.7
}])

prediction = model.predict(new_event)[0]
probs = model.predict_proba(new_event)[0]

print("\nPredicted Intrusion Type:", prediction)
print("\n=== Probability Distribution Across Intrusion Types ===")
for intr_type, prob in zip(model.classes_, probs):
    print(f"{intr_type}: {prob:.2f}")

# Probability Distribution Visualization
plt.figure(figsize=(6,4))
sns.barplot(x=model.classes_, y=probs, palette="coolwarm")
plt.title("Likelihood of Each Intrusion Type for New Event")
plt.ylabel("Probability")
plt.xlabel("Intrusion Type")
plt.show()
```

Conclusion

In conclusion, mapping intrusion types, comparing internal and external breaches, and building a predictive model together give the firm a clear view of its risks. External attacks exploit system flaws while insider threats are harder to detect, and traditional defenses often miss modern tactics like zero-days or fileless malware. The predictive model helps anticipate likely intrusions, highlight key risk factors, and guide proactive defense, shifting the firm from reactive response to stronger, resilient cybersecurity.