Group 12

Title: Ransomware Simulation, Detection, and Mitigation in a Virtualized Ubuntu Environment Using Java.

Abstract:

The ransomware threat has become an active cyber threat that performs data encryption then asks victims to pay for file decryption despite outsmarting security protocols. Signature-based antivirus solutions do not protect against modern variants of ransomware since these threats utilize polymorphic encryption in conjunction with advanced evasion techniques. This research applies virtualized Ubuntu with Java-based ransomware that employs AES-256 encryption to investigate the attack patterns and create defensive protocols for ransomware responses. The encryption mechanism selects one or more folders from a specified directory where it encrypts and permanently removes the saved files from the initial system. The WatchService API from Java operates through real time to identify suspicious file alterations and sudden encryption events. The system automatically detects ransomware activity by terminating the nefarious process to stop additional harm and this process is followed by a recovery operation which retrieves files from backup points. A behavioral-based detection system proves more effective than signature-based systems according to this research and effective proactive measures demonstrate how ransomware can be stopped before substantial data loss happens. Ransomware defense methods should consider adding machine learning-based anomaly detection with network-layer defenses to improve antiransomware capabilities.

Project Plan:

Step 2: Encryption

The ransomware program starts by applying AES-256 encryption to all files situated in the critical/ directory. Ransomware attackers prefer AES-256 encryption due to its formidable security strength in contemporary attacks. The ransomware program performs individual file encryption of target files in the scanning process while deleting unencrypted versions of the files. The files become inaccessible after encryption because the decryption key remains essential to reveal their contents during a realistic ransomware attack. The generated encryption key functions without local storage which prevents victims from recovering files by themselves.

Step 3: Infection

Firstly the encryption script receives Java .jar file compilation to function as ransomware executable during victim system infection simulation. The .jar file will receive manual execution in an Ubuntu virtual machine (VM) to show how common ransomware infections happen when people interact with phishing emails and exploit kits and malicious downloads. The .jar file executes in the background without user notification in order to start file encryption operations. During the simulation phase the spread of ransomware through the system and its immediate effects on the platform will become apparent.

Group 12

Step 4: Monitoring

The WatchService API will enable a real-time file activity detection system through Java development. The system operates through continuous monitoring of the "critical/" directory where it identifies any changes in file contents and new file additions or file deletion events. Any large growth of encrypted files with the .enc extension triggers the monitoring system to record this activity as a possible threat. The system operates like professional ransomware protection services by analyzing system behaviors instead of specific signatures which makes it capable of recognizing emerging ransomware variations.

Step 5: Detection

The system starts detecting ransomware behavior after the monitoring starts. The warning system monitors file renaming activity because multiple files using the .enc extensionextension occurs within a brief time span can signal ransomware intrusion. The system will produce an alert that notifies the administrator whenever too many system files undergo changes during a brief period of time. The detection method tracks resource utilization spikes since ransomware requires high CPU and disk usage during encryption of numerous system files. The detection of ransomware requires this step to prevent encryption of all system files.

Step 6: Mitigation

Automated response measures activate when ransomware activities get detected by the system. The ransomware script termination procedure activates to prevent file encryption completion which reduces system deterioration. Processes will restore files to their last backup state to allow data recovery without becoming a victim of ransom payments. Without an available backup the mitigation system works to place the infected machine in quarantine in order to stop its spread to other machines. Due to this protection measure attacks can happen but the resulting damage remains minimal while recovery becomes possible.

Research Papers:

1. Ransomware Prevention and Mitigation Techniques
2. Cryptographic Ransomware Encryption Detection: Survey
3. Ransomware Detection and Classification Strategies
4. Ransomware Attack Protection: A Cryptographic Approach
5. Ransomware Detection, Avoidance, and Mitigation Scheme