1. DDoS Protection System for Cloud: Architecture and Tool

What’s the Problem? When a company hosts a website or online service on the cloud, it’s vulnerable to attacks where hackers flood the site with excessive traffic. This can slow down or completely shut down the website. This type of attack is called a Distributed Denial-of-Service (DDoS) attack. Attackers use many computers to send so much traffic to a website that it can’t handle it.

What’s Needed? You need to create a system that protects websites from these attacks. The system should be able to:

- Detect when an attack is happening by recognizing unusual traffic patterns (like too many requests from one place or spikes at strange times).
- Automatically respond to these attacks to keep the website running smoothly.
- Quickly recover from the attack to minimize downtime.

Solution: Design a tool or set of tools that work with cloud-based websites to monitor traffic, detect DDoS attacks, and automatically protect and recover from them.

---

2. Software Solutions to Identify Users Behind Telegram, WhatsApp, and Instagram-based Drug Trafficking

What’s the Problem? Criminals are using encrypted messaging apps like Telegram, WhatsApp, and Instagram to sell drugs. They create channels, bots, or handles to conduct illegal sales and communicate. These platforms are private and hard to monitor, making it difficult to catch these criminals.

What’s Needed? You need to develop a software solution that:

- Identifies channels, bots, or handles on Telegram, WhatsApp, and Instagram that are involved in drug trafficking.
- Helps find out who is behind these accounts by gathering identifiable information like IP addresses, mobile numbers, or email addresses.

Solution: Create a software tool that can scan these platforms for drug-related activity and track down the people behind these illegal operations.

---

3. Application to Identify Government-Issued Personally Identifiable Information (PII) Embedded in Documents

What’s the Problem? When people upload or share documents online, they might accidentally include sensitive information like government IDs (e.g., Aadhaar, PAN, Driving License). This information is private and should be protected to prevent identity theft or fraud.

What’s Needed? Develop an application that:

- Scans documents or data to detect if they contain sensitive government-issued PII.
- Alerts users if their documents contain such information and helps them remove or hide it if necessary.
- Helps organizations ensure they are not storing or processing unnecessary sensitive information.

Solution: Create a tool or application that can find and handle government-issued personal information in documents, alerting users to protect their privacy and helping organizations comply with data protection regulations.

1. Web-Scraping Tool for Vulnerabilities

Problem: Organizations need to quickly know about serious security flaws in their equipment (like routers, servers, software) to fix them before they're exploited. Currently, they have to wait for databases like the National Vulnerability Database (NVD) to update, which can be slow.

What is Needed:

- A tool that automatically checks official websites of equipment manufacturers (OEMs) and other relevant sites for new critical or high-severity vulnerabilities.
- The tool should quickly notify the relevant people via email when it finds such vulnerabilities.

Solution:

1. **Web Scraping**: Create a script that visits these websites, looks for vulnerability updates, and reads the details.
2. **Data Extraction**: Extract key information like product name, severity level, and suggested fixes.
3. **Alerts**: Automatically send this information to specific email addresses.

2. Real-Time Cyber Incident Feed for Indian Cyber Space

Problem: To protect critical infrastructure, organizations need real-time information about cyber threats specifically related to India. This helps them respond to and mitigate risks effectively.

What is Needed:

- A system to gather and analyze reports about cyber incidents from various sources (like forums, social media, and news sites).
- This system should use machine learning to find where these reports are published and organize the data in a useful way.

Solution:

1. **Machine Learning**: Use algorithms to find relevant sources of cyber incident information.
2. **Data Collection**: Build a system to collect and store data from these sources.
3. **Insights and Reporting**: Create visualizations and reports to help understand the threats.

3. RE-DACT: Redaction Tool

Problem: Organizations need a tool to hide sensitive information in documents while keeping the document's structure and useful data intact. This is important for protecting privacy and data security.

What is Needed:

- A tool that can redact (hide) sensitive information from various types of files (text, images, PDFs) based on user preferences.
- The tool should allow different levels of redaction and not store or share the original data.

Solution:

1. **Redaction Features**: Use machine learning to identify and obscure sensitive information.
2. **User Interface**: Design an easy-to-use interface where users can choose how much to redact.
3. **Security**: Ensure that the tool does not save or expose the original data.

7. Audit Script for Windows 11 and Linux OS

Problem: Organizations need to ensure their systems comply with security best practices outlined by CIS (Center for Internet Security) benchmarks. Checking compliance manually is time-consuming and prone to errors.

What is Needed:

- **Automated Scripts**: Develop scripts that automatically check if Windows 11 and Linux systems meet CIS security benchmarks.
- **Features**:
  - **User-Friendly Interface**: A GUI to run audits and view reports.
  - **Customizable**: Adaptable to specific organizational needs.
  - **Reliable**: Accurate in identifying deviations from best practices.
  - **Updatable**: Easy to update as CIS benchmarks change.

Solution:

1. **Scripting**: Create PowerShell scripts for Windows and Bash/Python scripts for Linux to check system settings.
2. **Reporting**: Generate detailed reports of compliance or non-compliance.
3. **Customization**: Allow users to configure audits based on specific requirements.

8. Identify Cryptographic Algorithm Using AI/ML

Problem: Identifying which cryptographic algorithm was used from a dataset can reveal weaknesses and improve security.

What is Needed:

- **AI/ML Model**: Develop a model to analyze datasets and identify the cryptographic algorithm used.
- **Software Solution**: Build a tool that inputs data and outputs possible algorithms.

Solution:

1. **Data Analysis**: Use AI/ML techniques to analyze datasets and predict the cryptographic algorithm.
2. **Tool Development**: Create software that automates this identification process.

9. Agent-Less Windows System Vulnerability and Network Scanner

Problem: Windows systems often have vulnerabilities that can be exploited. Many users don't keep their systems updated.

What is Needed:

- **Agent-Less Scanner**: A tool that scans Windows systems for vulnerabilities without needing an installed agent.
- **Features**:
  - **System and Network Information**: Check settings, installed software, and network configurations.
  - **Open-Source Exploits**: Find available exploits and patches.
  - **Reporting**: Consolidate findings into a report.

Solution:

1. **Vulnerability Detection**: Create a scanner that can audit system settings and network configurations.
2. **Report Generation**: Compile results into an understandable format like PDF or HTML.

10. Universal Switch Set with Data Encryption

Problem: Legacy systems often lack modern security features, making them vulnerable. Retrofitting these systems can be complex and costly.

What is Needed:

- **Universal Switch Set**: A switch that can encrypt and decrypt data to secure legacy systems.
- **Features**:
  - **Modular Design**: Includes encryption/decryption engines and key management.
  - **Compatibility**: Works with various legacy applications.
  - **Key Management**: Securely handle encryption keys.

Solution:

1. **Encryption/Decryption**: Develop switches that integrate easily with old systems to add encryption.
2. **Customization**: Allow configuration for different encryption needs.

11. Centralized Application-Context Aware Firewall

Problem: Managing firewall rules for individual applications on endpoints can be complex. A centralized solution is needed to streamline this process.

What is Needed:

- **Application Firewall**: Controls network access for each application separately.
- **Central Management**: A web console to manage and deploy firewall policies.
- **Anomaly Detection**: Use AI/ML to detect unusual network behavior.

Solution:

1. **Firewall Agent**: Implement a firewall that can control traffic based on the application.
2. **Central Console**: Develop a web-based dashboard to manage policies and monitor network traffic.

12. Cyber Triage Tool for Digital Forensics

Problem: Investigators need an efficient way to analyze digital evidence and identify key information quickly.

What is Needed:

- **Forensic Tool**: Automate data collection and analysis from forensic images.
- **Features**:
  - **Data Collection**: Import and scan various data formats.
  - **Anomaly Detection**: Use AI/ML to find significant patterns.
  - **Reporting**: Generate detailed, customizable reports.

Solution:

1. **Automated Analysis**: Develop a tool that can scan data and identify indicators of compromise.
2. **Visualization and Reporting**: Provide interactive timelines and summaries.

13. De-anonymizing Entities on the TOR Network

Problem: The TOR network is used for illegal activities and hides the identities of its users. Identifying these users is challenging.

What is Needed:

- **Identification Tool**: Find the real IP addresses and other personal details of individuals operating illegal sites on TOR.

Solution:

1. **Tracking Techniques**: Develop methods to trace the actual IP addresses and other identifiable information from TOR network activity.

14. Improving Open Source Software Security Using Fuzzing

Problem: Open-source software can have vulnerabilities that are not always discovered through traditional testing.

What is Needed:

- **Fuzzing Harness**: Develop a harness to test the Sumatra PDF Reader using fuzzing techniques.
- **Features**:
  - **Target Functions**: Identify key functions to test.
  - **Fuzzer Setup**: Use tools to generate test inputs and analyze results.
  - **Reporting**: Document the findings and vulnerabilities.

Solution:

1. **Fuzzing Harness**: Create a tool that feeds various inputs to the Sumatra PDF Reader and records any crashes or issues.
2. **Technical Report**: Provide detailed documentation of the testing process and results.

15. Recovery of Deleted Data from XFS and Btrfs Filesystems

Problem: Recovering deleted data from advanced file systems like XFS and Btrfs is complex due to their sophisticated structures.

What is Needed:

- **Data Recovery Tool**: Develop techniques to recover deleted files and their metadata from XFS and Btrfs systems.
- **Features**:
  - **Comprehensive Recovery**: Retrieve various file types and associated metadata.
  - **User Interface**: Provide a GUI or CLI for easy navigation and reporting.

Solution:

1. **Recovery Algorithms**: Create methods to locate and recover deleted files and metadata.
2. **User-Friendly Interface**: Develop a tool that allows users to easily recover and manage their data.